

HW #5 Tips

Load the data

In this example, we load 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

```

```

(meta(FedPapersCorpus[[1]]))

```

```

## author : character(0)
## timestamp: 2019-10-01 17:00:16
## 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"
## [6] "publius"   "Madison"  "Alexand"  "Alexander" "James"
## [11] "Hamilton"  "Jay"      "well"     "might"    "without"
## [16] "small"     "single"   "several"  "but"      "very"
## [21] "can"       "must"     "also"     "any"      "and"
## [26] "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"       "will"
```

```
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 frequencies
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
##      514              338              658
##      dispt_fed_52.txt    dispt_fed_53.txt    dispt_fed_54.txt
##      565              701              582
##      dispt_fed_55.txt    dispt_fed_56.txt    dispt_fed_57.txt
##      647              553              613
##      dispt_fed_62.txt    dispt_fed_63.txt    Hamilton_fed_1.txt
##      698              955              483
## Hamilton_fed_11.txt    Hamilton_fed_12.txt    Hamilton_fed_13.txt
##      564              539              318
## Hamilton_fed_15.txt    Hamilton_fed_16.txt    Hamilton_fed_17.txt
##      815              558              477
## Hamilton_fed_21.txt    Hamilton_fed_22.txt    Hamilton_fed_23.txt
##      537              985              560
## Hamilton_fed_24.txt    Hamilton_fed_25.txt    Hamilton_fed_26.txt
##      519              570              670
## Hamilton_fed_27.txt    Hamilton_fed_28.txt    Hamilton_fed_29.txt
##      466              507              541
## Hamilton_fed_30.txt    Hamilton_fed_31.txt    Hamilton_fed_32.txt
##      585              510              442
## Hamilton_fed_33.txt    Hamilton_fed_34.txt    Hamilton_fed_35.txt
##      522              618              663
## Hamilton_fed_36.txt    Hamilton_fed_59.txt    Hamilton_fed_6.txt
##      824              603              461
## Hamilton_fed_60.txt    Hamilton_fed_61.txt    Hamilton_fed_65.txt
##      657              444              560
## Hamilton_fed_66.txt    Hamilton_fed_67.txt    Hamilton_fed_68.txt
##      646              443              449
## Hamilton_fed_69.txt    Hamilton_fed_7.txt    Hamilton_fed_70.txt
##      811              580              852
## Hamilton_fed_71.txt    Hamilton_fed_72.txt    Hamilton_fed_73.txt
##      473              539              696
## Hamilton_fed_74.txt    Hamilton_fed_75.txt    Hamilton_fed_76.txt
##      282              597              594
## Hamilton_fed_77.txt    Hamilton_fed_78.txt    Hamilton_fed_79.txt
##      586              891              301
```

```
## Hamilton_fed_8.txt Hamilton_fed_80.txt Hamilton_fed_81.txt
##           533           771           1188
## Hamilton_fed_82.txt Hamilton_fed_83.txt Hamilton_fed_84.txt
##           504           1598           1255
## Hamilton_fed_85.txt Hamilton_fed_9.txt HM_fed_18.txt
##           773           520           443
## HM_fed_19.txt HM_fed_20.txt Jay_fed_2.txt
##           466           395           477
## Jay_fed_3.txt Jay_fed_4.txt Jay_fed_5.txt
##           515           463           401
## Jay_fed_64.txt Madison_fed_10.txt Madison_fed_14.txt
##           692           884           553
## Madison_fed_37.txt Madison_fed_38.txt Madison_fed_39.txt
##           723           874           859
## Madison_fed_40.txt Madison_fed_41.txt Madison_fed_42.txt
##           857           1020           800
## Madison_fed_43.txt Madison_fed_44.txt Madison_fed_45.txt
##           993           927           724
## Madison_fed_46.txt Madison_fed_47.txt Madison_fed_48.txt
##           832           925           565
## Madison_fed_58.txt
##           655

## 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.

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

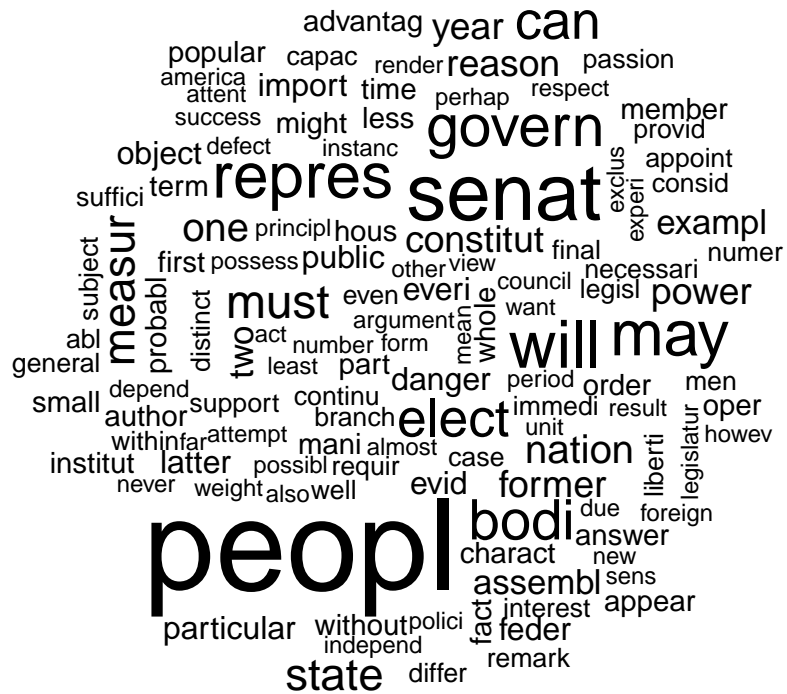
## Warning: Deprecated, use tibble::rownames_to_column() instead.
names(Papers_DF1)[1] <- "Author"
Papers_DF1[1:11,1] = "dispt"
Papers_DF1[12:62,1] = "hamil"
Papers_DF1[63:77,1] = "madis"
head(Papers_DF1)

