Stats425Project

Jimmy Le

3/7/2022

```
library(broom)
```

```
## Warning: package 'broom' was built under R version 4.1.2
```

```
library(dplyr)
```

```
##
## 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
```

```
library(ggplot2)
library(scales)
library(tm)
```

```
## Warning: package 'tm' was built under R version 4.1.2

## Loading required package: NLP

##
## Attaching package: 'NLP'

## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
library(topicmodels)
```

```
## Warning: package 'topicmodels' was built under R version 4.1.2
```

```
library(tidytext)
```

```
## Warning: package 'tidytext' was built under R version 4.1.2
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.1.2
```

```
library(pdftools)
```

```
## Warning: package 'pdftools' was built under R version 4.1.2

## Using poppler version 22.02.0
```

```r
library(wordcloud)
```

```
## Warning: package 'wordcloud' was built under R version 4.1.2

## Loading required package: RColorBrewer
```

```r
library(wordcloud2)
```

```
## Warning: package 'wordcloud2' was built under R version 4.1.2
```

```r
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 4.1.2

##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

```r
library(forcats)
```

```
## Warning: package 'forcats' was built under R version 4.1.2
```

```r
library(quanteda)
```

```
## Warning: package 'quanteda' was built under R version 4.1.2

## Package version: 3.2.0
## Unicode version: 13.0
## ICU version: 69.1

## Parallel computing: 16 of 16 threads used.

## See https://quanteda.io for tutorials and examples.

##
## Attaching package: 'quanteda'

## The following object is masked from 'package:tm':
##
##     stopwords

## The following objects are masked from 'package:NLP':
##
##     meta, meta<-
```

```r
files <- list.files("./Stats425Project", pattern = "pdf$")
setwd("./Stats425Project")
Corp <- Corpus(URISource(files, mode = "text"), readerControl = list(reader = readPDF))
inspect(Corp)
```

```
## <<VCorpus>>
## Metadata:  corpus specific: 0, document level (indexed): 0
## Content:  documents: 4
##
## [[1]]
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 113724
##
```

```
## [[2]]
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 43856
##
## [[3]]
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 30443
##
## [[4]]
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 169308
```

```
corp <- corpus(Corp)
```

```
exclude <- c("shall", "thee", "thy", "thus", "will", "come",
             "know", "may", "upon", "hath", "now", "well", "make",
             "let", "see", "tell", "yet", "like", "put", "speak",
             "give", "speak", "can", "comes", "makes", "sees", "tells",
             "likes", "puts", "speaks", "gives", "speaks", "knows",
             "say", "says", "take", "takes", "exeunt", "though", "hear",
             "think", "hears", "thinks", "listen", "listens", "hear",
             "hears", "follow" ,"commercially" ,"commercial" , "readable",
             "personal", "doth", "membership", "stand", "therefore",
             "complete", "tis", "electronic", "prohibited", "must",
             "look", "looks", "call", "calls", "done", "prove", "whose",
             "enter", "one", "words", "thou", "came", "much", "never",
             "wit", "leave", "even", "ever", "distributed" , "keep",
             "stay", "made", "scene", "many", "away", "exit", "shalt","http", "homepage",  "shakespeare
```

```
print("Simple Transformation")
```

```
## [1] "Simple Transformation"
```

```
Corp.simple <-tm_map(Corp, content_transformer(function(x, pattern) gsub(pattern, " ", x)) , "/|@|\\|")
Corp.simple[[1]]
```

```
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 113724
```

```
print("Conversion to Lower Case")
```

```
## [1] "Conversion to Lower Case"
```

```
Corp.lower <- tm_map(Corp.simple, content_transformer(tolower))
Corp.lower[[1]]
```

```
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 113724
```

```
print("Remove Numbers")
```

```
## [1] "Remove Numbers"
```

```
Corp.number <- tm_map(Corp.lower, removeNumbers)
Corp.number[[1]]
```

```
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 113724
```

```
print("Remove Punctuation")
```

```
## [1] "Remove Punctuation"
```

```
Corp.punct <- tm_map(Corp.number, removePunctuation)
Corp.punct[[1]]
```

```
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 108070
```

```
print("Remove English Stop Words")
```

```
## [1] "Remove English Stop Words"
```

```
Corp.EngStop <- tm_map(Corp.punct, removeWords, stopwords("english"))
Corp.EngStop[[1]]
```

```
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 84531
```

```
print("Remove Own Stop Words")
```

```
## [1] "Remove Own Stop Words"
```

```
Corp.MyStop <- tm_map(Corp.EngStop, removeWords, exclude)
Corp.MyStop[[1]]
```

```
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 78197
```

```
print("Strip Whitespace")
```

```
## [1] "Strip Whitespace"
```

```
Corp.WhiteSpace <- tm_map(Corp.MyStop, stripWhitespace)
Corp.WhiteSpace[[1]]
```

```
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 56249
```

```
print("Specific Transformation")
```

```
## [1] "Specific Transformation"
```

```
toString <- content_transformer(function(x, from, to) gsub(from, to, x))
Corp.SpecialTransformation <- tm_map(Corp.WhiteSpace, toString, "©", " ")
Corp.SpecialTransformation[[1]]
```

```
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 56249
```

```
print("Stemming")
```

```
## [1] "Stemming"
```

```
Corp.stem <- tm_map(Corp.SpecialTransformation, stemDocument)
Corp.stem[[1]]
```

```
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 51906
```

```
#inspect(Corp.stem[[4]])
#Corp.stem[[4]]$content[2]
length(Corp.stem[[4]]$content) #number of pages
```

```
## [1] 103
```

```
dtm <- DocumentTermMatrix(Corp.stem)
inspect(dtm)
```

```
## <<DocumentTermMatrix (documents: 4, terms: 3298)>>
## Non-/sparse entries: 7702/5490
## Sparsity           : 42%
## Maximal term length: 18
## Weighting          : term frequency (tf)
## Sample             :
##                              Terms
## Docs                           banquo good hand king ladi macbeth macduff murder
##   Macbeth Original Play.pdf      76   54   36   45   96     287     107     57
##   Macbeth1948.pdf                13   27   22   27    1      26      12     12
##   Macbeth2015.pdf                12   20   18   24    7      44      20     16
##   Macbeth2020.pdf                84   34   77   43  199     488     123     61
##                              Terms
## Docs                           ross time
##   Macbeth Original Play.pdf      53   50
##   Macbeth1948.pdf                 0   27
##   Macbeth2015.pdf                 1   18
##   Macbeth2020.pdf                87   34
```

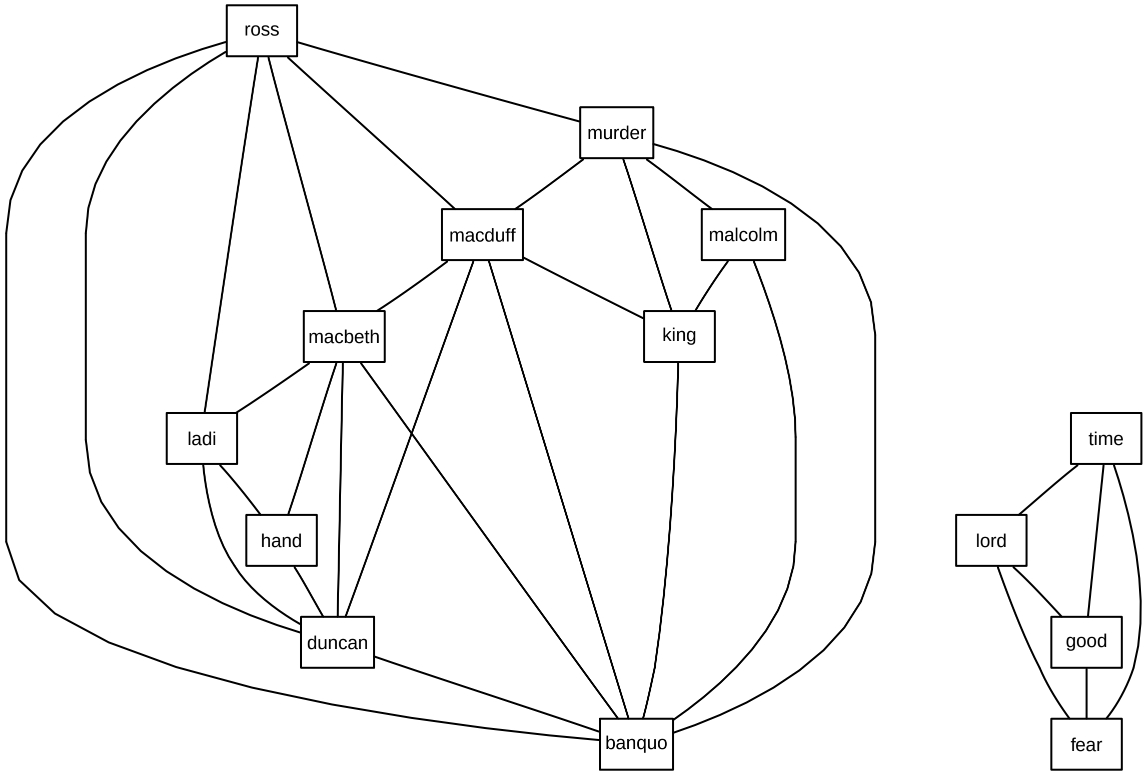
```
ft <-findFreqTerms(dtm,lowfreq =  110)
ft
```

```
##  [1] "banquo"  "duncan"  "fear"    "good"    "hand"    "king"    "ladi"
##  [8] "lord"    "macbeth" "macduff" "malcolm" "murder"  "ross"    "time"
```

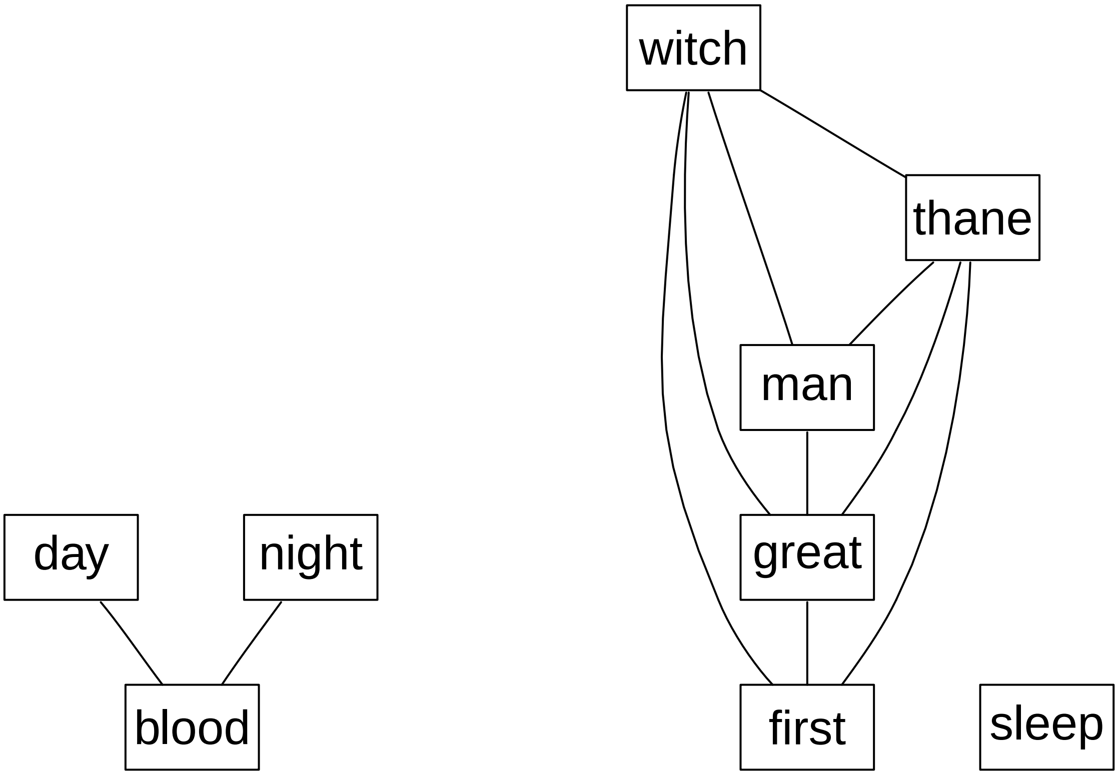
```
mft <- findFreqTerms(dtm,lowfreq = 80, highfreq =  110)
mft
```

