HW #5 Tips

```
knitr::opts_chunk$set(echo = TRUE)
library(tm)
```

```
## Warning: package 'tm' was built under R version 3.5.3
```

```
## Loading required package: NLP
```

```
library(stringr)
```

```
## Warning: package 'stringr' was built under R version 3.5.3
```

```
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
library(stringi)
```

```
## Warning: package 'stringi' was built under R version 3.5.3
```

```
library(Matrix)
library(tidytext)
```

```
## Warning: package 'tidytext' was built under R version 3.5.3
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.3
```

```
##
## 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
```

```r
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'

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

```r
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 3.5.3

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

```r
library(rpart)
library(rattle)
```

```
## Loading required package: tibble

## Warning: package 'tibble' was built under R version 3.5.3

## Loading required package: bitops

## Rattle: A free graphical interface for data science with R.
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```r
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 3.5.3
```

```r
library(RColorBrewer)



# Unused Libs
#library(slam)
#library(quanteda)
#library(SnowballC)
#library(arules)
#library(proxy)
#library(cluster)
#library(Cairo)
#library(CORElearn)
#library(mclust)
#library(plyr)
#library(proxy)
```

Load the data

In this example, we loaad the Fed Papers in Corpus format. Its always a good idea to peak at the data to be sure it has loaded correctly!!

```
#Load Fed Papers Corpus
FedPapersCorpus <- Corpus(DirSource("FedPapersCorpus"))
(numberFedPapers<-length(FedPapersCorpus))
```

```
## [1] 85
```

```
## The following will show you that you read in all the documents
(summary(FedPapersCorpus))
```

```
##                      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
```

```r
(meta(FedPapersCorpus[[1]]))
```

```
##   author       : character(0)
##   datetimestamp: 2020-08-06 19:35:42
##   description  : character(0)
##   heading      : character(0)
##   id            : dispt_fed_49.txt
##   language      : en
##   origin        : character(0)
```

```
(meta(FedPapersCorpus[[1]],5))
```

```
## [1] "dispt_fed_49.txt"
```

Cleaning and Preprocessing

Choosing some good stop words can really go a long way to improve modeling results. There are also many other parameters one can tweak and tune using the DocumentTermMatrix function. See many below.

```
#Data Preparation and Transformation on Fed Papers
##Remove punctuation,numbers, and space
(getTransformations())
```

```
## [1] "removeNumbers"     "removePunctuation" "removeWords"
## [4] "stemDocument"      "stripWhitespace"
```

```
(nFedPapersCorpus<-length(FedPapersCorpus))
```

```
## [1] 85
```

```
##Ignore extremely rare words i.e. terms that appear in less then 1% of the documents
(minTermFreq <-30)
```

```
## [1] 30
```

```
##Ignore overly common words i.e. terms that appear in more than 50% of the documents
(maxTermFreq <-1000)
```

```
## [1] 1000
```

```
(MyStopwords <- c("will","one","two", "may","less","publius","Madison","Alexand", "Alexander", "James",
```

```
##  [1] "will"      "one"       "two"       "may"       "less"      "publius"
##  [7] "Madison"   "Alexand"   "Alexander" "James"     "Hamilton"  "Jay"
## [13] "well"      "might"     "without"   "small"     "single"    "several"
## [19] "but"       "very"      "can"       "must"      "also"      "any"
## [25] "and"       "are"       "however"   "into"      "almost"    "can"
## [31] "for"       "add"       "Author"
```

```
(STOPS <-stopwords('english'))
```

```
##   [1] "i"          "me"         "my"         "myself"     "we"
##   [6] "our"        "ours"       "ourselves"  "you"        "your"
##  [11] "yours"      "yourself"   "yourselves" "he"         "him"
##  [16] "his"        "himself"    "she"        "her"        "hers"
##  [21] "herself"    "it"         "its"        "itself"     "they"
##  [26] "them"       "their"      "theirs"     "themselves" "what"
##  [31] "which"      "who"        "whom"       "this"       "that"
##  [36] "these"      "those"      "am"         "is"         "are"
##  [41] "was"        "were"       "be"         "been"       "being"
##  [46] "have"       "has"        "had"        "having"     "do"
##  [51] "does"       "did"        "doing"      "would"      "should"
##  [56] "could"      "ought"      "i'm"        "you're"     "he's"
##  [61] "she's"      "it's"       "we're"      "they're"    "i've"
##  [66] "you've"     "we've"      "they've"    "i'd"        "you'd"
##  [71] "he'd"       "she'd"      "we'd"       "they'd"     "i'll"
##  [76] "you'll"     "he'll"      "she'll"     "we'll"      "they'll"
##  [81] "isn't"      "aren't"     "wasn't"     "weren't"    "hasn't"
##  [86] "haven't"    "hadn't"     "doesn't"    "don't"      "didn't"
##  [91] "won't"      "wouldn't"   "shan't"     "shouldn't"  "can't"
##  [96] "cannot"     "couldn't"   "mustn't"    "let's"      "that's"
## [101] "who's"      "what's"     "here's"     "there's"    "when's"
## [106] "where's"    "why's"      "how's"      "a"          "an"
## [111] "the"        "and"        "but"        "if"         "or"
## [116] "because"    "as"         "until"      "while"      "of"
## [121] "at"         "by"         "for"        "with"       "about"
## [126] "against"    "between"    "into"       "through"    "during"
## [131] "before"     "after"      "above"      "below"      "to"
## [136] "from"       "up"         "down"       "in"         "out"
## [141] "on"         "off"        "over"       "under"      "again"
## [146] "further"    "then"       "once"       "here"       "there"
## [151] "when"       "where"      "why"        "how"        "all"
## [156] "any"        "both"       "each"       "few"        "more"
## [161] "most"       "other"      "some"       "such"       "no"
## [166] "nor"        "not"        "only"       "own"        "same"
## [171] "so"         "than"       "too"        "very"
```

```
Papers_DTM <- DocumentTermMatrix(FedPapersCorpus,
                         control = list(
                           stopwords = TRUE,
                           wordLengths=c(3, 15),
                           removePunctuation = T,
                           removeNumbers = T,
                           tolower=T,
                           stemming = T,
                           remove_separators = T,
                           stopwords = MyStopwords,
                           removeWords=STOPS,
                           removeWords=MyStopwords,
                           bounds = list(global = c(minTermFreq, maxTermFreq))
                         ))

##inspect FedPapers Document Term Matrix (DTM)
DTM <- as.matrix(Papers_DTM)
#(DTM[1:11,1:10])
```