## # A tibble: 6 x 428
##   Author   abl absolut accord   act addit administr admit adopt advantag
##   <chr>   <dbl>   <dbl>   <dbl> <dbl> <dbl>       <dbl> <dbl> <dbl>   <dbl>
## 1 dispt  0.004     0       0       0     0         0.002 0.002 0       0.008
```

```
## 2 dispt 0      0.006 0      0      0      0.006 0      0      0.003
## 3 dispt 0.002 0.003 0      0      0.002 0.002 0.005 0      0
## 4 dispt 0.002 0.002 0      0.002 0.002 0      0      0.002 0.004
## 5 dispt 0      0      0.001 0.003 0      0      0.001 0      0.003
## 6 dispt 0      0      0.003 0.002 0      0      0.009 0.002 0.007
## # ... with 418 more variables: affair <dbl>, 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>, ...
```

#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
##	42	24	19	18	18	16
##	bodi	can	elect	must	measur	state
##	15	14	14	12	11	11
##	nation	one	constitut	former	power	reason
##	9	9	8	8	8	8
##	year	assembl	exampl	two	danger	everi
##	8	7	7	7	6	6
##	evid	feder	import	latter	object	particular
##	6	6	6	6	6	6
##	public	advantag	answer	appear	author	charact
##	6	5	5	5	5	5
##	fact	first	hous	institut	less	mani
##	5	5	5	5	5	5
##	member	might	oper	order	part	popular
##	5	5	5	5	5	5
##	probl	small				
##	5	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
trainRatio <- .60
set.seed(11) # Set Seed so that same sample can be reproduced in future also
sample <- sample.int(n = nrow(Papers_DF1), size = floor(trainRatio*nrow(Papers_DF1)), replace = FALSE)
train <- Papers_DF1[sample, ]
test <- Papers_DF1[-sample, ]
# train / test ratio
length(sample)/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 0.40      0      1.00   1.00 0.1428011
## 2 0.36      1      0.60   0.84 0.1405870
## 3 0.00      2      0.24   0.48 0.1211708
##
## Variable importance
##      upon      jame      kind matter      thing alexand      union      branch      legis1
##      17      14      9      9      9      8      8      5      5
##      peac      attent      might
##      5      5      5
##
## Node number 1: 51 observations,      complexity param=0.4
## predicted class=hamil expected loss=0.4901961 P(node) =1
## class counts:      10      26      11      1      1      1      1
## probabilities: 0.196 0.510 0.216 0.020 0.020 0.020 0.020
## left son=2 (27 obs) right son=3 (24 obs)
## Primary splits:
##      upon < 0.0035 to the right, improve=15.907410, (0 missing)
##      jame < 5e-04 to the left, improve=13.379630, (0 missing)
##      alexand < 5e-04 to the right, improve= 9.398111, (0 missing)
##      hamilton < 5e-04 to the right, improve= 9.398111, (0 missing)
##      kind < 5e-04 to the right, improve= 6.268519, (0 missing)
## Surrogate splits:
##      jame < 5e-04 to the left, agree=0.922, adj=0.833, (0 split)
##      kind < 5e-04 to the right, agree=0.784, adj=0.542, (0 split)
##      matter < 5e-04 to the right, agree=0.765, adj=0.500, (0 split)
##      thing < 0.0015 to the right, agree=0.765, adj=0.500, (0 split)
##      alexand < 5e-04 to the right, agree=0.745, adj=0.458, (0 split)
##
## Node number 2: 27 observations
## predicted class=hamil expected loss=0.03703704 P(node) =0.5294118
## class counts:      0      26      1      0      0      0      0
## probabilities: 0.000 0.963 0.037 0.000 0.000 0.000 0.000
##
## Node number 3: 24 observations,      complexity param=0.36
## predicted class=dispt expected loss=0.5833333 P(node) =0.4705882
## class counts:      10      0      10      1      1      1      1
## probabilities: 0.417 0.000 0.417 0.042 0.042 0.042 0.042
## left son=6 (14 obs) right son=7 (10 obs)
## Primary splits:
##      union < 0.0055 to the left, improve=7.128571, (0 missing)
##      might < 0.0035 to the right, improve=5.842857, (0 missing)
##      alexand < 5e-04 to the right, improve=5.833333, (0 missing)
##      hamilton < 5e-04 to the right, improve=5.833333, (0 missing)
##      elect < 0.0015 to the right, improve=4.750000, (0 missing)