```
## [1] "blood" "day"   "first" "great" "man"   "night" "sleep" "thane" "witch"
```

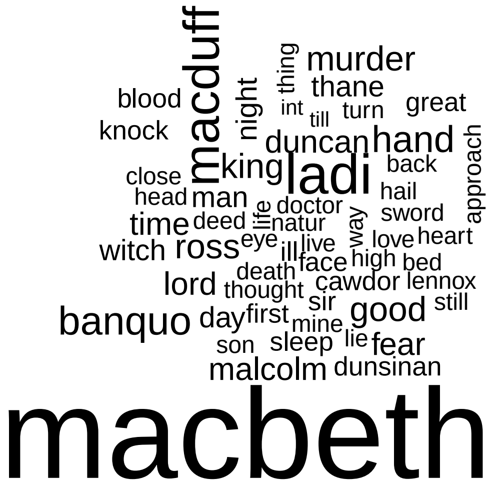
```
plot(dtm, terms = ft, corThreshold = 0.95)
```



```r
plot(dtm, terms = mft, corThreshold = 0.95)
```



```
wordcloud(Corp.stem, min.freq = 50)
```

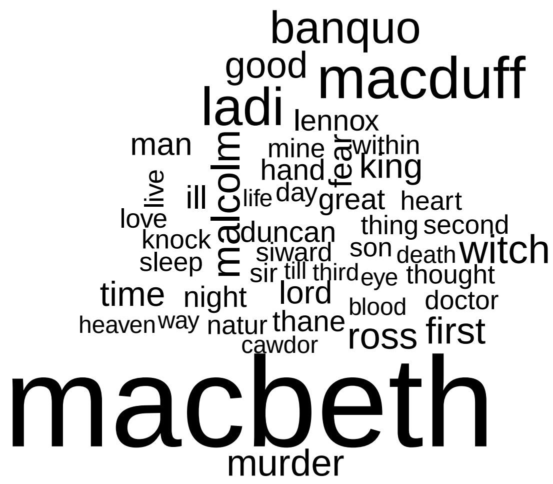


```
#wordcloud2(findMostFreqTerms(dtm, 50))
```

```
# a <-findMostFreqTerms(dtm, 50)
# a[[1]]
# data.frame(a)
wordcloud(Corp.stem[[1]]$content, min.freq = 20) #Original Play
```

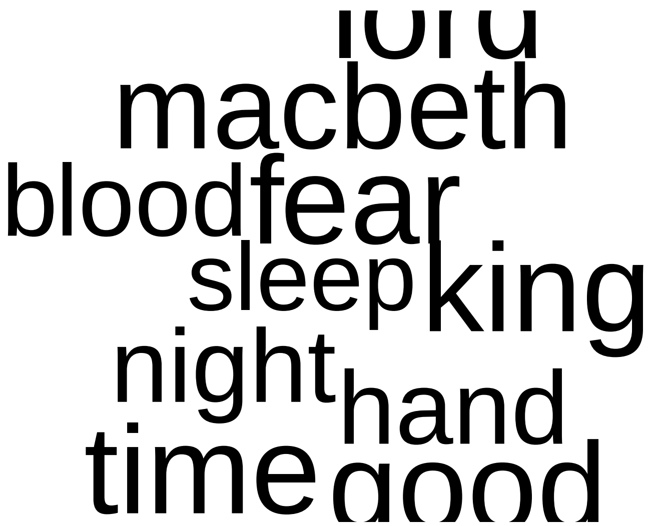
```
## Warning in tm_map.SimpleCorpus(corpus, tm::removePunctuation): transformation
## drops documents
```

```
## Warning in tm_map.SimpleCorpus(corpus, function(x) tm::removeWords(x,
## tm::stopwords())): transformation drops documents
```



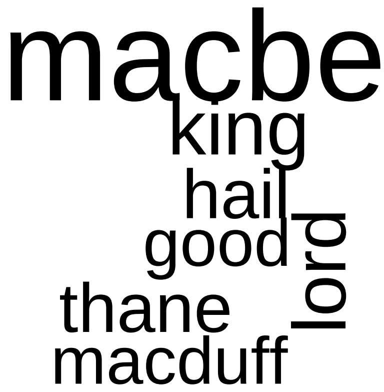
```r
wordcloud(Corp.stem[[2]]$content, min.freq = 20) #1948
```

```
## Warning in tm_map.SimpleCorpus(corpus, tm::removePunctuation): transformation
## drops documents

## Warning in tm_map.SimpleCorpus(corpus, tm::removePunctuation): transformation
## drops documents
```



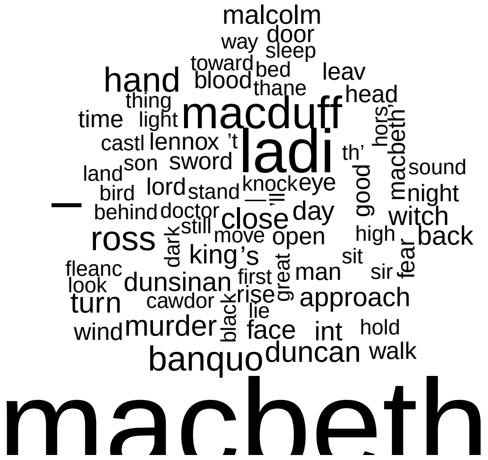
```
wordcloud(Corp.stem[[3]]$content, min.freq = 20) #2015
```

```
## Warning in tm_map.SimpleCorpus(corpus, tm::removePunctuation): transformation
## drops documents

## Warning in tm_map.SimpleCorpus(corpus, tm::removePunctuation): transformation
## drops documents
```



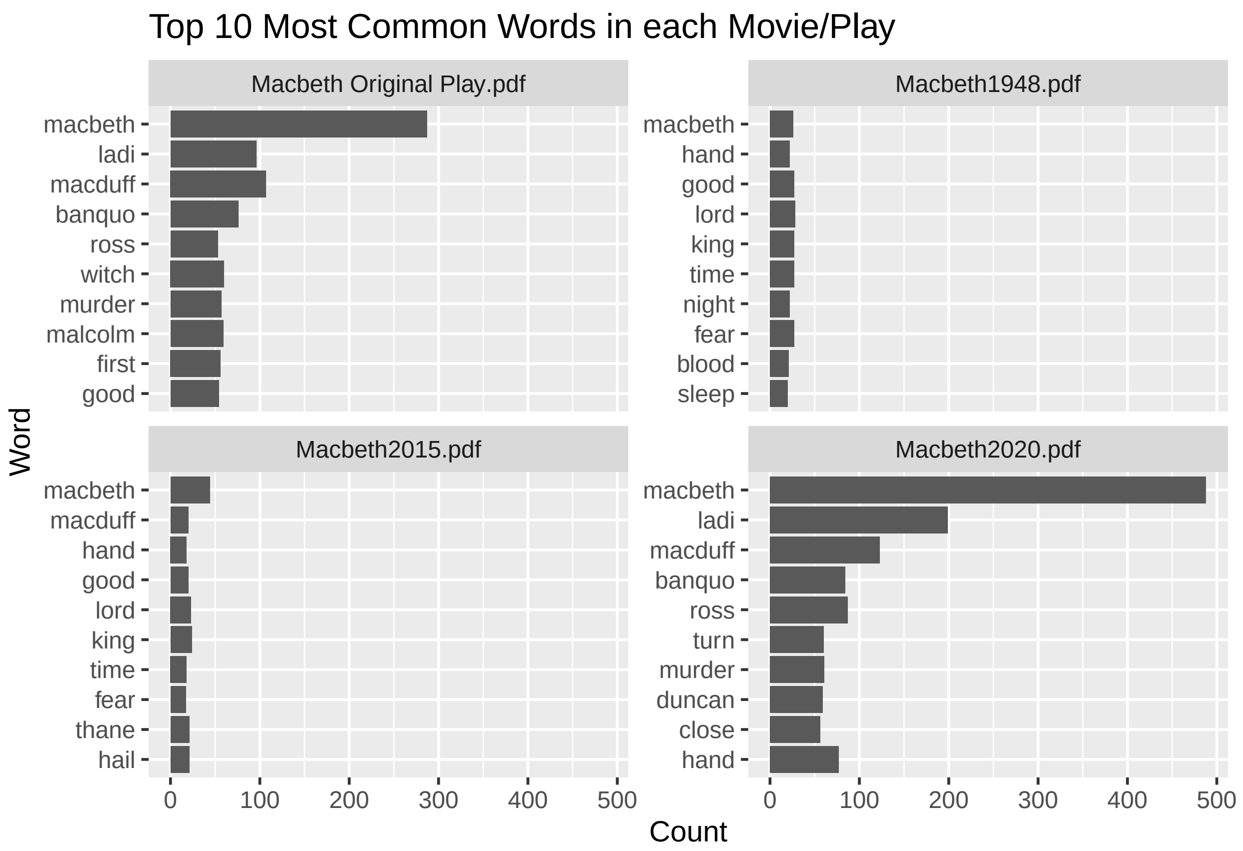
```
wordcloud(Corp.stem[[4]]$content, min.freq = 20) #2020
```

```
## Warning in tm_map.SimpleCorpus(corpus, tm::removePunctuation): transformation
## drops documents