Vectorization

Vectorizing words is often done by encoding frequency information. Below we take a peak at the frequency of the words. Next some normalization techniques are tried. Which works best . . . ?? Try many and assess the results!!!

```
##Look at word freuquncies
WordFreq <- colSums(as.matrix(Papers_DTM))
(head(WordFreq))
```

```
##       abl   absolut    accord       act     addit administr
##        74        63        71       139        61        90
```

```
(length(WordFreq))
```

```
## [1] 427
```

```
ord <- order(WordFreq)
(WordFreq[head(ord)])
```

```
##    jame   expos furnish    word  unless   bound
##      30      34      36      36      37      38
```

```
(WordFreq[tail(ord)])
```

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

```
## Row Sums per Fed Papers
(Row_Sum_Per_doc <- rowSums((as.matrix(Papers_DTM))))
```

```
##    dispt_fed_49.txt    dispt_fed_50.txt    dispt_fed_51.txt    dispt_fed_52.txt
##                 514                 338                 658                 565
##    dispt_fed_53.txt    dispt_fed_54.txt    dispt_fed_55.txt    dispt_fed_56.txt
##                 701                 582                 647                 553
##    dispt_fed_57.txt    dispt_fed_62.txt    dispt_fed_63.txt  Hamilton_fed_1.txt
##                 613                 698                 955                 483
## Hamilton_fed_11.txt Hamilton_fed_12.txt Hamilton_fed_13.txt Hamilton_fed_15.txt
##                 564                 539                 318                 815
## Hamilton_fed_16.txt Hamilton_fed_17.txt Hamilton_fed_21.txt Hamilton_fed_22.txt
##                 558                 477                 537                 985
## Hamilton_fed_23.txt Hamilton_fed_24.txt Hamilton_fed_25.txt Hamilton_fed_26.txt
##                 560                 519                 570                 670
## Hamilton_fed_27.txt Hamilton_fed_28.txt Hamilton_fed_29.txt Hamilton_fed_30.txt
##                 466                 507                 541                 585
## Hamilton_fed_31.txt Hamilton_fed_32.txt Hamilton_fed_33.txt Hamilton_fed_34.txt
##                 510                 442                 522                 618
## Hamilton_fed_35.txt Hamilton_fed_36.txt Hamilton_fed_59.txt  Hamilton_fed_6.txt
##                 663                 824                 603                 461
## Hamilton_fed_60.txt Hamilton_fed_61.txt Hamilton_fed_65.txt Hamilton_fed_66.txt
##                 657                 444                 560                 646
## Hamilton_fed_67.txt Hamilton_fed_68.txt Hamilton_fed_69.txt  Hamilton_fed_7.txt
##                 443                 449                 811                 580
## Hamilton_fed_70.txt Hamilton_fed_71.txt Hamilton_fed_72.txt Hamilton_fed_73.txt
##                 852                 473                 539                 696
## Hamilton_fed_74.txt Hamilton_fed_75.txt Hamilton_fed_76.txt Hamilton_fed_77.txt
##                 282                 597                 594                 586
## Hamilton_fed_78.txt Hamilton_fed_79.txt  Hamilton_fed_8.txt Hamilton_fed_80.txt
##                 891                 301                 533                 771
## Hamilton_fed_81.txt Hamilton_fed_82.txt Hamilton_fed_83.txt Hamilton_fed_84.txt
##                1188                 504                1598                1255
## Hamilton_fed_85.txt  Hamilton_fed_9.txt       HM_fed_18.txt       HM_fed_19.txt
##                 773                 520                 443                 466
##       HM_fed_20.txt       Jay_fed_2.txt       Jay_fed_3.txt       Jay_fed_4.txt
##                 395                 477                 515                 463
##       Jay_fed_5.txt      Jay_fed_64.txt  Madison_fed_10.txt  Madison_fed_14.txt
##                 401                 692                 884                 553
##  Madison_fed_37.txt  Madison_fed_38.txt  Madison_fed_39.txt  Madison_fed_40.txt
##                 723                 874                 859                 857
##  Madison_fed_41.txt  Madison_fed_42.txt  Madison_fed_43.txt  Madison_fed_44.txt
##                1020                 800                 993                 927
##  Madison_fed_45.txt  Madison_fed_46.txt  Madison_fed_47.txt  Madison_fed_48.txt
##                 724                 832                 925                 565
##  Madison_fed_58.txt
##                 655
```

```r
## Create a normalized version of Papers_DTM
Papers_M <- as.matrix(Papers_DTM)
Papers_M_N1 <- apply(Papers_M, 1, function(i) round(i/sum(i),3))
Papers_Matrix_Norm <- t(Papers_M_N1)


## Convert to matrix and view
Papers_dtm_matrix = as.matrix(Papers_DTM)
#str(Papers_dtm_matrix)
#(Papers_dtm_matrix[c(1:11),c(2:10)])
```

Label the Data

Below we label the data, prepare for modeling, and create some wordclouds for fun.