```

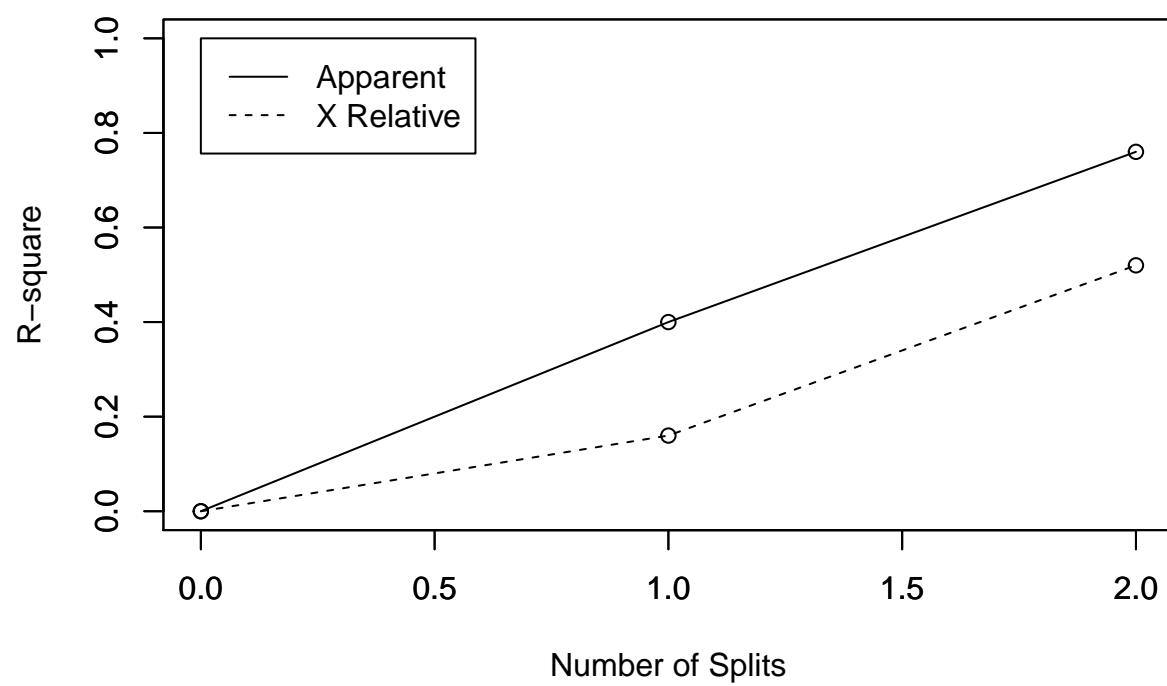
```

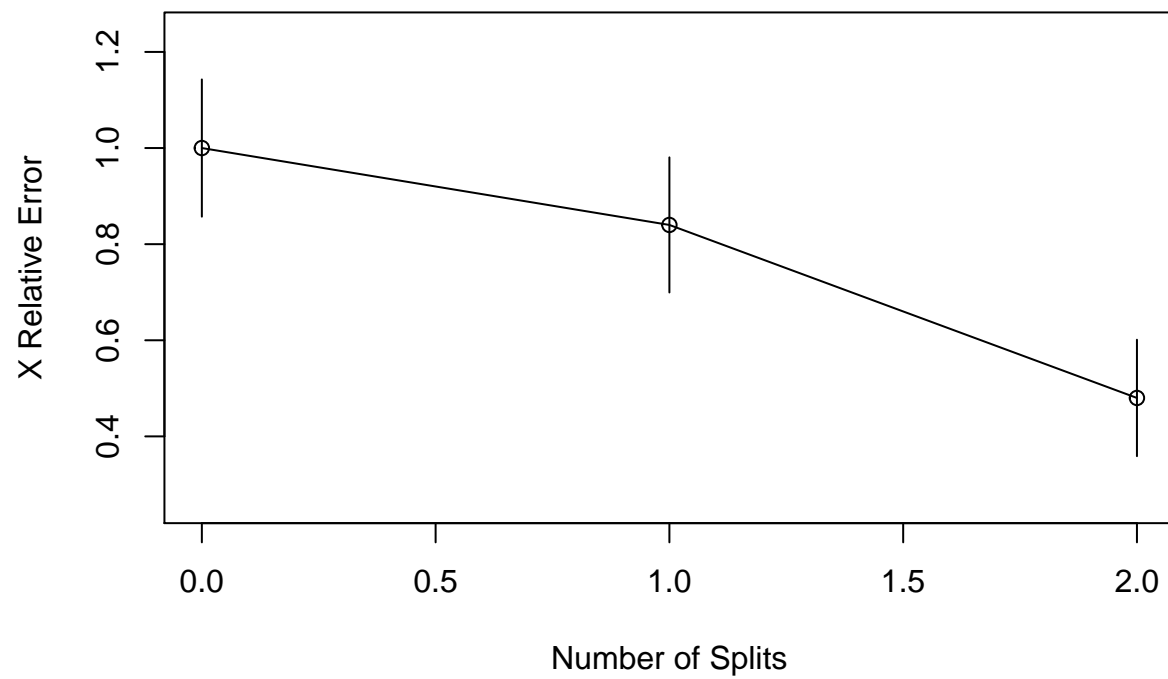
## Surrogate splits:
##   branch < 0.0015 to the right, agree=0.875, adj=0.7, (0 split)
##   legis1 < 0.0035 to the right, agree=0.875, adj=0.7, (0 split)
##   peac < 0.001 to the left, agree=0.875, adj=0.7, (0 split)
##   attent < 5e-04 to the right, agree=0.833, adj=0.6, (0 split)
##   might < 0.0035 to the right, agree=0.833, adj=0.6, (0 split)
##
## Node number 6: 14 observations
##   predicted class=dispt   expected loss=0.2857143   P(node) =0.2745098
##   class counts:    10    0    1    1    0    1    1
##   probabilities: 0.714 0.000 