## Warning in tm_map.SimpleCorpus(corpus, tm::removePunctuation): transformation
## drops documents
```

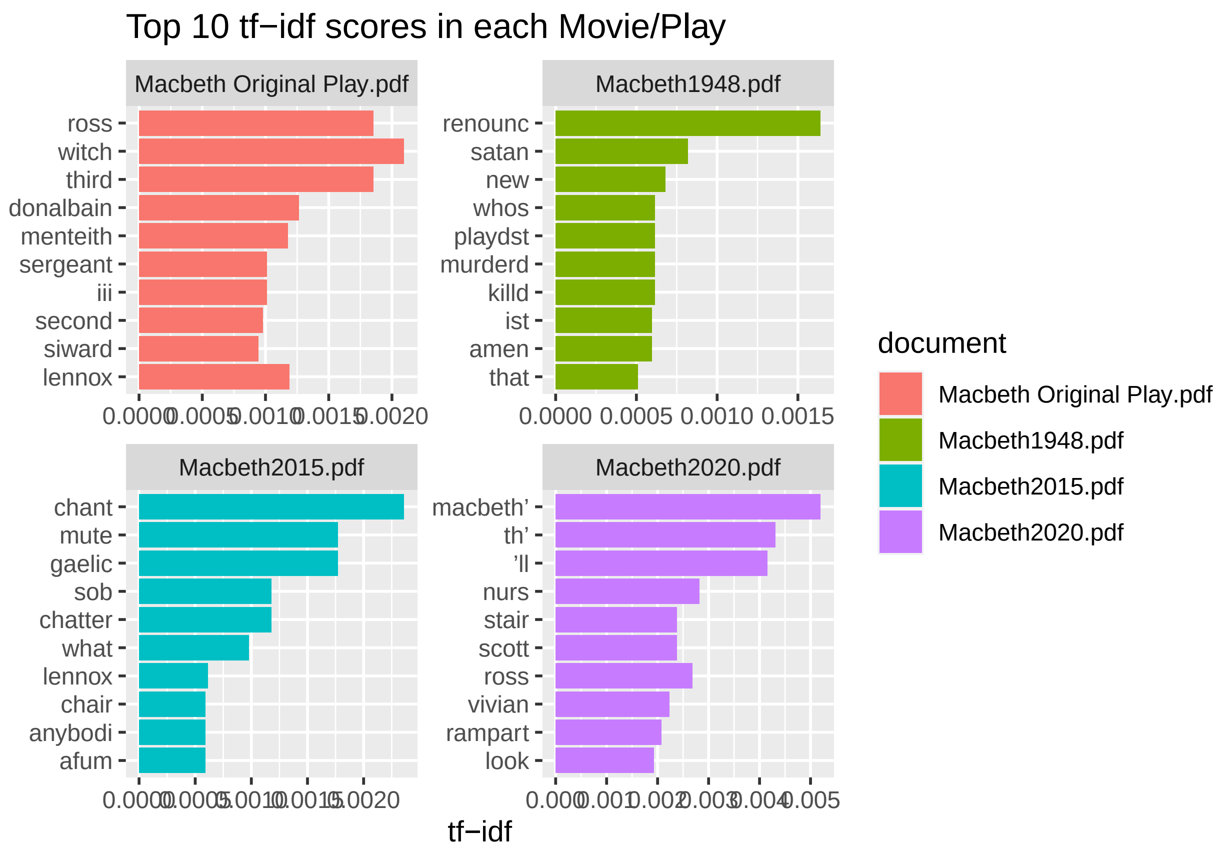


```
Tidydf <- tidy(dtm)
top10words_each_doc <-Tidydf %>% group_by(document) %>% arrange(desc(count), .by_group = TRUE) %>% top_
```

```
ggplot(top10words_each_doc, aes(count, reorder(term,count))) + geom_col() + facet_wrap(~document, ncol
```

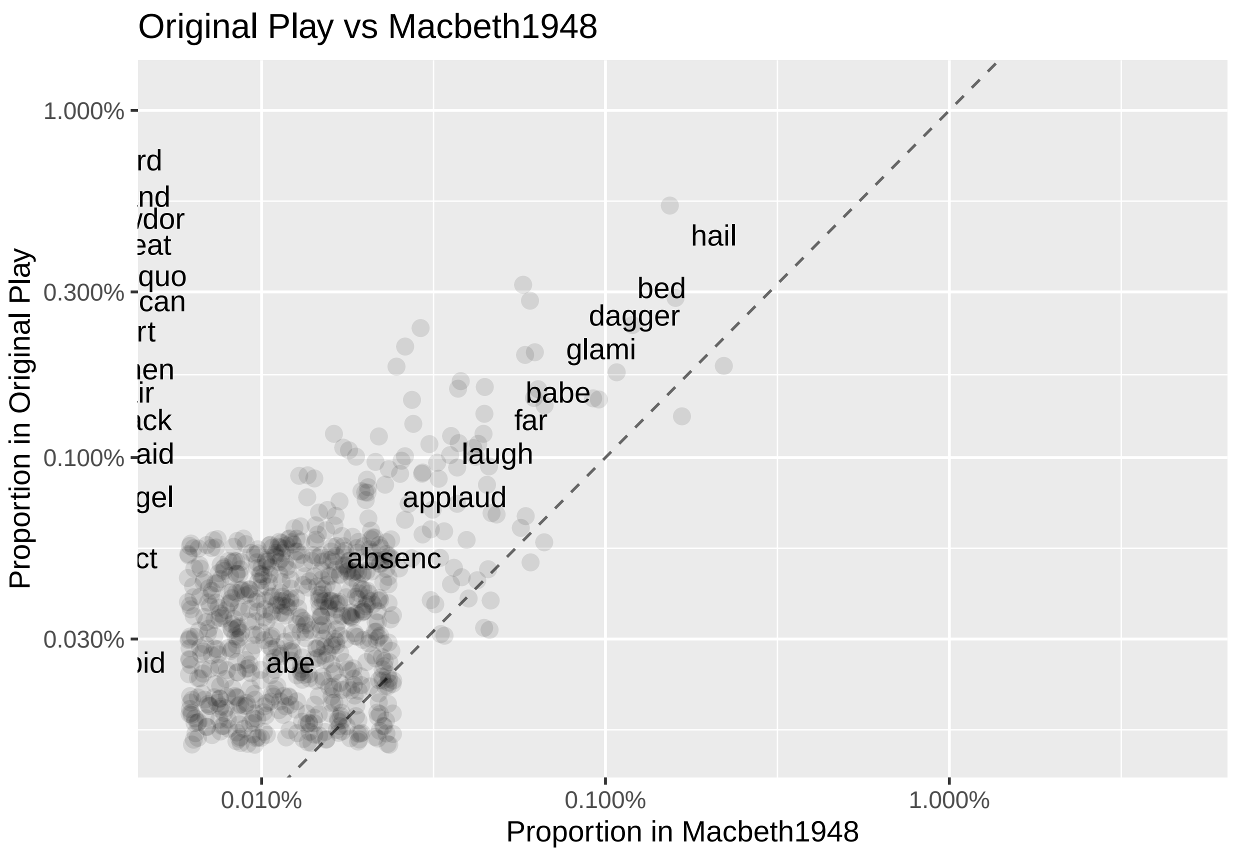


```
Tidydf %>% group_by(document) %>% bind_tf_idf(term,document,count) %>% arrange(desc(tf_idf), .by_group =
```



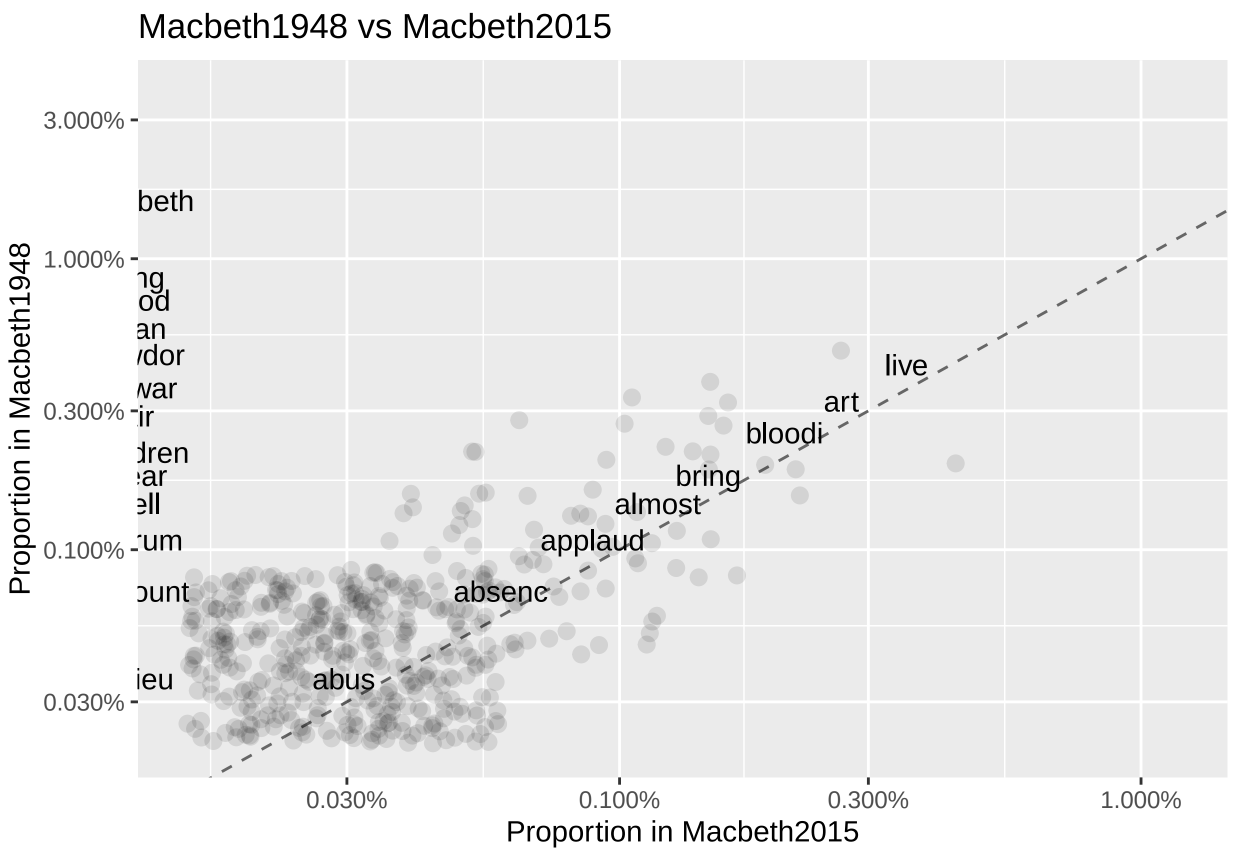
```r
corfreq <-Tidydf %>% group_by(document)  %>% mutate(proportion = count/sum(count),) %>% spread(document
corfreq12 <- corfreq %>% filter(!is.na(`Macbeth Original Play.pdf`)) %>% filter(!is.na(Macbeth1948.pdf)
corfreqlast2 <- corfreq %>% filter(!is.na(Macbeth2015.pdf)) %>% filter(!is.na(Macbeth2020.pdf))
ggplot(corfreq, aes(x = `Macbeth Original Play.pdf`, y = Macbeth1948.pdf)) +
  geom_abline(color = "gray40", lty = 2) +
  geom_jitter(alpha = 0.1, size = 2.5, width = 0.3, height = 0.3) +
  geom_text(aes(label = term), check_overlap = TRUE, vjust = 1.5) +
  scale_x_log10(labels = percent_format()) +
  scale_y_log10(labels = percent_format()) +
  theme(legend.position="none") +
  labs(y = "Proportion in Original Play", x = "Proportion in Macbeth1948")+
  ggtitle("Original Play vs Macbeth1948")
```