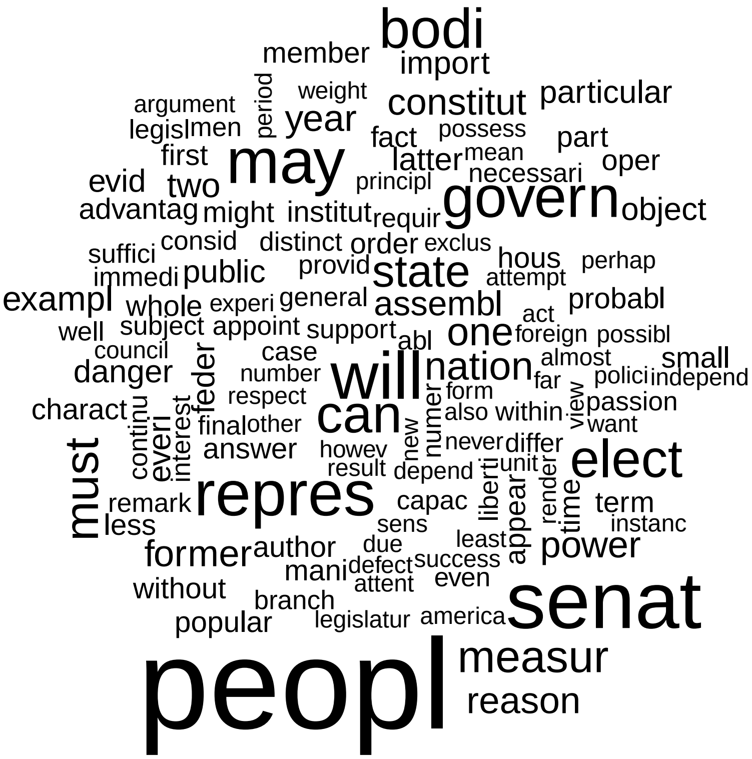
```r
## Also convert to DF
Papers_DF <- as.data.frame(as.matrix(Papers_Matrix_Norm))
Papers_DF1<- Papers_DF%>%add_rownames()
```

```
## Warning: Deprecated, use tibble::rownames_to_column() instead.
```

```r
names(Papers_DF1)[1]<-"Author"
Papers_DF1[1:11,1]="dispt"
Papers_DF1[12:62,1]="hamil"
Papers_DF1[63:85,1]="madis"
head(Papers_DF1)
```

```
## # A tibble: 6 x 428
##   Author   abl absolut accord   act addit administr admit adopt advantag affair
##   <chr>  <dbl>   <dbl>  <dbl> <dbl> <dbl>     <dbl> <dbl> <dbl>    <dbl>  <dbl>
## 1 dispt  0.004   0       0      0     0         0.002 0.002 0        0.008  0
## 2 dispt  0       0.006   0      0     0         0.006 0     0        0.003  0
## 3 dispt  0.002   0.003   0      0     0.002     0.002 0.005 0        0      0.002
## 4 dispt  0.002   0.002   0      0.002 0.002     0     0     0.002    0.004  0
## 5 dispt  0       0       0.001  0.003 0         0     0.001 0        0.003  0.013
## 6 dispt  0       0       0.003  0.002 0         0     0.009 0.002    0.007  0
## # ... with 417 more variables: affect <dbl>, afford <dbl>, alexand <dbl>,
## #   almost <dbl>, alon <dbl>, alreadi <dbl>, also <dbl>, alway <dbl>,
## #   america <dbl>, among <dbl>, amount <dbl>, anoth <dbl>, answer <dbl>,
## #   appear <dbl>, appli <dbl>, applic <dbl>, appoint <dbl>, apprehens <dbl>,
## #   argument <dbl>, aris <dbl>, articl <dbl>, assembl <dbl>, attempt <dbl>,
## #   attend <dbl>, attent <dbl>, author <dbl>, avoid <dbl>, becom <dbl>,
## #   best <dbl>, better <dbl>, bodi <dbl>, bound <dbl>, branch <dbl>,
## #   britain <dbl>, calcul <dbl>, call <dbl>, can <dbl>, capac <dbl>,
## #   care <dbl>, carri <dbl>, case <dbl>, caus <dbl>, certain <dbl>,
## #   chang <dbl>, charact <dbl>, circumst <dbl>, citizen <dbl>, civil <dbl>,
## #   class <dbl>, clear <dbl>, collect <dbl>, combin <dbl>, commit <dbl>,
## #   common <dbl>, communiti <dbl>, complet <dbl>, compos <dbl>, concern <dbl>,
## #   conclus <dbl>, conduct <dbl>, confeder <dbl>, confederaci <dbl>,
## #   confid <dbl>, confin <dbl>, congress <dbl>, connect <dbl>, consequ <dbl>,
## #   consid <dbl>, consider <dbl>, consist <dbl>, constitu <dbl>,
## #   constitut <dbl>, contend <dbl>, continu <dbl>, contrari <dbl>,
## #   control <dbl>, convent <dbl>, council <dbl>, countri <dbl>, cours <dbl>,
## #   danger <dbl>, decid <dbl>, decis <dbl>, declar <dbl>, defect <dbl>,
## #   defens <dbl>, degre <dbl>, deliber <dbl>, depart <dbl>, depend <dbl>,
## #   deriv <dbl>, descript <dbl>, design <dbl>, desir <dbl>, determin <dbl>,
## #   differ <dbl>, difficulti <dbl>, direct <dbl>, dispos <dbl>, disposit <dbl>,
## #   ...
```

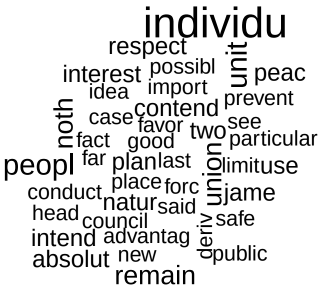
```
#Wordcloud Visualization Hamilton, Madison and Disputed Papers
DisputedPapersWC<- wordcloud(colnames(Papers_dtm_matrix), Papers_dtm_matrix[11,])
```



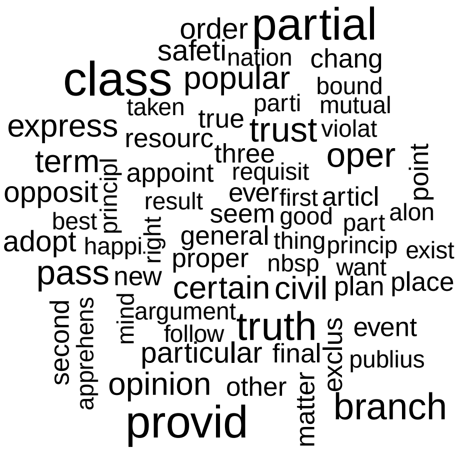
```
(head(sort(as.matrix(Papers_dtm_matrix)[11,], decreasing = TRUE), n=50))
```