0.071 0.071 0.000 0.071 0.071
##
## Node number 7: 10 observations
##   predicted class=madis   expected loss=0.1   P(node) =0.1960784
##   class counts:    0    0    9    0    1    0    0
##   probabilities: 0.000 0.000 0.900 0.000 0.100 0.000 0.000
#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] union upon
##
## Root node error: 25/51 = 0.4902
##
## n= 51
##
##   CP nsplit rel error xerror   xstd
## 1 0.40    0    1.00   1.00 0.14280
## 2 0.36    1    0.60   0.84 0.14059
## 3 0.00    2    0.24   0.48 0.12117

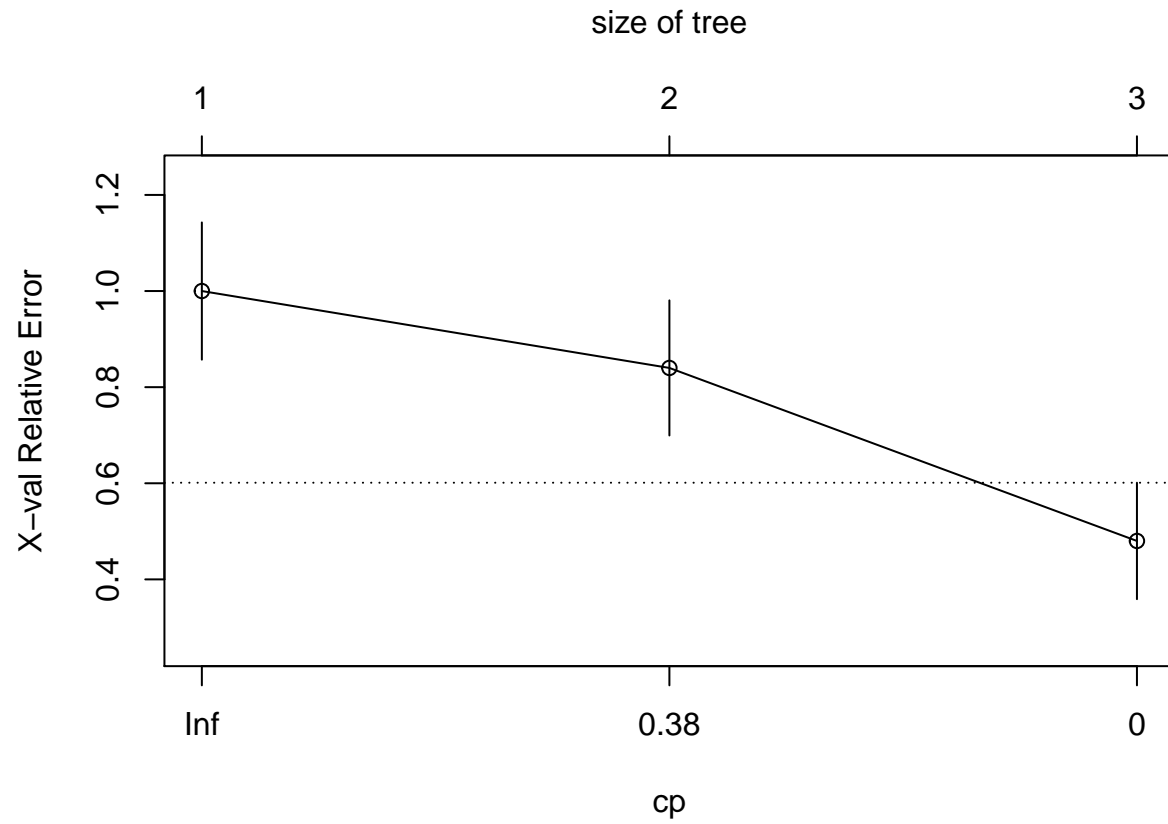
## Warning in rsq.rpart(train_tree1): may not be applicable for this method