```
## Warning: Transformation introduced infinite values in continuous x-axis

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous x-axis

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 4471 rows containing missing values (geom_point).
```



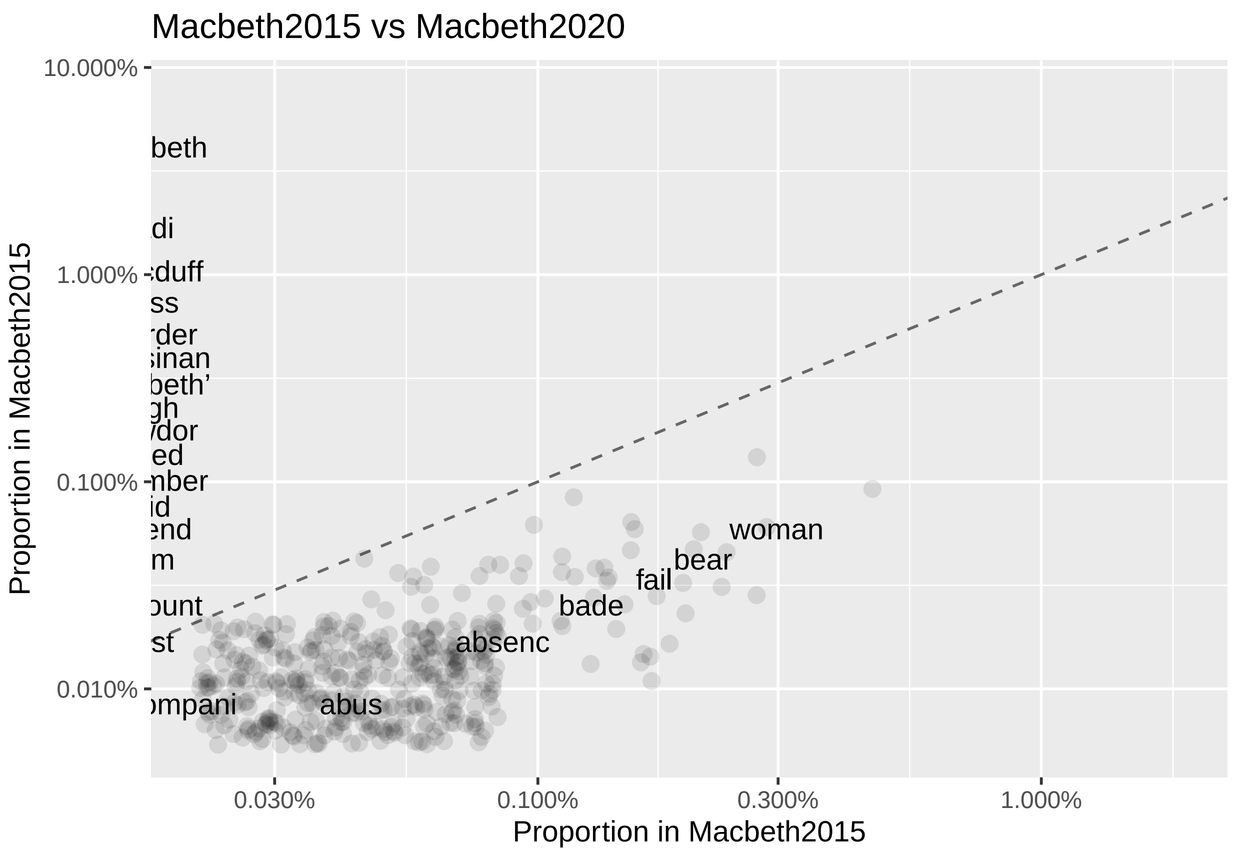
```
ggplot(corfreq, aes(x = Macbeth1948.pdf, y = Macbeth2015.pdf)) +
  geom_abline(color = "gray40", lty = 2) +
  geom_jitter(alpha = 0.1, size = 2.5, width = 0.3, height = 0.3) +
  geom_text(aes(label = term), check_overlap = TRUE, vjust = 1.5) +
  scale_x_log10(labels = percent_format()) +
  scale_y_log10(labels = percent_format()) +
  theme(legend.position="none") +
  labs(y = "Proportion in Macbeth1948", x = "Proportion in Macbeth2015")+
  ggtitle("Macbeth1948 vs Macbeth2015")
```

```
## Warning: Transformation introduced infinite values in continuous x-axis

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous x-axis

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 4736 rows containing missing values (geom_point).
```



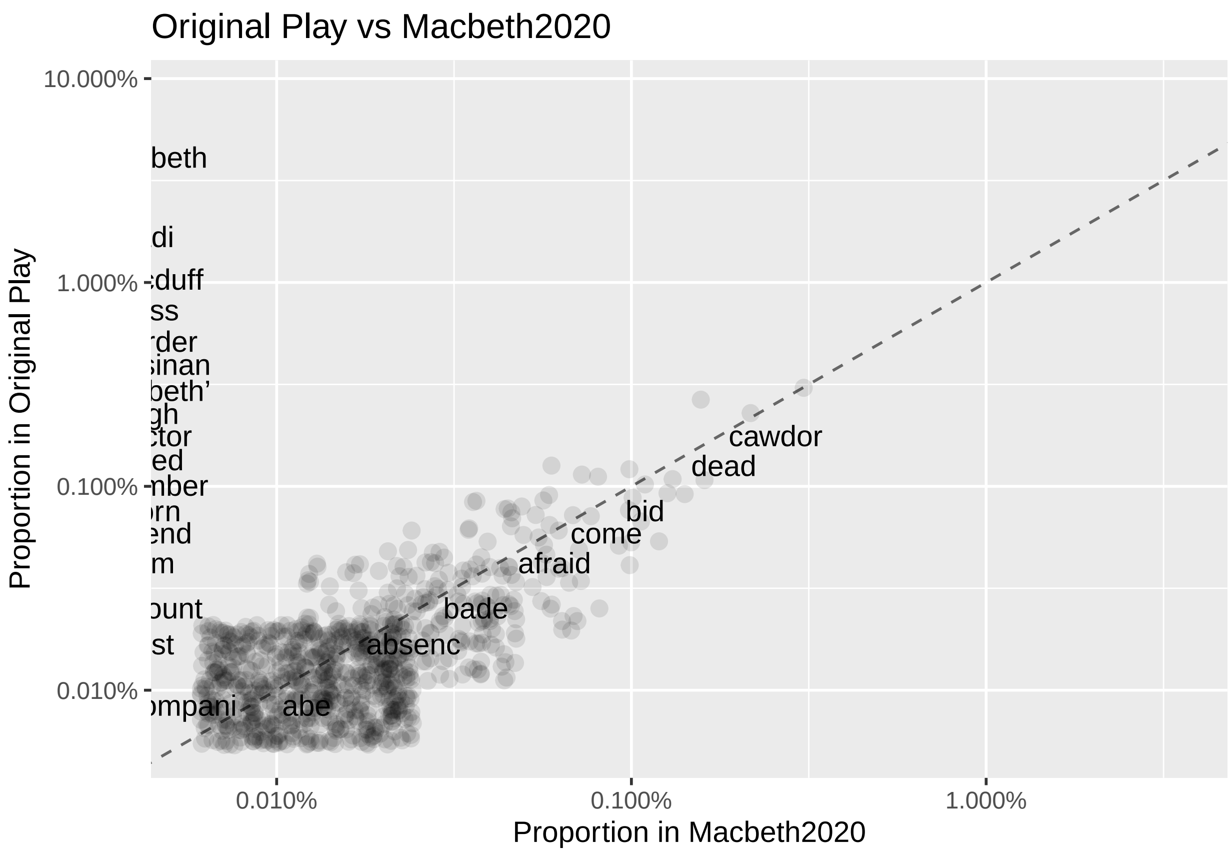
```
ggplot(corfreq, aes(x = Macbeth2015.pdf, y = Macbeth2020.pdf)) +
  geom_abline(color = "gray40", lty = 2) +
  geom_jitter(alpha = 0.1, size = 2.5, width = 0.3, height = 0.3) +
  geom_text(aes(label = term), check_overlap = TRUE, vjust = 1.5) +
  scale_x_log10(labels = percent_format()) +
  scale_y_log10(labels = percent_format()) +
  theme(legend.position="none") +
  labs(y = "Proportion in Macbeth2015", x = "Proportion in Macbeth2015")+
  ggtitle("Macbeth2015 vs Macbeth2020")
```

```
## Warning: Transformation introduced infinite values in continuous x-axis

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous x-axis

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 4784 rows containing missing values (geom_point).
```



```
ggplot(corfreq, aes(x = `Macbeth Original Play.pdf`, y = Macbeth2020.pdf)) +
  geom_abline(color = "gray40", lty = 2) +
  geom_jitter(alpha = 0.1, size = 2.5, width = 0.3, height = 0.3) +
  geom_text(aes(label = term), check_overlap = TRUE, vjust = 1.5) +
  scale_x_log10(labels = percent_format()) +
  scale_y_log10(labels = percent_format()) +
  theme(legend.position="none") +
  labs(y = "Proportion in Original Play", x = "Proportion in Macbeth2020")+
  ggtitle("Original Play vs Macbeth2020")
```

```
## Warning: Transformation introduced infinite values in continuous x-axis

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous x-axis

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 4319 rows containing missing values (geom_point).
```



```
ab <-cor.test(corfreq$`Macbeth Original Play.pdf`, corfreq$Macbeth1948.pdf)
bc <-cor.test(corfreq$Macbeth1948.pdf, corfreq$Macbeth2015.pdf)
cd <-cor.test(corfreq$Macbeth2015.pdf, corfreq$Macbeth2020.pdf)
ac <-cor.test(corfreq$`Macbeth Original Play.pdf`, corfreq$Macbeth2015.pdf)
ae <-cor.test(corfreq$`Macbeth Original Play.pdf`, corfreq$Macbeth2020.pdf)
cat("P-val is:", ab$p.value,"\n","r = ",unname(ab[["estimate"]]))
```