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

```
HamiltonPapersWC <-wordcloud(colnames(Papers_dtm_matrix),Papers_dtm_matrix[12:62,])
```



```
MadisonPapersHW <-wordcloud(colnames(Papers_dtm_matrix), Papers_dtm_matrix[63:77,])
```



Experimental Design

Now that the data is labeled, its time to design an experiment. Below we randomly select a train and test set for validation using function: sample.int() .

```
##Make Train and Test sets
numDisputed = 11
numTotalPapers = nrow(Papers_DF1)
trainRatio <- .60
set.seed(11) # Set Seed so that same sample can be reproduced in future also
sample <- sample.int(n = numTotalPapers-numDisputed, size = floor(trainRatio*numTotalPapers), replace =
newSample = sample + numDisputed
train <- Papers_DF1[newSample, ]
test <- Papers_DF1[-newSample, ]
# train / test ratio
length(newSample)/nrow(Papers_DF1)
```

```
## [1] 0.6
```

Classification

We are now ready to train and test using classifiers. Below we use a few different decision tree models. Try different params and prunings to get varied results.

Use fancyRpartPlot to visualize the learned tree models. What do these diagrams display???

```
##Decision Tree Models
#Train Tree Model 1
train_tree1 <- rpart(Author ~ ., data = train, method="class", control=rpart.control(cp=0))
summary(train_tree1)
```

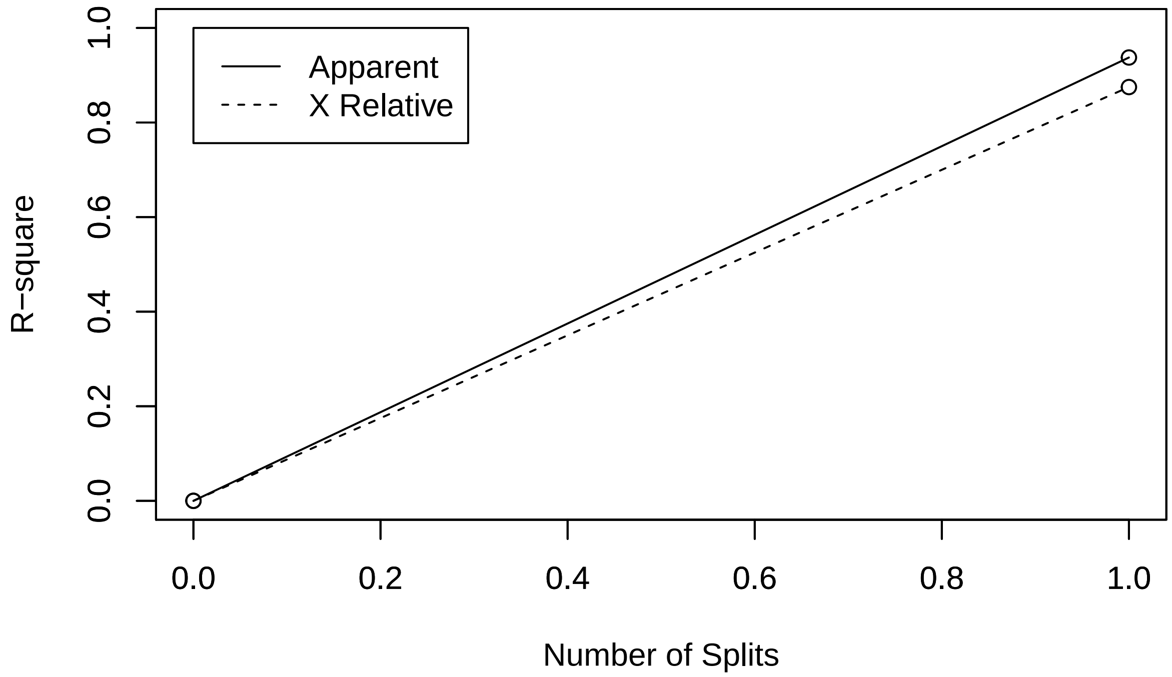
```
## Call:
## rpart(formula = Author ~ ., data = train, method = "class", control = rpart.control(cp = 0))
##   n= 51
## 
##       CP nsplit rel error xerror       xstd
## 1 0.9375      0    1.0000  1.000 0.20710422
## 2 0.0000      1    0.0625  0.125 0.08663791
## 
## Variable importance
##  alexand hamilton     upon     jame     form    thing 
##       23       23       20       15        9        9 
## 
## Node number 1: 51 observations,    complexity param=0.9375
##   predicted class=hamil  expected loss=0.3137255  P(node) =1
##     class counts:    35    16
##    probabilities: 0.686 0.314 
##   left son=2 (36 obs) right son=3 (15 obs)
##   Primary splits:
##       alexand  < 5e-04  to the right, improve=20.01634, (0 missing)
##       hamilton < 5e-04  to the right, improve=20.01634, (0 missing)
##       upon     < 0.003  to the right, improve=20.01634, (0 missing)
##       jame     < 5e-04  to the left,  improve=14.59013, (0 missing)
##       thing    < 0.0015 to the right, improve=11.29412, (0 missing)
##   Surrogate splits:
##       hamilton < 5e-04  to the right, agree=1.000, adj=1.000, (0 split)
##       upon     < 0.0015 to the right, agree=0.961, adj=0.867, (0 split)
##       jame     < 5e-04  to the left,  agree=0.902, adj=0.667, (0 split)
##       form     < 0.0065 to the left,  agree=0.824, adj=0.400, (0 split)
##       thing    < 0.0015 to the right, agree=0.824, adj=0.400, (0 split)
## 
## Node number 2: 36 observations
##   predicted class=hamil  expected loss=0.02777778  P(node) =0.7058824
##     class counts:    35     1
##    probabilities: 0.972 0.028 
## 
## Node number 3: 15 observations
##   predicted class=madis  expected loss=0  P(node) =0.2941176
##     class counts:     0    15
##    probabilities: 0.000 1.000
```