```

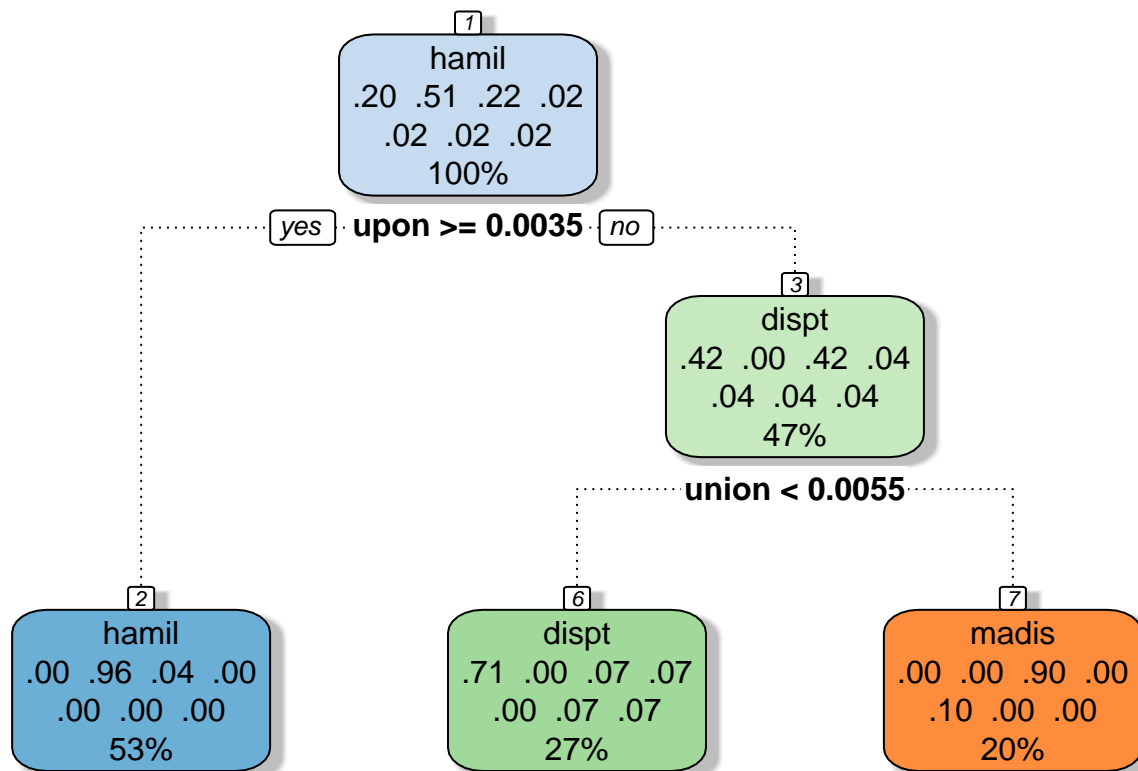




```
plotcp(train_tree1)
```



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



Rattle 2019-Oct-01 13:00:21 jerem

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

```
##
## true
## Authorship      dispt hamil madis Madison_fed_43.txt
## dispt           1     0     1             0
## hamil           0    25     0             0
## madis           0     0     3             1
## Madison_fed_42.txt 0     0     0             0
## Madison_fed_44.txt 0     0     0             0
## Madison_fed_48.txt 0     0     0             0
## Madison_fed_58.txt 0     0     0             0
##
## true
## Authorship      Madison_fed_45.txt Madison_fed_46.txt
## dispt           0             1
## hamil           0             0
## madis           1             0
## Madison_fed_42.txt 0             0
## Madison_fed_44.txt 0             0
## Madison_fed_48.txt 0             0
## Madison_fed_58.txt 0             0
##
## true
## Authorship      Madison_fed_47.txt
## dispt           1
## hamil           0
## madis           0
```



```

## Madison_fed_42.txt          0
## Madison_fed_44.txt          0
## Madison_fed_48.txt          0
## Madison_fed_58.txt          0

#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 0.40      0      1.00   1.00 0.1428011
## 2 0.36      1      0.60   1.00 0.1428011
## 3 0.04      2      0.24   0.44 0.1174901
## 4 0.00      8      0.00   0.52 0.1244911
##
## Variable importance
##      upon      jame alexand      kind      matter      thing      union      branch
##      13       11      11        7        7        7        6        4
## legisl      peac      attent      might hamilton      necess      shall      abl
##      4        4        4        4        3        2        2        2
## great      clear confeder      adopt advantag      alon      appoint      assembl
##      2        1        1        1        1        1        1        1
## avoid
##      1
##
## Node number 1: 51 observations,      complexity param=0.4
## predicted class=hamil      expected loss=0.4901961 P(node) =1
## class counts:      10      26      11      1      1      1      1
## probabilities: 0.196 0.510 0.216 0.020 0.020 0.020 0.020
## left son=2 (27 obs) right son=3 (24 obs)
## Primary splits:
##      upon < 0.0035 to the right, improve=15.907410, (0 missing)
##      jame < 5e-04 to the left, improve=13.379630, (0 missing)
##      alexand < 5e-04 to the right, improve= 9.398111, (0 missing)
##      hamilton < 5e-04 to the right, improve= 9.398111, (0 missing)
##      kind < 5e-04 to the right, improve= 6.268519, (0 missing)
## Surrogate splits:
##      jame < 5e-04 to the left, agree=0.922, adj=0.833, (0 split)
##      kind < 5e-04 to the right, agree=0.784, adj=0.542, (0 split)
##      matter < 5e-04 to the right, agree=0.765, adj=0.500, (0 split)
##      thing < 0.0015 to the right, agree=0.765, adj=0.500, (0 split)
##      alexand < 5e-04 to the right, agree=0.745, adj=0.458, (0 split)
##
## Node number 2: 27 observations,      complexity param=0.04
## predicted class=hamil      expected loss=0.03703704 P(node) =0.5294118
## class counts:      0      26      1      0      0      0      0
## probabilities: 0.000 0.963 0.037 0.000 0.000 0.000 0.000
## left son=4 (26 obs) right son=5 (1 obs)
## Primary splits:
##      alexand < 5e-04 to the right, improve=1.925926, (0 missing)

```

```

##      hamilton < 5e-04  to the right, improve=1.925926, (0 missing)
##      anoth    < 0.0085 to the left,  improve=1.925926, (0 missing)
##      congress < 0.01   to the left,  improve=1.925926, (0 missing)
##      individu < 0.007  to the left,  improve=1.925926, (0 missing)
##
## Node number 3: 24 observations,      complexity param=0.36
##   predicted class=dispt              expected loss=0.5833333 P(node) =0.4705882
##   class counts:      10      0      10      1      1      1      1
##   probabilities: 0.417 0.000 0.417 0.042 0.042 0.042 0.042
##   left son=6 (14 obs) right son=7 (10 obs)
##   Primary splits:
##       union    < 0.0055 to the left,  improve=7.128571, (0 missing)
##       might    < 0.0035 to the right, improve=5.842857, (0 missing)
##       alexand  < 5e-04  to the right, improve=5.833333, (0 missing)
##       hamilton < 5e-04  to the right, improve=5.833333, (0 missing)
##       elect    < 0.0015 to the right, improve=4.750000, (0 missing)
##   Surrogate splits:
##       branch  < 0.0015 to the right, agree=0.875, adj=0.7, (0 split)
##       legisl  < 0.0035 to the right, agree=0.875, adj=0.7, (0 split)
##       peac    < 0.001  to the left,  agree=0.875, adj=0.7, (0 split)
##       attent  < 5e-04  to the right, agree=0.833, adj=0.6, (0 split)
##       might   < 0.0035 to the right, agree=0.833, adj=0.6, (0 split)
##
## Node number 4: 26 observations
##   predicted class=hamil              expected loss=0 P(node) =0.5098039
##   class counts:      0      26      0      0      0      0      0
##   probabilities: 0.000 1.000 0.000 0.000 0.000 0.000 0.000
##
## Node number 5: 1 observations
##   predicted class=madis              expected loss=0 P(node) =0.01960784
##   class counts:      0      0      1      0      0      0      0
##   probabilities: 0.000 0.000 1.000 0.000 0.000 0.000 0.000
##
## Node number 6: 14 observations,      complexity param=0.04
##   predicted class=dispt              expected loss=0.2857143 P(node) =0.2745098
##   class counts:      10      0      1      1      0      1      1
##   probabilities: 0.714 0.000 0.071 0.071 0.000 0.071 0.071
##   left son=12 (10 obs) right son=13 (4 obs)
##   Primary splits:
##       alexand  < 5e-04  to the right, improve=3.571429, (0 missing)
##       hamilton < 5e-04  to the right, improve=3.571429, (0 missing)
##       necess   < 0.0035 to the left,  improve=2.753247, (0 missing)
##       shall    < 0.0045 to the left,  improve=2.753247, (0 missing)
##       clear    < 0.0015 to the left,  improve=2.071429, (0 missing)
##   Surrogate splits:
##       hamilton < 5e-04  to the right, agree=1.000, adj=1.00, (0 split)
##       necess   < 0.0035 to the left, agree=0.929, adj=0.75, (0 split)
##       shall    < 0.0045 to the left, agree=0.929, adj=0.75, (0 split)
##       clear    < 0.0015 to the left, agree=0.857, adj=0.50, (0 split)
##       confeder < 0.0025 to the left, agree=0.857, adj=0.50, (0 split)
##
## Node number 7: 10 observations,      complexity param=0.04
##   predicted class=madis              expected loss=0.1 P(node) =0.1960784
##   class counts:      0      0      9      0      1      0      0