```
## P-val is: 4.635013e-05
##  r =  -0.05643346
```

```
cat("P-val is:", bc$p.value,"\n","r = ",unname(bc[["estimate"]]))
```

```
## P-val is: 0.378234
##  r =  -0.0122172
```

```
cat("P-val is:", cd$p.value,"\n","r = ",unname(cd[["estimate"]]))
```

```
## P-val is: 0.0006429174
##  r =  -0.04729415
```

```
cat("P-val is:", ac$p.value,"\n","r = ",unname(ac[["estimate"]]))
```

```
## P-val is: 7.661818e-06
##  r =  -0.06198047
```

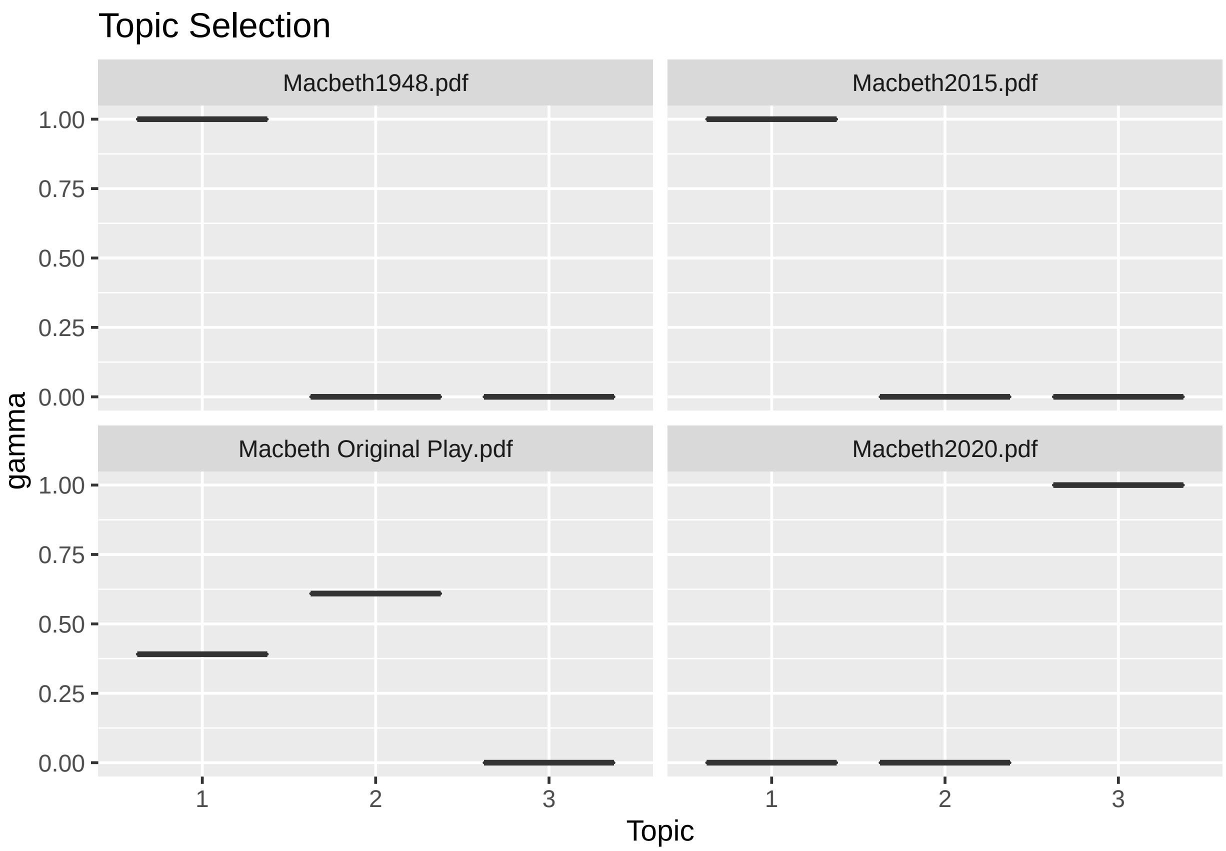
```
cat("P-val is:", ae$p.value,"\n","r = ",unname(ae[["estimate"]]))
```

```
## P-val is: 0.00492625
##  r =  -0.03897245
```

```r
LDA.model <- LDA(dtm, k = 3, control = list(seed = 1128))
LDA.tidy <- tidy(LDA.model, matrix = "beta")
LDA.tidy <- LDA.tidy %>% group_by(topic)
sortedLDA <-arrange(LDA.tidy, desc(beta), .by_group = TRUE)
sortedLDA %>% top_n(10, beta)
```

```
## # A tibble: 30 x 3
## # Groups:   topic [3]
##    topic term       beta
##    <int> <chr>     <dbl>
##  1     1 macbeth 0.0121
##  2     1 king    0.00921
##  3     1 lord    0.00886
##  4     1 good    0.00831
##  5     1 time    0.00790
##  6     1 fear    0.00788
##  7     1 thane   0.00703
##  8     1 hand    0.00686
##  9     1 man     0.00610
## 10     1 hail    0.00605
## # ... with 20 more rows
```

```r
LDA.tidy2 <- tidy(LDA.model, matrix = "gamma")
LDA.tidy2 <-LDA.tidy2 %>% mutate(document = reorder(document, gamma * topic))
ggplot(LDA.tidy2, aes(factor(topic), gamma)) + geom_boxplot() + facet_wrap(~ document)  + ggtitle("Topi
```



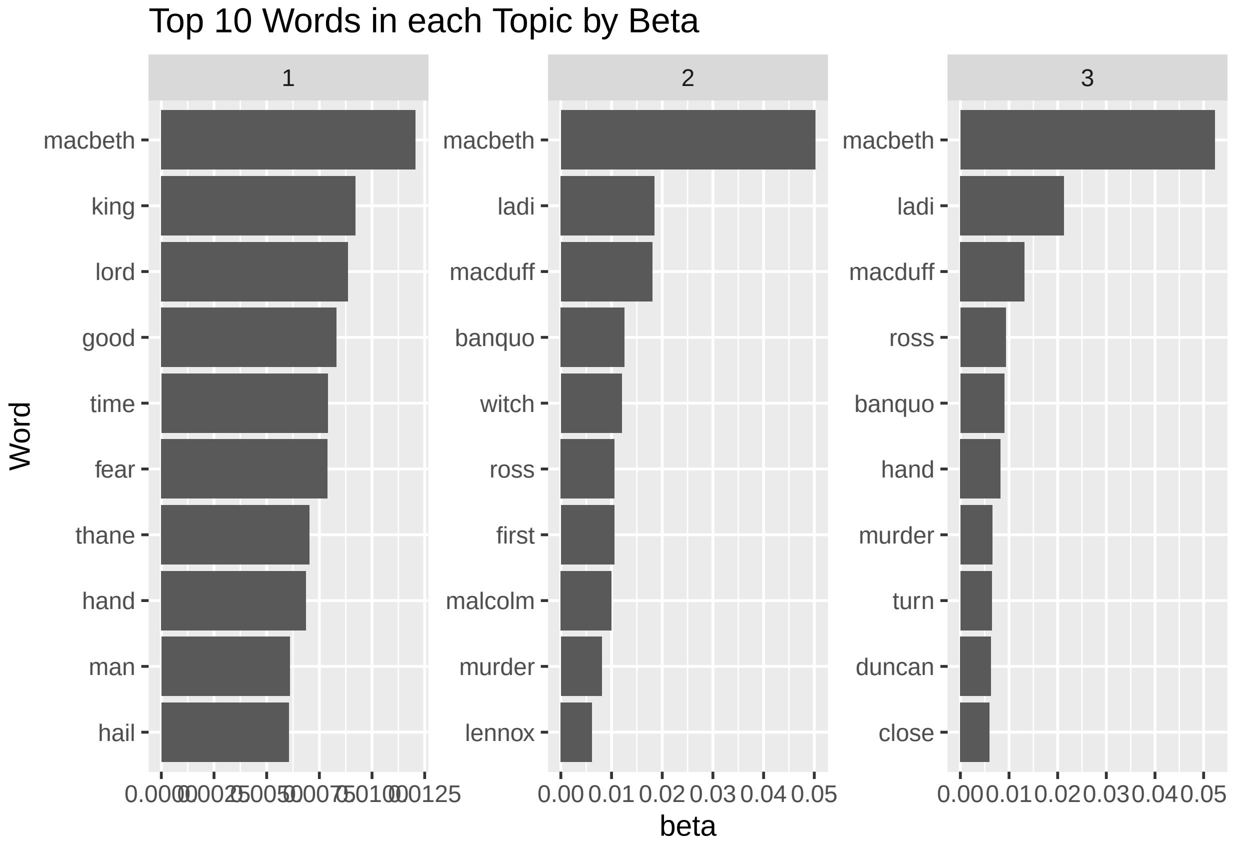
```
top10LDA <- top_n(sortedLDA, 10)
```

```
## Selecting by beta
```

```
top10LDA
```

```
## # A tibble: 30 x 3
## # Groups:   topic [3]
##    topic term       beta
##    <int> <chr>     <dbl>
##  1     1 macbeth 0.0121
##  2     1 king    0.00921
##  3     1 lord    0.00886
##  4     1 good    0.00831
##  5     1 time    0.00790
##  6     1 fear    0.00788
##  7     1 thane   0.00703
##  8     1 hand    0.00686
##  9     1 man     0.00610
## 10     1 hail    0.00605
## # ... with 20 more rows
```

```
ggplot(top10LDA, aes(reorder_within(term, beta, topic),beta)) + geom_col(show.legend = FALSE) +facet_wra
"free") +coord_flip() +scale_x_reordered() + ggtitle("Top 10 Words in each Topic by Beta") + xlab("Word
```



```
classification <- LDA.tidy %>% group_by(term) %>% top_n(1, beta) %>% ungroup()
#classification #What model thinks the chapter belongs to which topic
```

```
Missclassification <- LDA.tidy %>% group_by(term) %>% top_n(2, beta) %>% slice_min(n=1,beta) %>% transmu
classification %>% inner_join(Missclassification, by = "term")
```

```
## # A tibble: 3,298 x 4
##    topic term           beta `Incorrect Prediction`
##    <int> <chr>         <dbl>                  <int>
##  1     3 -accompani 0.000107                      2
##  2     3 -appar     0.000107                      2
##  3     3 -two-      0.000107                      2
##  4     3 ‘em        0.000536                      2
##  5     3 ‘gainst    0.000107                      2
##  6     3 ‘hail      0.000107                      2
##  7     3 ‘twere     0.000428                      2
##  8     3 ‘twixt     0.000107                      2
##  9     3 ‘twould    0.000107                      2
## 10     3 ’dst       0.000214                      2
## # ... with 3,288 more rows
```

```
assignments <- augment(LDA.model, data = Tidydf)
assignments
```

```
## # A tibble: 7,702 x 4
##    document                  term    count .topic
##    <chr>                     <chr>   <dbl>  <dbl>
##  1 Macbeth Original Play.pdf abe         1      1
##  2 Macbeth Original Play.pdf abhor       1      2
##  3 Macbeth Original Play.pdf abid        2      2
##  4 Macbeth Original Play.pdf abjur       1      2
##  5 Macbeth Original Play.pdf abound      1      2
##  6 Macbeth Original Play.pdf abroad      2      2
##  7 Macbeth Original Play.pdf absenc      2      1
##  8 Macbeth Original Play.pdf absent      1      2
##  9 Macbeth Original Play.pdf absolut     3      2
## 10 Macbeth Original Play.pdf abus        1      1
## # ... with 7,692 more rows
```