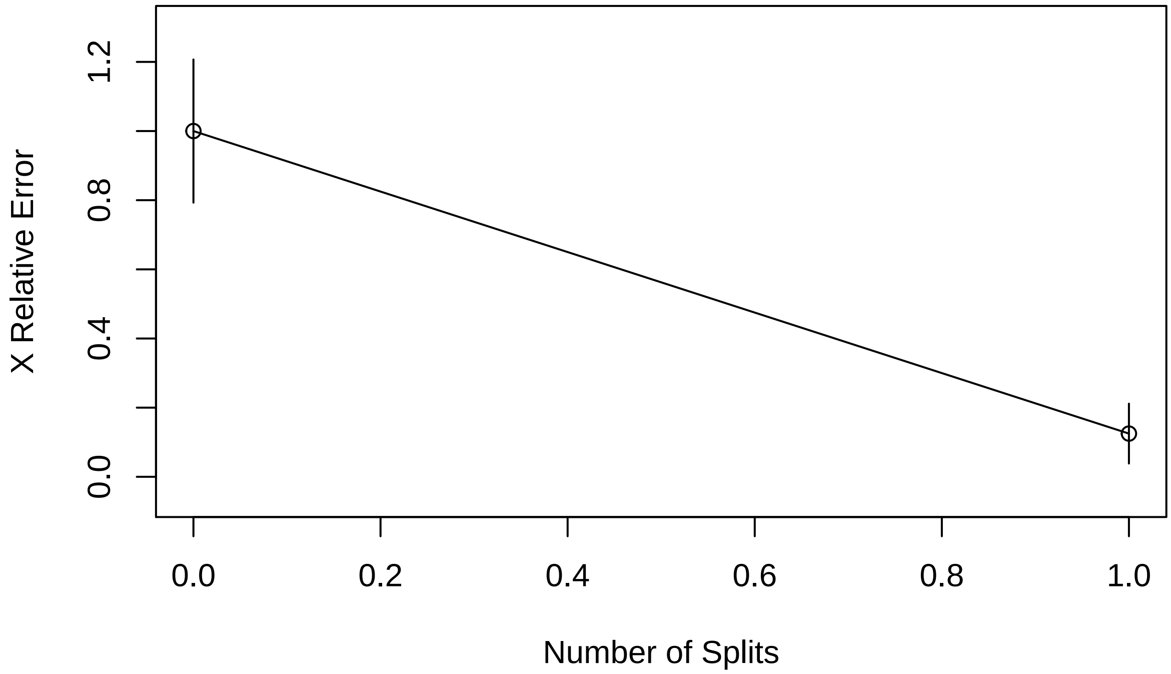
```
#predict the test dataset using the model for train tree No. 1
predicted1= predict(train_tree1, test, type="class")
#plot number of splits
rsq.rpart(train_tree1)
```

```
## 
```

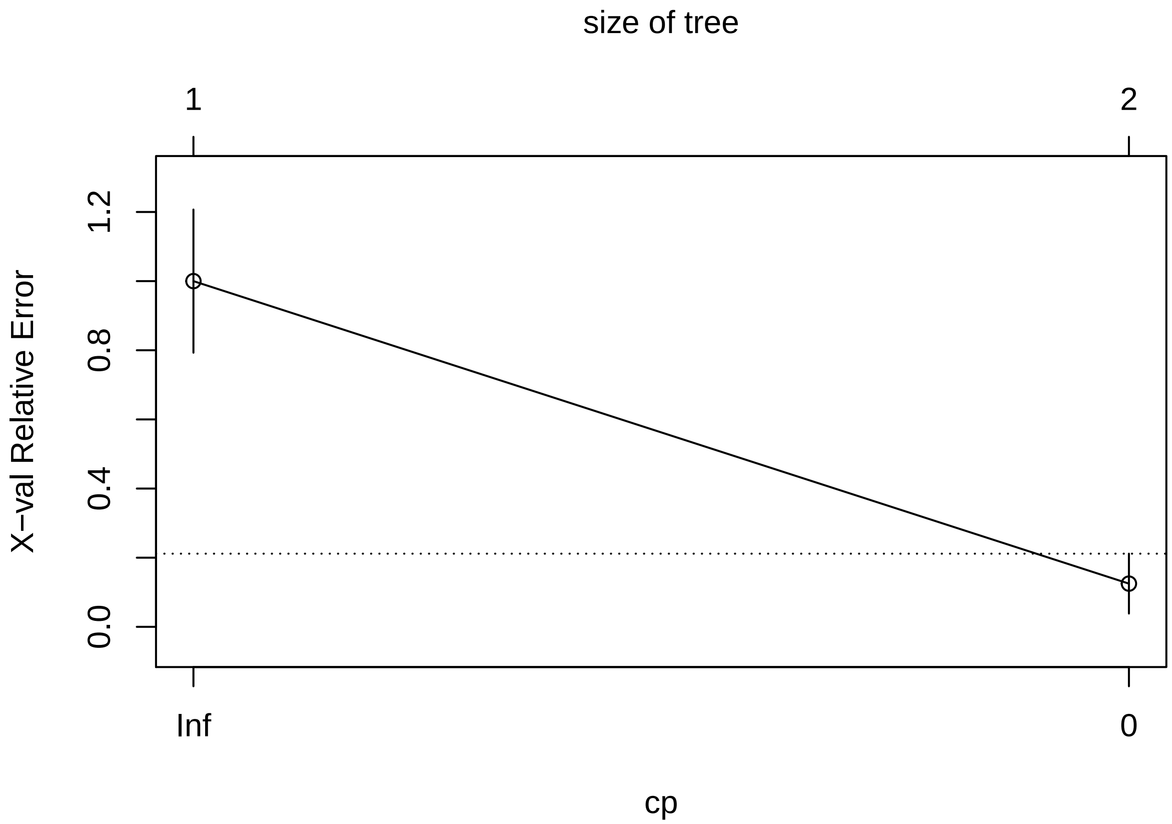
```
## Classification tree:
## rpart(formula = Author ~ ., data = train, method = "class", control = rpart.control(cp = 0))
##
## Variables actually used in tree construction:
## [1] alexand
##
## Root node error: 16/51 = 0.31373
##
## n= 51
##
##       CP nsplit rel error xerror     xstd
## 1 0.9375      0    1.0000  1.000 0.207104
## 2 0.0000      1    0.0625  0.125 0.086638
```

```
## Warning in rsq.rpart(train_tree1): may not be applicable for this method
```