```

```

## probabilities: 0.000 0.000 0.900 0.000 0.100 0.000 0.000
## left son=14 (9 obs) right son=15 (1 obs)
## Primary splits:
## great < 0.0015 to the right, improve=1.8, (0 missing)
## present < 5e-04 to the right, improve=1.8, (0 missing)
## attempt < 0.005 to the left, improve=1.8, (0 missing)
## avoid < 5e-04 to the left, improve=1.8, (0 missing)
## complet < 0.0025 to the left, improve=1.8, (0 missing)
##
## Node number 12: 10 observations
## predicted class=dispt expected loss=0 P(node) =0.1960784
## class counts: 10 0 0 0 0 0 0
## probabilities: 1.000 0.000 0.000 0.000 0.000 0.000 0.000
##
## Node number 13: 4 observations, complexity param=0.04
## predicted class=madis expected loss=0.75 P(node) =0.07843137
## class counts: 0 0 1 1 0 1 1
## probabilities: 0.000 0.000 0.250 0.250 0.000 0.250 0.250
## left son=26 (2 obs) right son=27 (2 obs)
## Primary splits:
## adopt < 5e-04 to the right, improve=1, (0 missing)
## advantag < 0.0015 to the left, improve=1, (0 missing)
## alon < 5e-04 to the left, improve=1, (0 missing)
## answer < 5e-04 to the right, improve=1, (0 missing)
## appoint < 0.0015 to the right, improve=1, (0 missing)
## Surrogate splits:
## advantag < 0.0015 to the left, agree=1, adj=1, (0 split)
## alon < 0.0015 to the left, agree=1, adj=1, (0 split)
## appoint < 0.0015 to the right, agree=1, adj=1, (0 split)
## assembl < 0.003 to the left, agree=1, adj=1, (0 split)
## avoid < 5e-04 to the right, agree=1, adj=1, (0 split)
##
## Node number 14: 9 observations
## predicted class=madis expected loss=0 P(node) =0.1764706
## class counts: 0 0 9 0 0 0 0
## probabilities: 0.000 0.000 1.000 0.000 0.000 0.000 0.000
##
## Node number 15: 1 observations
## predicted class=Madison_fed_44.txt expected loss=0 P(node) =0.01960784
## class counts: 0 0 0 0 1 0 0
## probabilities: 0.000 0.000 0.000 0.000 1.000 0.000 0.000
##
## Node number 26: 2 observations, complexity param=0.04
## predicted class=madis expected loss=0.5 P(node) =0.03921569
## class counts: 0 0 1 1 0 0 0
## probabilities: 0.000 0.000 0.500 0.500 0.000 0.000 0.000
## left son=52 (1 obs) right son=53 (1 obs)
## Primary splits:
## abl < 5e-04 to the right, improve=1, (0 missing)
## absolut < 5e-04 to the left, improve=1, (0 missing)
## act < 0.0025 to the right, improve=1, (0 missing)
## administr < 0.0025 to the right, improve=1, (0 missing)
## admit < 0.0015 to the left, improve=1, (0 missing)
##

```

```

## Node number 27: 2 observations,      complexity param=0.04
##   predicted class=Madison_fed_48.txt   expected loss=0.5   P(node) =0.03921569
##   class counts:      0      0      0      0      0      1      1
##   probabilities: 0.000 0.000 0.000 0.000 0.000 0.500 0.500
##   left son=54 (1 obs) right son=55 (1 obs)
##   Primary splits:
##     abl      < 0.001 to the left,  improve=1, (0 missing)
##     absolut < 0.001 to the left,  improve=1, (0 missing)
##     accord  < 0.001 to the right, improve=1, (0 missing)
##     act      < 0.0035 to the right, improve=1, (0 missing)
##     addit    < 0.0025 to the left, improve=1, (0 missing)
##
## Node number 52: 1 observations
##   predicted class=madis                expected loss=0   P(node) =0.01960784
##   class counts:      0      0      1      0      0      0      0
##   probabilities: 0.000 0.000 1.000 0.000 0.000 0.000 0.000
##
## Node number 53: 1 observations
##   predicted class=Madison_fed_42.txt   expected loss=0   P(node) =0.01960784
##   class counts:      0      0      0      1      0      0      0
##   probabilities: 0.000 0.000 0.000 1.000 0.000 0.000 0.000
##
## Node number 54: 1 observations
##   predicted class=Madison_fed_48.txt   expected loss=0   P(node) =0.01960784
##   class counts:      0      0      0      0      0      1      0
##   probabilities: 0.000 0.000 0.000 0.000 0.000 1.000 0.000
##
## Node number 55: 1 observations
##   predicted class=Madison_fed_58.txt   expected loss=0   P(node) =0.01960784
##   class counts:      0      0      0      0      0      0      1
##   probabilities: 0.000 0.000 0.000 0.000 0.000 0.000 1.000