```
missclassifiedterms <-assignments %>%  left_join(Missclassification) %>%group_by(term) %>% rename("Predi
```

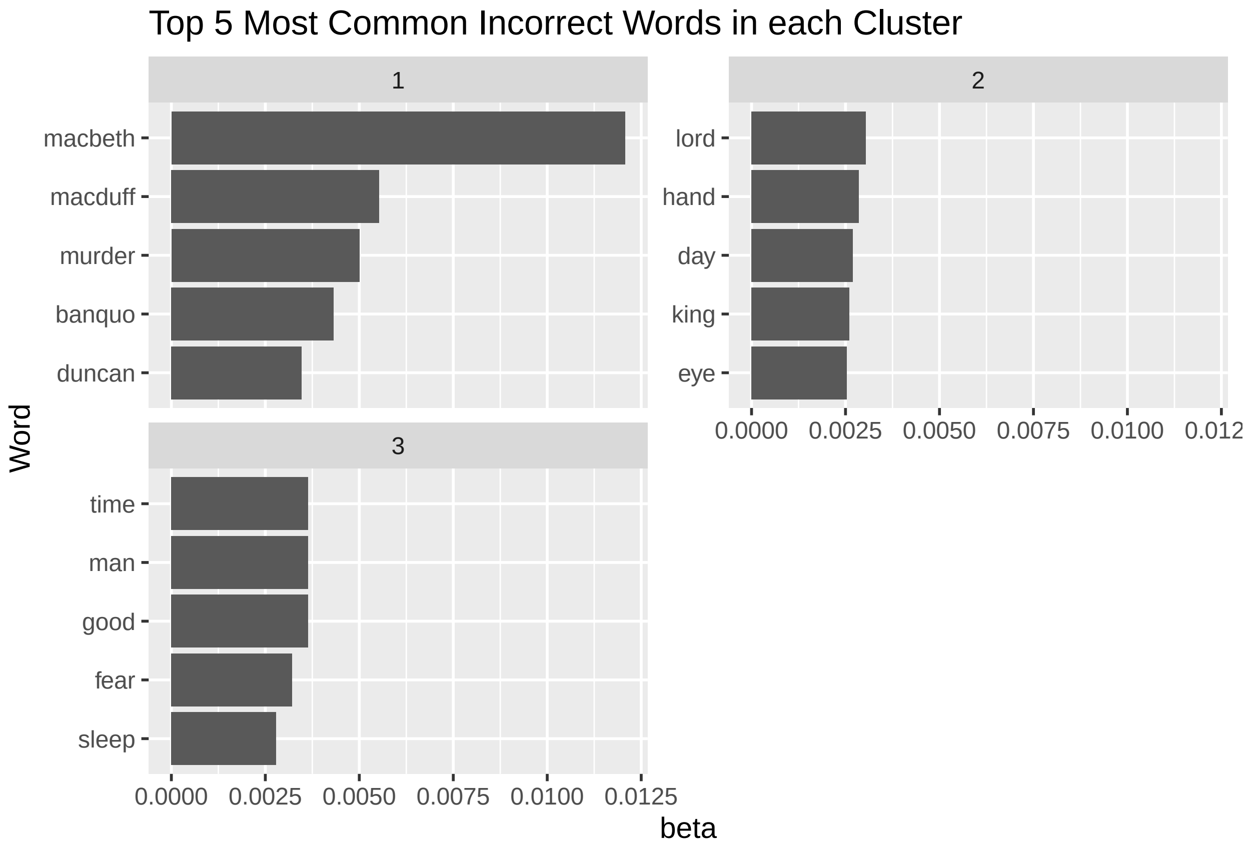
```
## Joining, by = "term"
```

```
#%>% mutate(percent=count/sum(count)) %>% filter(term != consensus)
# %>%
# ggplot(aes(consensus, term, fill = percent)) +geom_tile() +
# scale_fill_gradient2(high = "red", label = percent_format()) +theme_minimal() +
# theme(axis.text.x = element_text(angle = 90, hjust = 1),
# panel.grid = element_blank()) +
# labs(x = "Document words were assigned to",y = "Book words came from",fill = "% of
# assignments")
missclassifiedterms
```

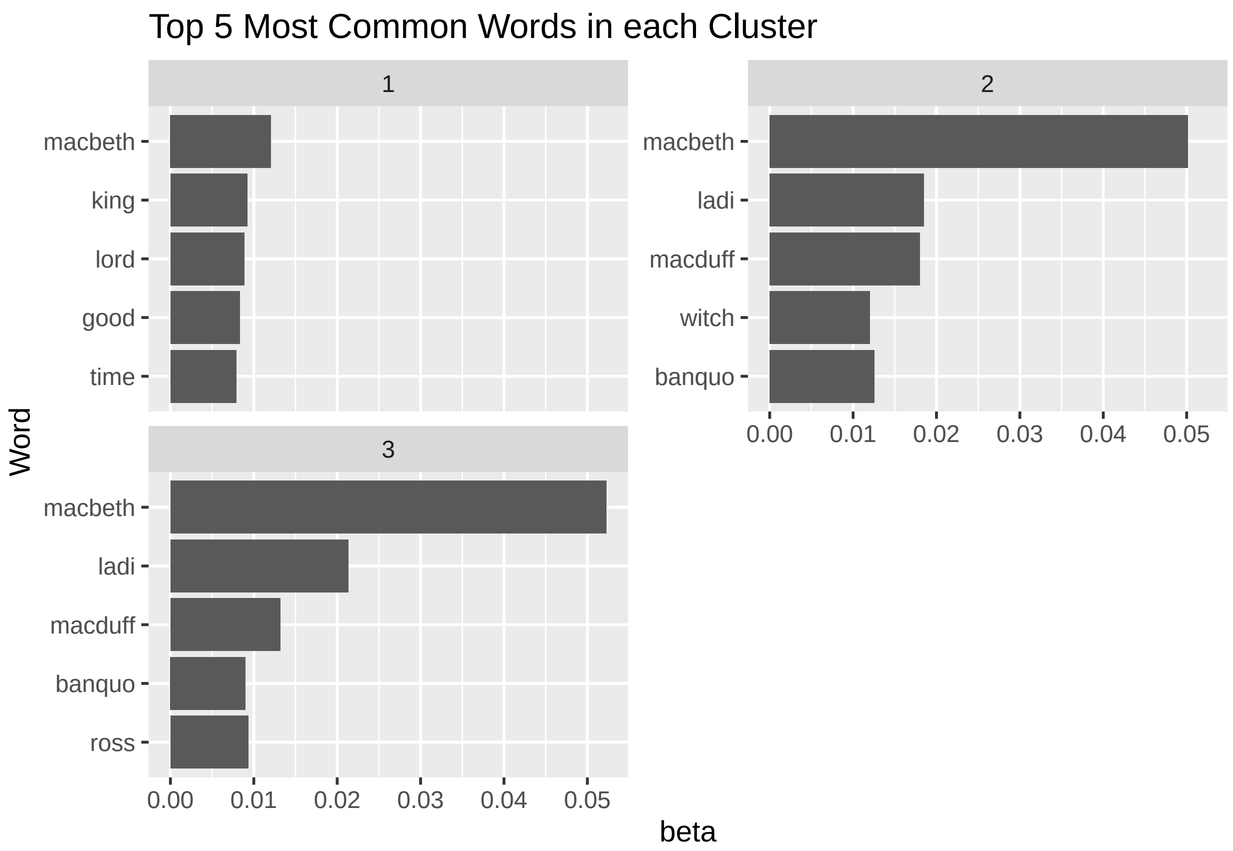
```
## # A tibble: 20 x 5
## # Groups:   document [4]
##    document                  term    count Prediction `Incorrect Prediction`
##    <chr>                     <chr>   <dbl>      <dbl>                  <int>
##  1 Macbeth Original Play.pdf macbeth   287          2                      2
##  2 Macbeth Original Play.pdf macduff   107          2                      3
##  3 Macbeth Original Play.pdf ladi       96          2                       2
##  4 Macbeth Original Play.pdf banquo     76          2                       3
##  5 Macbeth Original Play.pdf witch      60          2                       3
##  6 Macbeth1948.pdf           lord       28          1                       3
##  7 Macbeth1948.pdf           fear       27          1                       2
##  8 Macbeth1948.pdf           good       27          1                       2
##  9 Macbeth1948.pdf           king       27          1                       3
## 10 Macbeth1948.pdf           time       27          1                       2
## 11 Macbeth2015.pdf           macbeth    44          1                       2
## 12 Macbeth2015.pdf           king       24          1                       3
## 13 Macbeth2015.pdf           lord       23          1                       3
## 14 Macbeth2015.pdf           hail       21          1                       3
## 15 Macbeth2015.pdf           thane      21          1                       3
## 16 Macbeth2020.pdf           macbeth   488          3                       2
## 17 Macbeth2020.pdf           ladi      199          3                       2
## 18 Macbeth2020.pdf           macduff   123          3                       3
## 19 Macbeth2020.pdf           ross       87          3                       3
## 20 Macbeth2020.pdf           banquo     84          3                       3
```

```
TopIncorrectWords <-LDA.tidy %>% group_by(term) %>% slice_max(n=5, beta, with_ties = FALSE) %>% slice_mi
TopCorrectWords <-LDA.tidy %>% group_by(topic) %>% slice_max(beta, n = 5, with_ties = FALSE)
```

```
ggplot(TopIncorrectWords, aes(beta, reorder(term,beta))) + geom_col() + facet_wrap(~topic, ncol = 2, sca
```



```
ggplot(TopCorrectWords, aes(beta, reorder(term,beta))) + geom_col() + facet_wrap(~topic, ncol = 2, scal
```



```
LDA.tidy %>% group_by(term) %>% mutate(WordLength = nchar(term))  %>% group_by(topic) %>% slice_max(n =
```

```
## # A tibble: 3 x 2
##   topic  mean
##   <int> <dbl>
## 1     1  4.52
## 2     2  4.9
## 3     3  4.68
```

```
LDA.tidy %>% group_by(term) %>% mutate(WordLength = nchar(term))  %>% group_by(topic) %>% slice_max(n =
```

```
## # A tibble: 3 x 2
##   topic  mean
##   <int> <dbl>
## 1     1  4.78
## 2     2  4.98
## 3     3  4.63
```

```
#WordLengths <-LDA.tidy %>% group_by(term) %>% mutate(WordLength = nchar(term))  %>% group_by(topic)  %
WordLengths <-LDA.tidy %>% group_by(term) %>% mutate(WordLength = nchar(term)) %>% group_by(topic) %>% s
WordLengths
```

```
## <list_of<
##   tbl_df<
##     topic     : integer
##     term      : character
##     beta      : double
##     WordLength: integer
##   >
## >[3]>
## [[1]]
## # A tibble: 300 x 4
##    topic term       beta WordLength
##    <int> <chr>     <dbl>      <int>
##  1     1 macbeth 0.0121           7
##  2     1 king    0.00921          4
##  3     1 lord    0.00886          4
##  4     1 good    0.00831          4
##  5     1 time    0.00790          4
##  6     1 fear    0.00788          4
##  7     1 thane   0.00703          5
##  8     1 hand    0.00686          4
##  9     1 man     0.00610          3
## 10     1 hail    0.00605          4
## # ... with 290 more rows
##
## [[2]]
## # A tibble: 300 x 4
##    topic term       beta WordLength
##    <int> <chr>     <dbl>      <int>
##  1     2 macbeth 0.0502           7
##  2     2 ladi    0.0185           4
##  3     2 macduff 0.0180           7
##  4     2 banquo  0.0125           6
##  5     2 witch   0.0120           5
##  6     2 ross    0.0106           4
##  7     2 first   0.0105           5
##  8     2 malcolm 0.00998          7
##  9     2 murder  0.00806          6
## 10     2 lennox  0.00616          6
## # ... with 290 more rows
##
## [[3]]
## # A tibble: 300 x 4
##    topic term       beta WordLength
##    <int> <chr>     <dbl>      <int>
##  1     3 macbeth 0.0523           7
##  2     3 ladi    0.0213           4
##  3     3 macduff 0.0132           7
##  4     3 ross    0.00932          4
##  5     3 banquo  0.00900          6
##  6     3 hand    0.00825          4
##  7     3 murder  0.00653          6
##  8     3 turn    0.00643          4
##  9     3 duncan  0.00632          6
## 10     3 close   0.00600          5
## # ... with 290 more rows
```