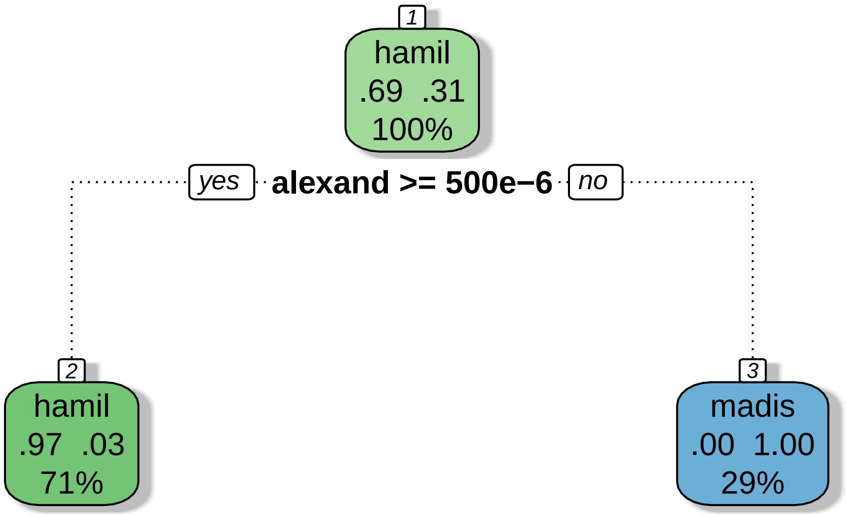




```r
plotcp(train_tree1)
```



```
#plot the decision tree
fancyRpartPlot(train_tree1)
```



Rattle 2020−Aug−06 15:35:44 jerem

```r
#confusion matrix to find correct and incorrect predictions
table(Authorship=predicted1, true=test$Author)
```

```
##           true
## Authorship dispt hamil madis
##      hamil    11    16     2
##      madis     0     0     5
```

```r
#Train Tree Model 2
train_tree2 <- rpart(Author ~ ., data = train, method="class", control=rpart.control(cp=0, minsplit = 2
summary(train_tree2)
```

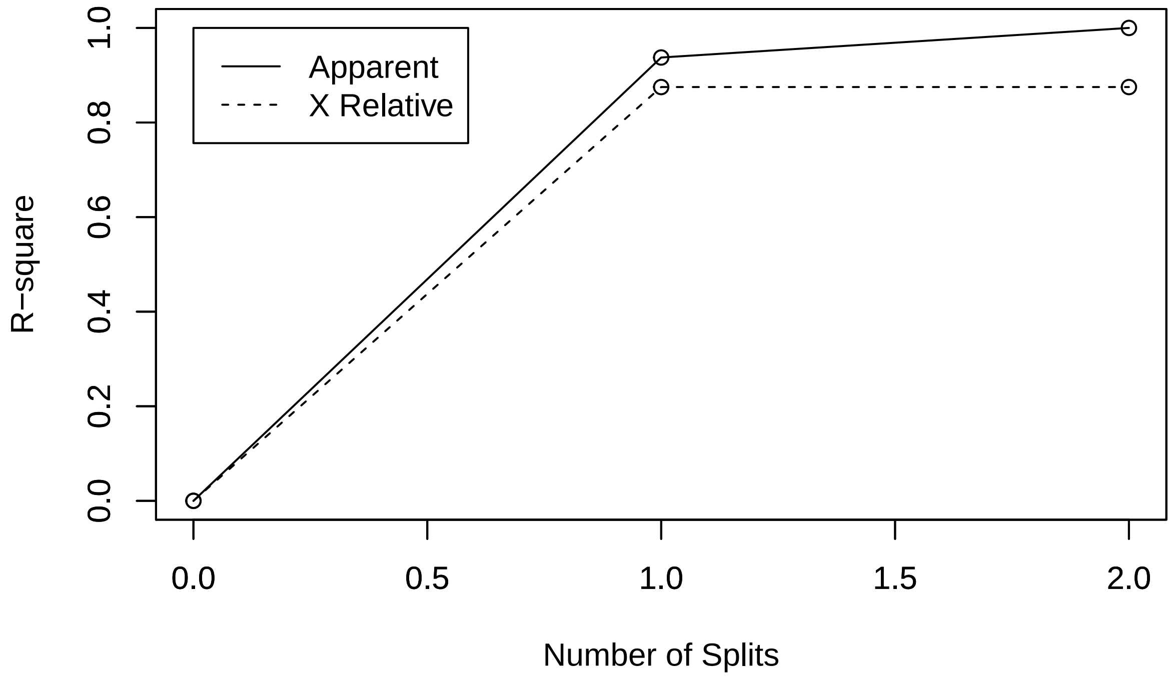
```
## Call:
## rpart(formula = Author ~ ., data = train, method = "class", control = rpart.control(cp = 0,
##     minsplit = 2, maxdepth = 5))
##   n= 51
##
##       CP nsplit rel error xerror       xstd
## 1 0.9375      0    1.0000  1.000 0.20710422
## 2 0.0625      1    0.0625  0.125 0.08663791
## 3 0.0000      2    0.0000  0.125 0.08663791
##
## Variable importance
##  alexand hamilton     upon     jame     form    thing   accord
##       23       23       20       15        9        9        2
```