```

```

#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)

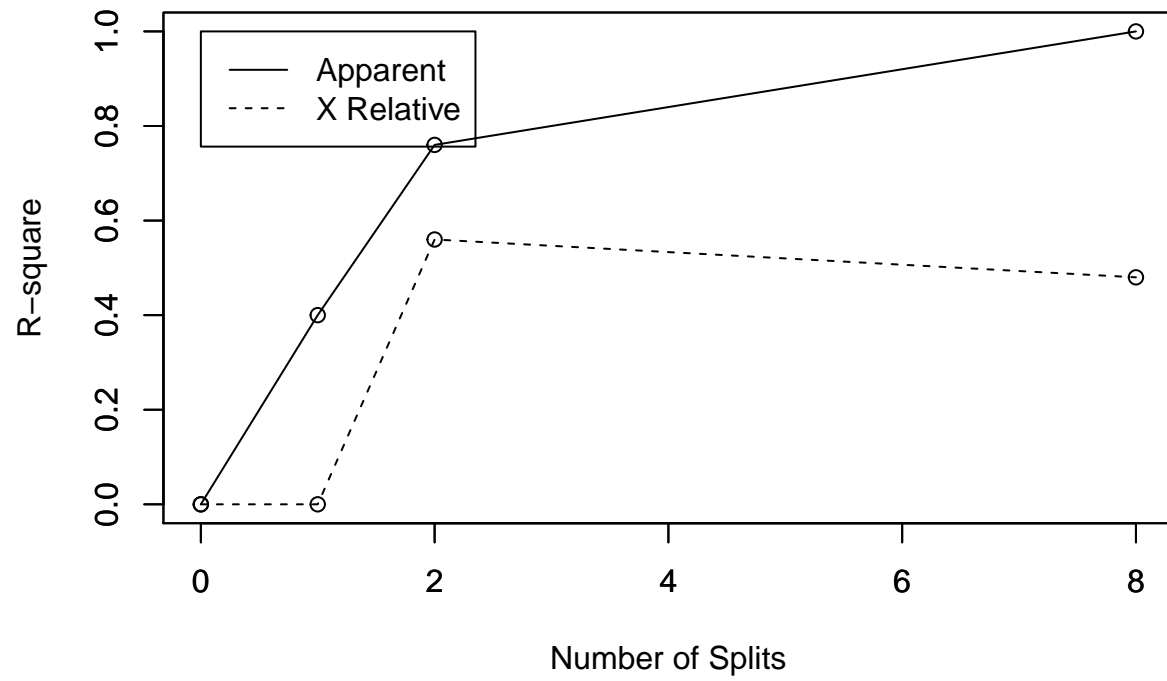
```

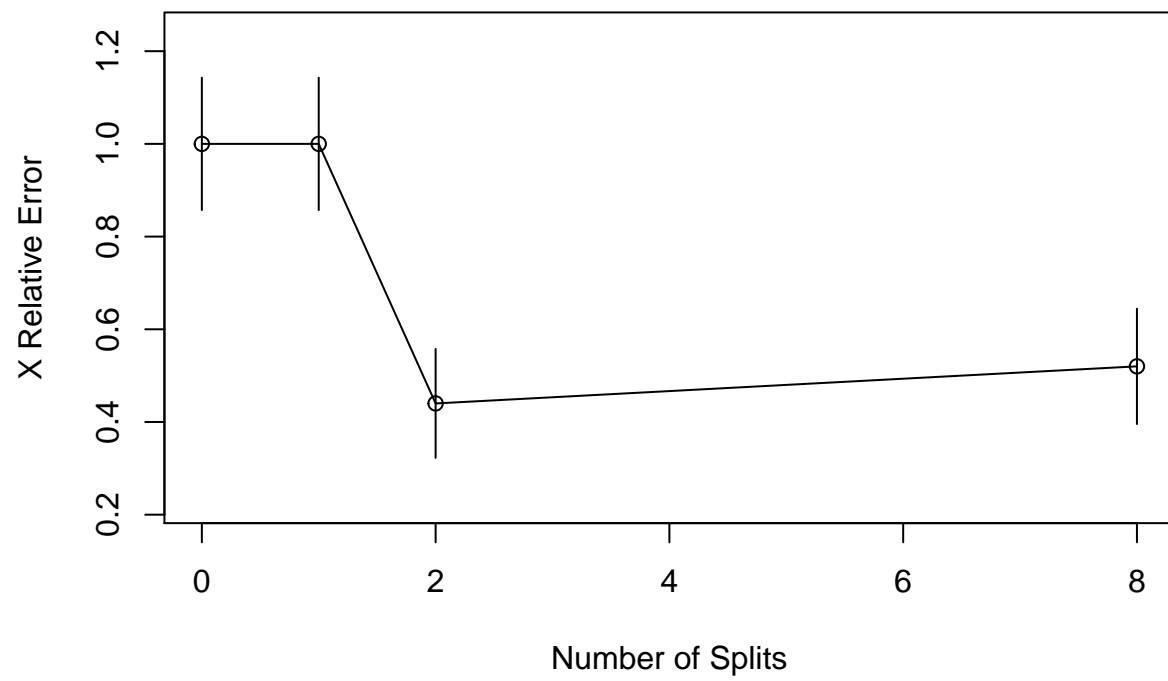
```

##
## 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] abl      adopt  alexand great  union  upon
##
## Root node error: 25/51 = 0.4902
##
## n= 51
##
##      CP nsplit rel error xerror   xstd
## 1 0.40      0      1.00   1.00 0.14280
## 2 0.36      1      0.60   1.00 0.14280
## 3 0.04      2      0.24   0.44 0.11749
## 4 0.00      8      0.00   0.52 0.12449

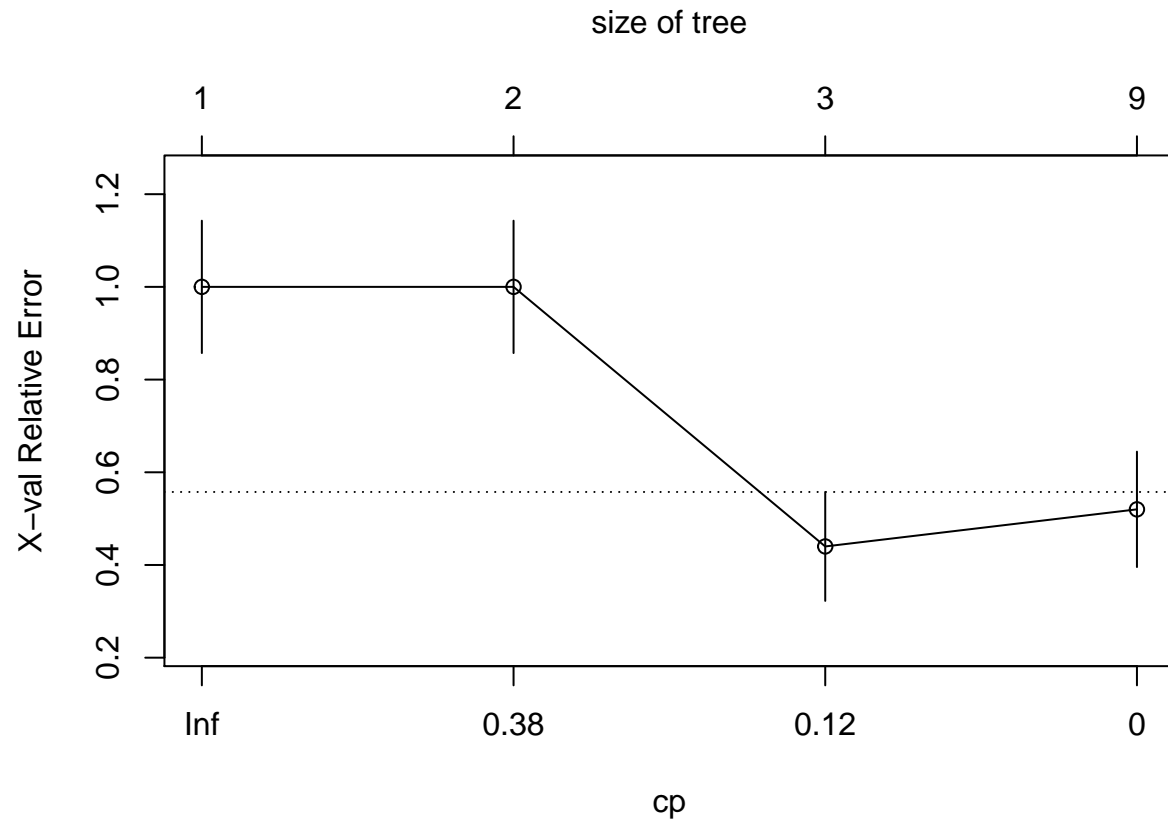
```