```r
#%>% spread(topic,WordLength)
WordLengths[[1]] #topic 1
```

```
## # A tibble: 300 x 4
##    topic term       beta WordLength
##    <int> <chr>     <dbl>      <int>
##  1     1 macbeth 0.0121          7
##  2     1 king    0.00921         4
##  3     1 lord    0.00886         4
##  4     1 good    0.00831         4
##  5     1 time    0.00790         4
##  6     1 fear    0.00788         4
##  7     1 thane   0.00703         5
##  8     1 hand    0.00686         4
##  9     1 man     0.00610         3
## 10     1 hail    0.00605         4
## # ... with 290 more rows
```

```
cor.test(WordLengths[[1]][["WordLength"]], WordLengths[[2]][["WordLength"]])
```

```
## 
##  Pearson's product-moment correlation
## 
## data:  WordLengths[[1]][["WordLength"]] and WordLengths[[2]][["WordLength"]]
## t = 1.0939, df = 298, p-value = 0.2749
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.05035819  0.17522919
## sample estimates:
##       cor 
## 0.0632433
```

```
cor.test(WordLengths[[2]][["WordLength"]], WordLengths[[3]][["WordLength"]])
```

```
## 
##  Pearson's product-moment correlation
## 
## data:  WordLengths[[2]][["WordLength"]] and WordLengths[[3]][["WordLength"]]
## t = 1.5954, df = 298, p-value = 0.1117
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.0214364  0.2031516
## sample estimates:
##        cor 
## 0.09202783
```

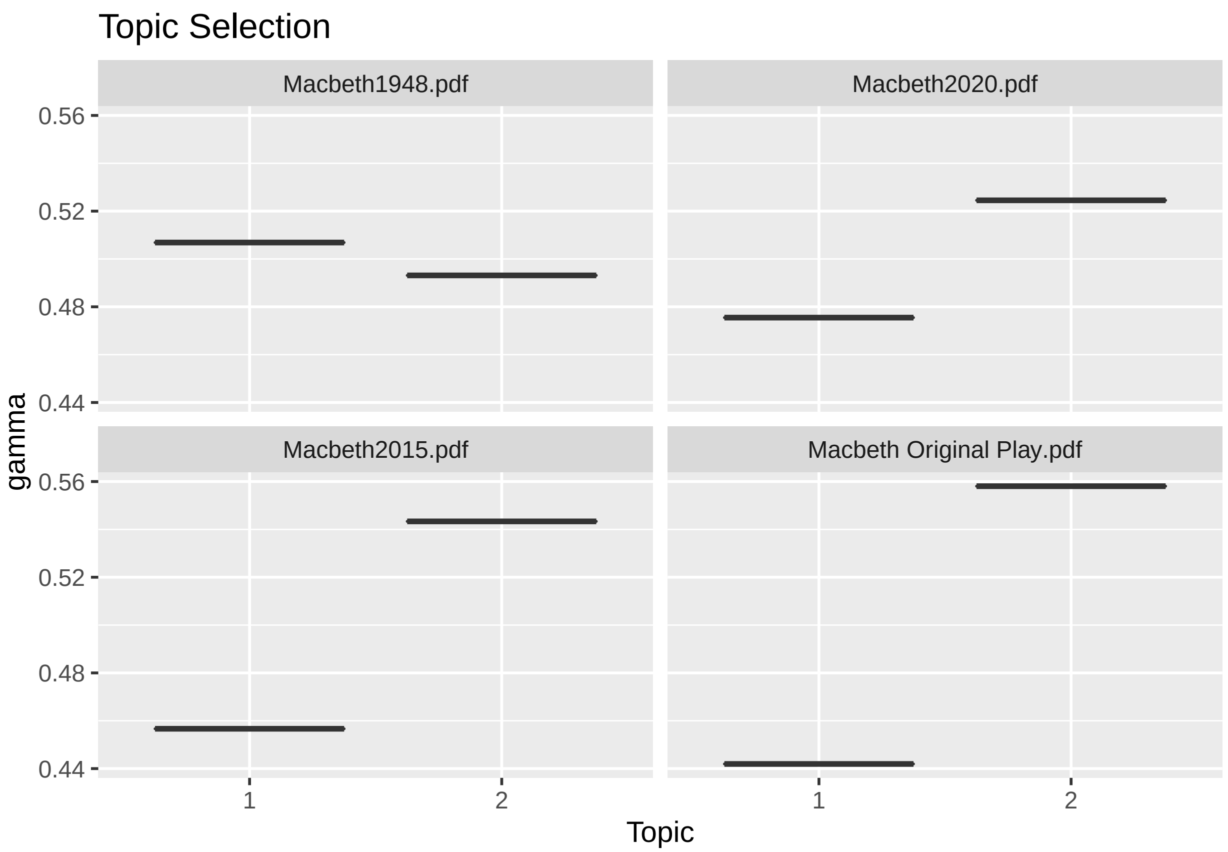
```
cor.test(WordLengths[[1]][["WordLength"]], WordLengths[[3]][["WordLength"]])
```

```
## 
##  Pearson's product-moment correlation
## 
## data:  WordLengths[[1]][["WordLength"]] and WordLengths[[3]][["WordLength"]]
## t = 0.82718, df = 298, p-value = 0.4088
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.0657350  0.1602345
## sample estimates:
##        cor 
## 0.04786212
```

```
LDA.model <- LDA(dtm, k = 2, control = list(seed = 1128))
LDA.tidy <- tidy(LDA.model, matrix = "beta")
LDA.tidy <- LDA.tidy %>% group_by(topic)
sortedLDA <-arrange(LDA.tidy, desc(beta), .by_group = TRUE)
sortedLDA %>% top_n(10, beta)
```

```
## # A tibble: 20 x 3
## # Groups:   topic [2]
##    topic term       beta
##    <int> <chr>     <dbl>
##  1     1 ladi    0.0253
##  2     1 murder  0.0127
##  3     1 ross    0.0116
##  4     1 king    0.00873
##  5     1 lord    0.00736
##  6     1 hand    0.00661
##  7     1 lennox  0.00643
##  8     1 good    0.00636
##  9     1 malcolm 0.00613
## 10     1 live    0.00578
## 11     2 macbeth 0.0642
## 12     2 macduff 0.0172
## 13     2 banquo  0.0145
## 14     2 time    0.00752
## 15     2 hand    0.00653
## 16     2 man     0.00635
## 17     2 witch   0.00573
## 18     2 good    0.00531
## 19     2 duncan  0.00518
## 20     2 fear    0.00464
```

```
LDA.tidy2 <- tidy(LDA.model, matrix = "gamma")
LDA.tidy2 <-LDA.tidy2 %>% mutate(document = reorder(document, gamma * topic))
ggplot(LDA.tidy2, aes(factor(topic), gamma)) + geom_boxplot() + facet_wrap(~ document)  + ggtitle("Topi
```



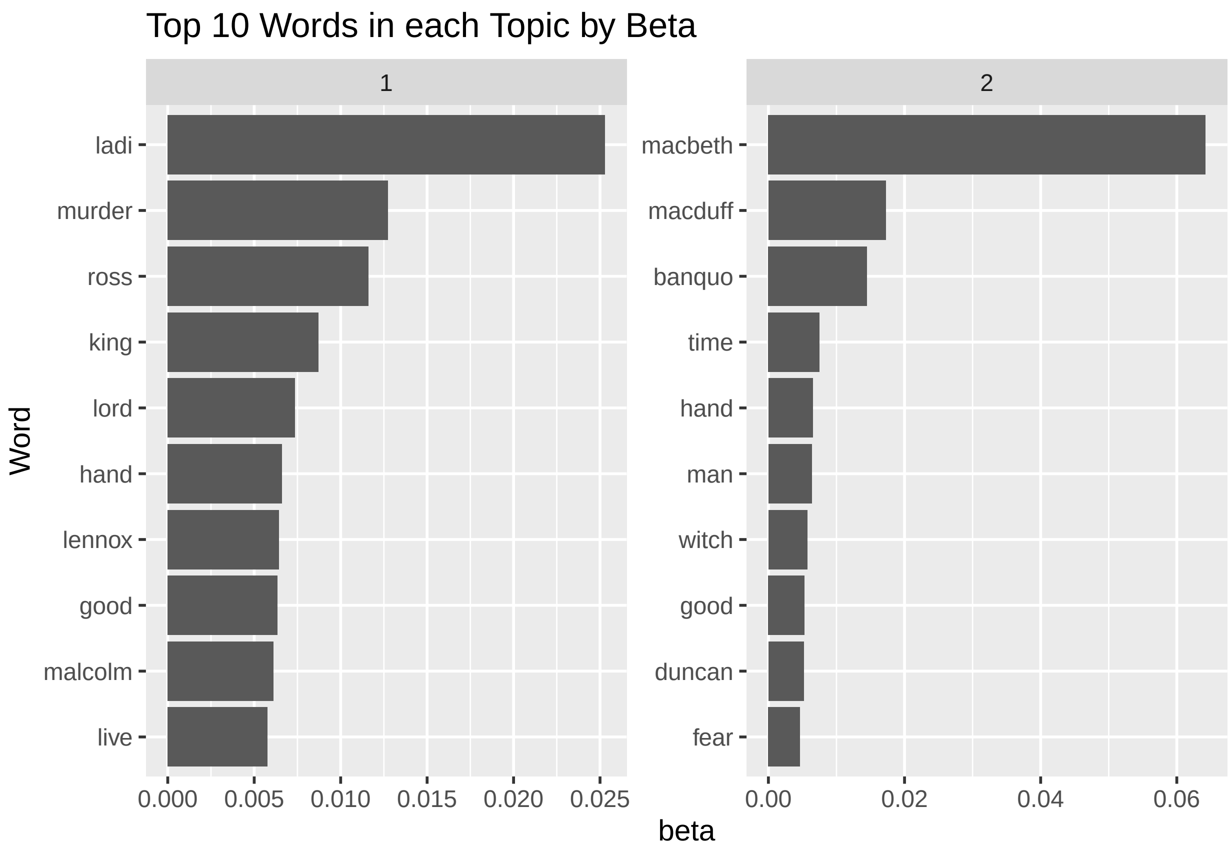
```
top10LDA <- top_n(sortedLDA, 10)
```