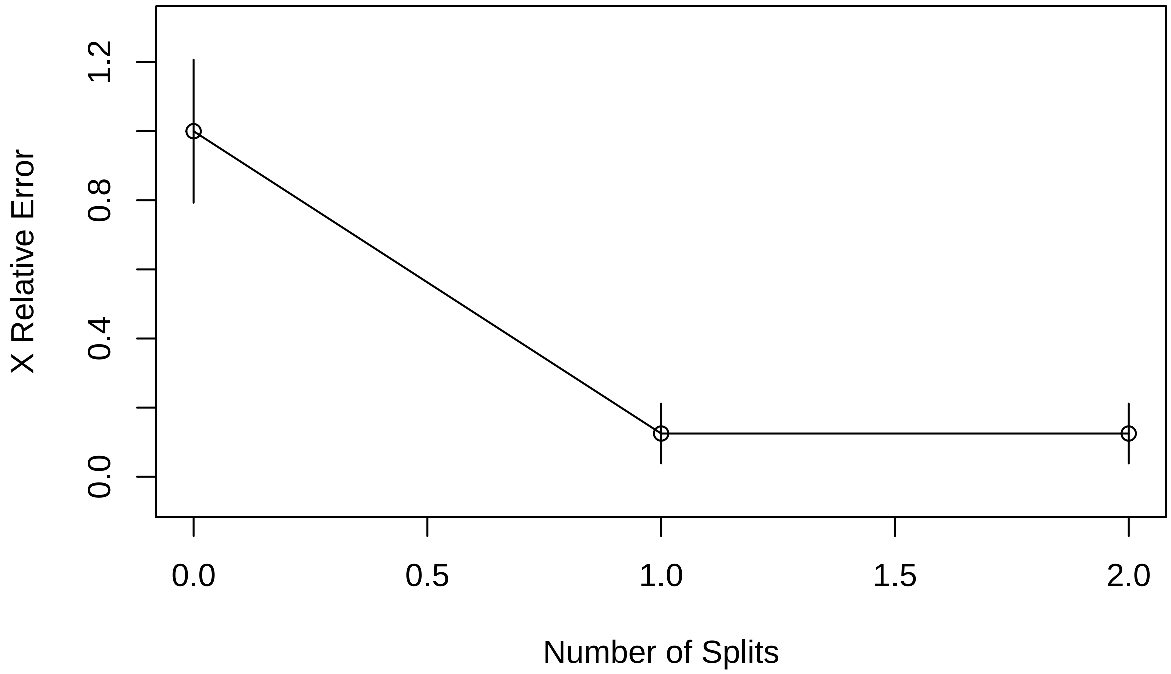
```
## 
## Node number 1: 51 observations,    complexity param=0.9375
##   predicted class=hamil  expected loss=0.3137255  P(node) =1
##     class counts:    35    16
##    probabilities: 0.686 0.314 
##   left son=2 (36 obs) right son=3 (15 obs)
##   Primary splits:
##       alexand  < 5e-04  to the right, improve=20.01634, (0 missing)
##       hamilton < 5e-04  to the right, improve=20.01634, (0 missing)
##       upon     < 0.003  to the right, improve=20.01634, (0 missing)
##       jame     < 5e-04  to the left,  improve=14.59013, (0 missing)
##       thing    < 0.0015 to the right, improve=11.29412, (0 missing)
##   Surrogate splits:
##       hamilton < 5e-04  to the right, agree=1.000, adj=1.000, (0 split)
##       upon     < 0.0015 to the right, agree=0.961, adj=0.867, (0 split)
##       jame     < 5e-04  to the left,  agree=0.902, adj=0.667, (0 split)
##       form     < 0.0065 to the left,  agree=0.824, adj=0.400, (0 split)
##       thing    < 0.0015 to the right, agree=0.824, adj=0.400, (0 split)
## 
## Node number 2: 36 observations,    complexity param=0.0625
##   predicted class=hamil  expected loss=0.02777778  P(node) =0.7058824
##     class counts:    35     1
##    probabilities: 0.972 0.028 
##   left son=4 (35 obs) right son=5 (1 obs)
##   Primary splits:
##       accord  < 0.0065 to the left,  improve=1.944444, (0 missing)
##       affair  < 0.004  to the left,  improve=1.944444, (0 missing)
##       alexand < 0.005  to the left,  improve=1.944444, (0 missing)
##       among   < 0.008  to the left,  improve=1.944444, (0 missing)
##       becom   < 0.0045 to the left,  improve=1.944444, (0 missing)
## 
## Node number 3: 15 observations
##   predicted class=madis  expected loss=0  P(node) =0.2941176
##     class counts:     0    15
##    probabilities: 0.000 1.000 
## 
## Node number 4: 35 observations
##   predicted class=hamil  expected loss=0  P(node) =0.6862745
##     class counts:    35     0
##    probabilities: 1.000 0.000 
## 
## Node number 5: 1 observations
##   predicted class=madis  expected loss=0  P(node) =0.01960784
##     class counts:     0     1
##    probabilities: 0.000 1.000
```

```r
#predict the test dataset using the model for train tree No. 1
predicted2= predict(train_tree2, test, type="class")
#plot number of splits
rsq.rpart(train_tree2)
```