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

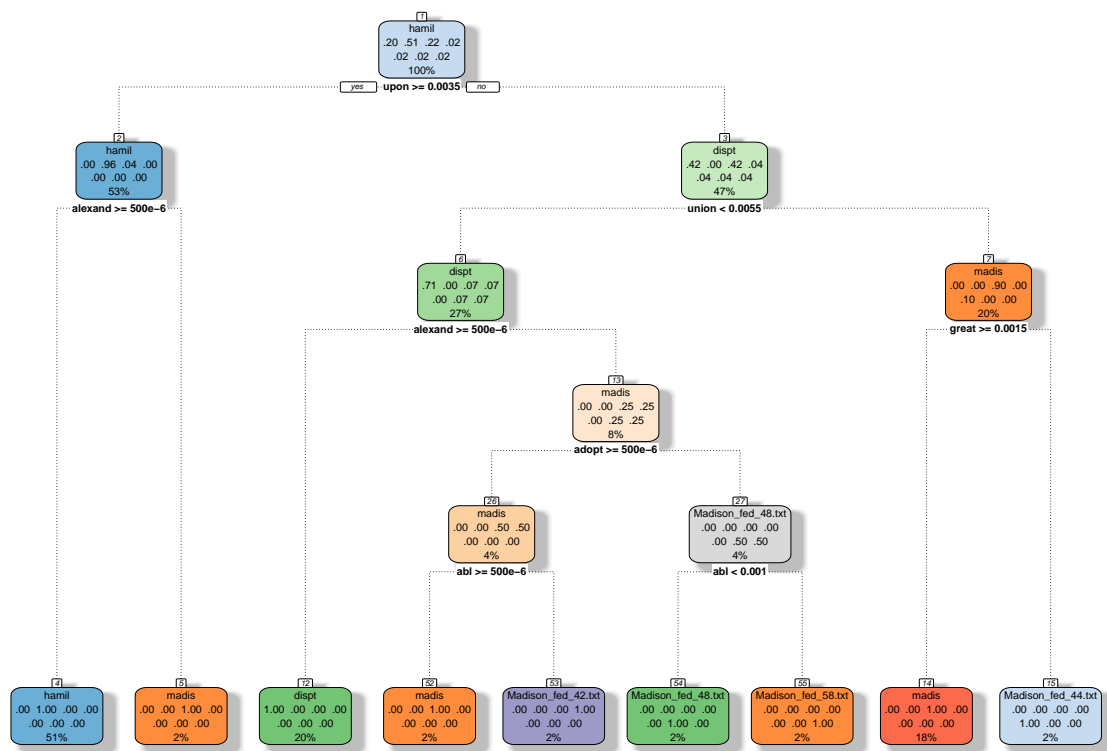




```
plotcp(train_tree2)
```



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



Rattle 2019-Oct-01 13:00:21 jerem

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

```
## true
## Authorship      dispt hamil madis Madison_fed_43.txt
## dispt           1     0     0                     0
## hamil           0    25     0                     0
## madis           0     0     2                     1
## Madison_fed_42.txt 0     0     0                     0
## Madison_fed_44.txt 0     0     1                     0
## Madison_fed_48.txt 0     0     0                     0
## Madison_fed_58.txt 0     0     1                     0
## true
## Authorship      Madison_fed_45.txt Madison_fed_46.txt
## dispt           0                     0
## hamil           0                     0
## madis           1                     0
## Madison_fed_42.txt 0                     0
## Madison_fed_44.txt 0                     0
## Madison_fed_48.txt 0                     0
## Madison_fed_58.txt 0                     1
## true
## Authorship      Madison_fed_47.txt
## dispt           0
## hamil           0
## madis           0
```


##	Madison_fed_42.txt	1
##	Madison_fed_44.txt	0
##	Madison_fed_48.txt	0
##	Madison_fed_58.txt	0