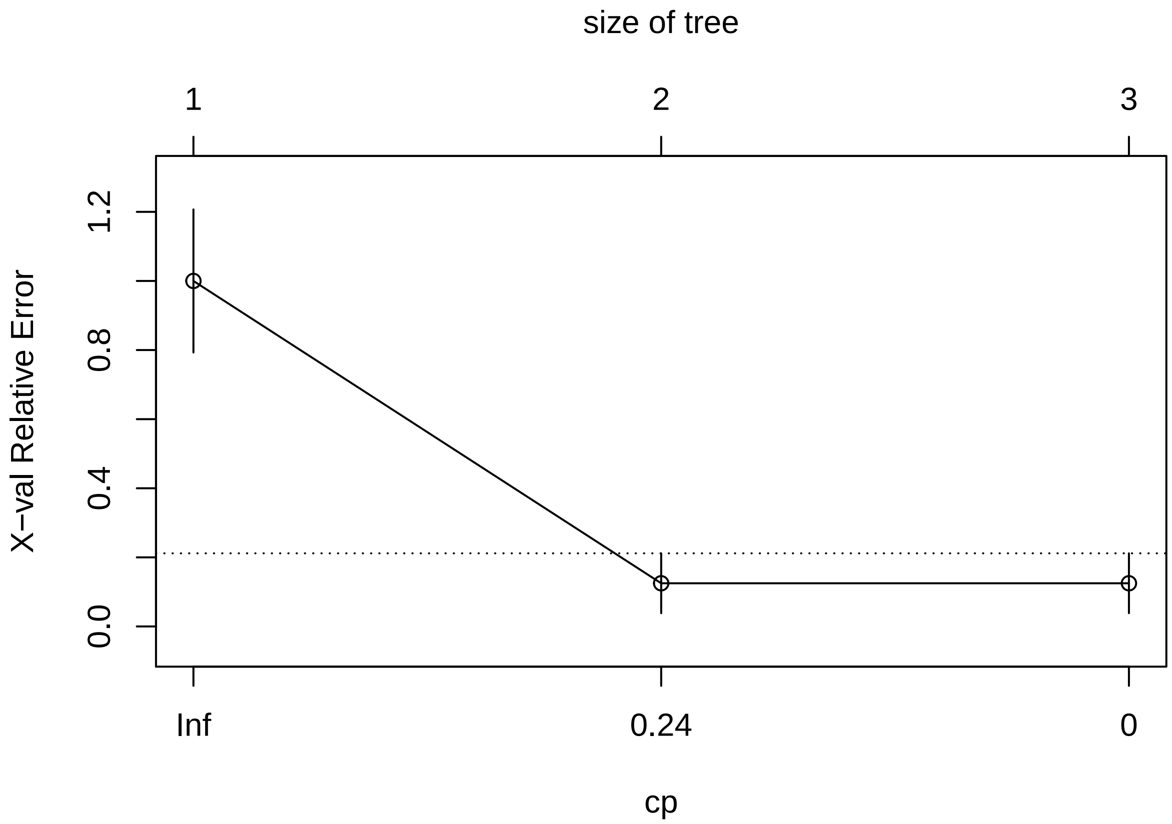
```
## 
## Classification tree:
## rpart(formula = Author ~ ., data = train, method = "class", control = rpart.control(cp = 0,
##     minsplit = 2, maxdepth = 5))
##
## Variables actually used in tree construction:
## [1] accord  alexand
##
## Root node error: 16/51 = 0.31373
##
## n= 51
##
##       CP nsplit rel error xerror     xstd
## 1 0.9375      0    1.0000  1.000 0.207104
## 2 0.0625      1    0.0625  0.125 0.086638
## 3 0.0000      2    0.0000  0.125 0.086638
```

```
## Warning in rsq.rpart(train_tree2): may not be applicable for this method
```

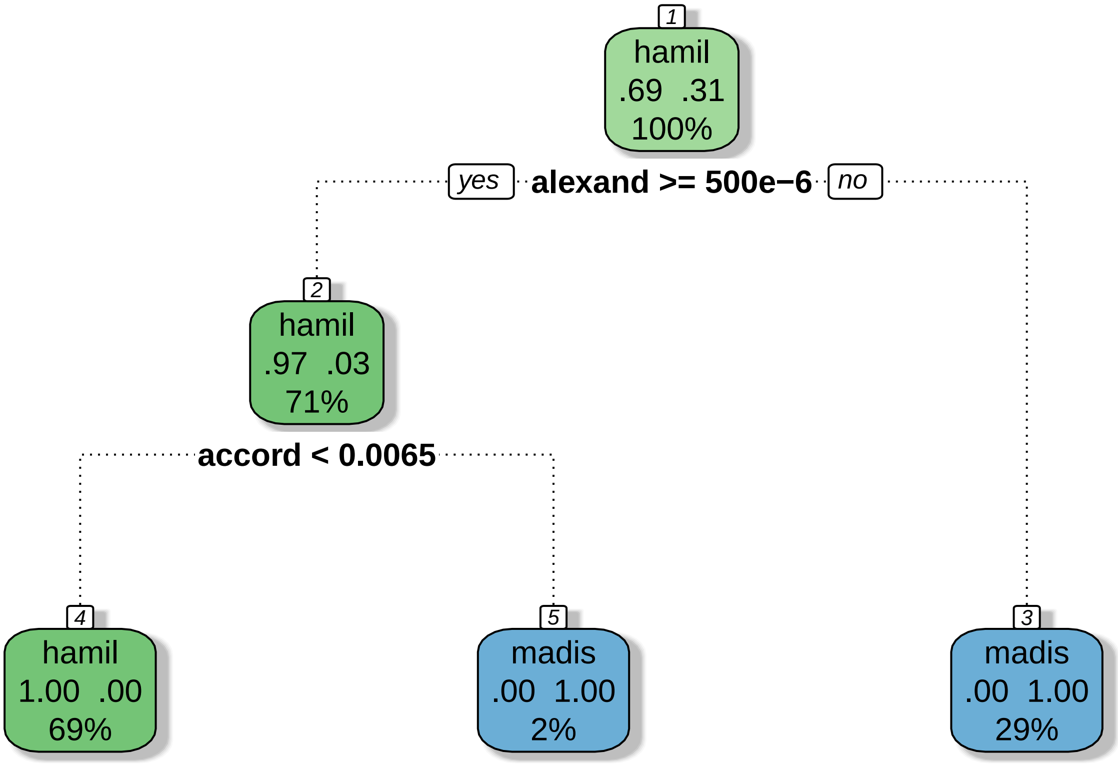




```
plotcp(train_tree2)
```



```
#plot the decision tree
fancyRpartPlot(train_tree2)
```



Rattle 2020−Aug−06 15:35:44 jerem

```
#confusion matrix to find correct and incorrect predictions
table(Authorship=predicted2, true=test$Author)
```

```
##             true
## Authorship dispt hamil madis
##      hamil    11    16     2
##      madis     0     0     5
```