```
## Selecting by beta
```

```
top10LDA
```

```
## # A tibble: 20 x 3
## # Groups:   topic [2]
##    topic term       beta
##    <int> <chr>     <dbl>
##  1     1 ladi    0.0253
##  2     1 murder  0.0127
##  3     1 ross    0.0116
##  4     1 king    0.00873
##  5     1 lord    0.00736
##  6     1 hand    0.00661
##  7     1 lennox  0.00643
##  8     1 good    0.00636
##  9     1 malcolm 0.00613
## 10     1 live    0.00578
## 11     2 macbeth 0.0642
## 12     2 macduff 0.0172
## 13     2 banquo  0.0145
## 14     2 time    0.00752
## 15     2 hand    0.00653
## 16     2 man     0.00635
## 17     2 witch   0.00573
## 18     2 good    0.00531
## 19     2 duncan  0.00518
## 20     2 fear    0.00464
```

```
ggplot(top10LDA, aes(reorder_within(term, beta, topic),beta)) + geom_col(show.legend = FALSE) +facet_wra
"free") +coord_flip() +scale_x_reordered() + ggtitle("Top 10 Words in each Topic by Beta") + xlab("Word
```



```
classification <- LDA.tidy %>% group_by(term) %>% top_n(1, beta) %>% ungroup()
#classification #What model thinks the chapter belongs to which topic
Missclassification <- LDA.tidy %>% group_by(term) %>% top_n(2, beta) %>% slice_min(n=1,beta) %>% transmu
classification %>% inner_join(Missclassification, by = "term")
```

```
## # A tibble: 3,298 x 4
##    topic term             beta `Incorrect Prediction`
##    <int> <chr>           <dbl>                  <int>
##  1     1 -accompani 0.0000687                      2
##  2     2 -appar     0.0000762                      1
##  3     1 -two-      0.0000661                      2
##  4     2 ‘em        0.000328                       1
##  5     1 ‘gainst    0.0000577                      2
##  6     1 ‘hail      0.0000478                      2
##  7     2 ‘twere     0.000244                       1
##  8     1 ‘twixt     0.0000449                      2
##  9     1 ‘twould    0.0000583                      2
## 10     2 ’dst       0.0000903                      1
## # ... with 3,288 more rows
```

```r
assignments <- augment(LDA.model, data = Tidydf)
assignments
```

```
## # A tibble: 7,702 x 4
##    document                  term    count .topic
##    <chr>                     <chr>   <dbl>  <dbl>
##  1 Macbeth Original Play.pdf abe         1      1
##  2 Macbeth Original Play.pdf abhor       1      2
##  3 Macbeth Original Play.pdf abid        2      1
##  4 Macbeth Original Play.pdf abjur       1      2
##  5 Macbeth Original Play.pdf abound      1      2
##  6 Macbeth Original Play.pdf abroad      2      2
##  7 Macbeth Original Play.pdf absenc      2      2
##  8 Macbeth Original Play.pdf absent      1      2
##  9 Macbeth Original Play.pdf absolut     3      2
## 10 Macbeth Original Play.pdf abus        1      2
## # ... with 7,692 more rows
```

```r
missclassifiedterms <-assignments %>%  left_join(Missclassification) %>%group_by(term) %>% rename("Predi
```

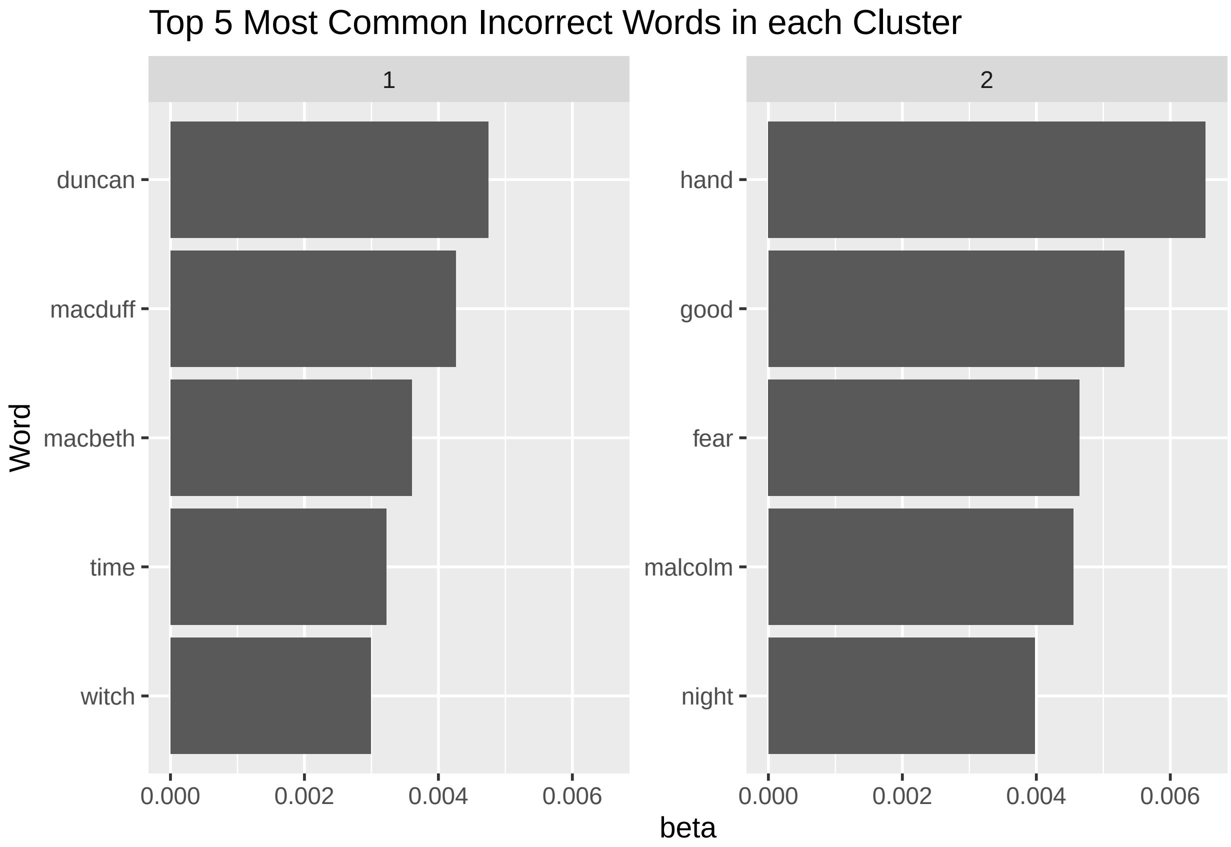
```
## Joining, by = "term"
```

```r
#%>% mutate(percent=count/sum(count)) %>% filter(term != consensus)
# %>%
# ggplot(aes(consensus, term, fill = percent)) +geom_tile() +
# scale_fill_gradient2(high = "red", label = percent_format()) +theme_minimal() +
# theme(axis.text.x = element_text(angle = 90, hjust = 1),
# panel.grid = element_blank()) +
# labs(x = "Document words were assigned to",y = "Book words came from",fill = "% of
# assignments")
missclassifiedterms
```

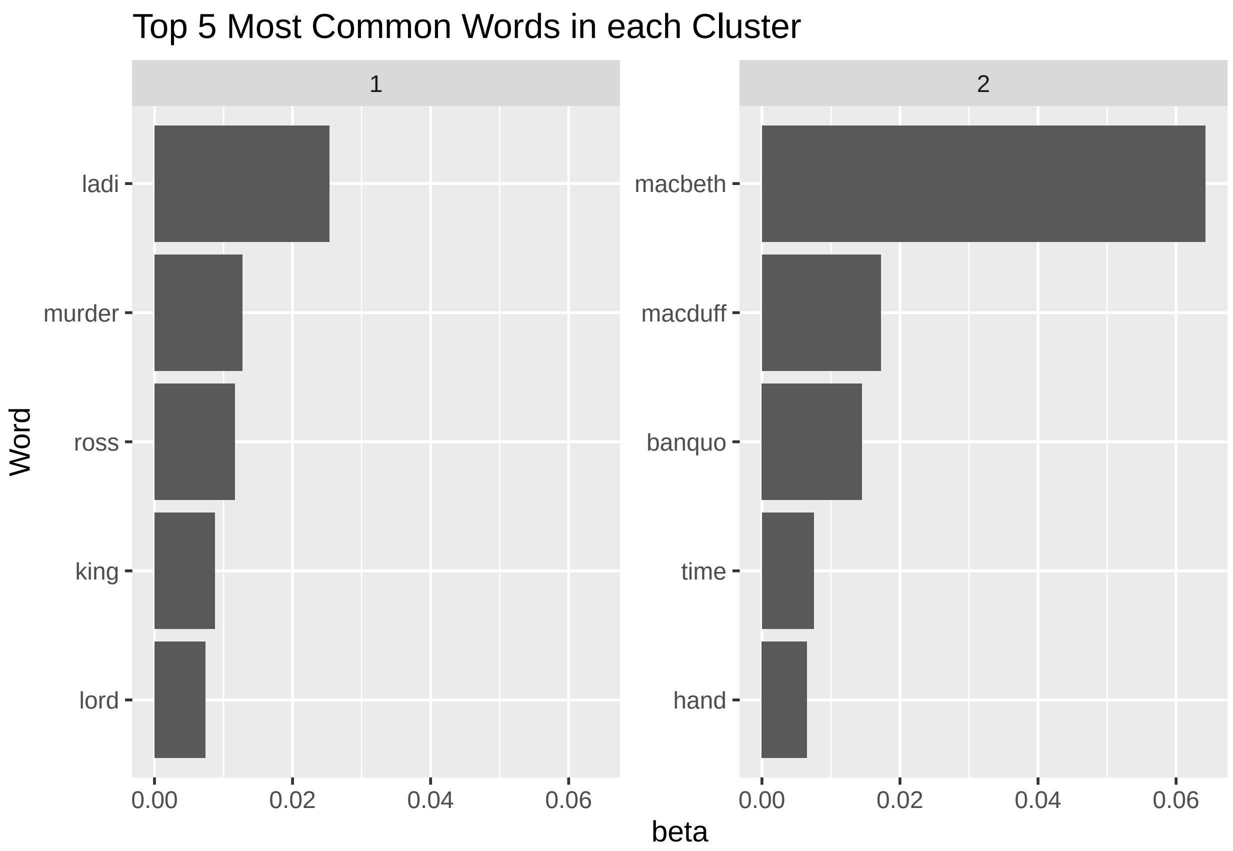
```
## # A tibble: 20 x 5
## # Groups:   document [4]
##    document                  term    count Prediction `Incorrect Prediction`
##    <chr>                     <chr>   <dbl>      <dbl>                  <int>
##  1 Macbeth Original Play.pdf macbeth   287          2                      1
##  2 Macbeth Original Play.pdf macduff   107          2                      1
##  3 Macbeth Original Play.pdf ladi       96          1                      2
##  4 Macbeth Original Play.pdf banquo     76          2                      1
##  5 Macbeth Original Play.pdf witch      60          2                      1
##  6 Macbeth1948.pdf           lord       28          1                      2
##  7 Macbeth1948.pdf           fear       27          1                      2
##  8 Macbeth1948.pdf           good       27          1                      2
##  9 Macbeth1948.pdf           king       27          1                      2
## 10 Macbeth1948.pdf           time       27          2                      1
## 11 Macbeth2015.pdf           macbeth    44          2                      1
## 12 Macbeth2015.pdf           king       24          1                      2
## 13 Macbeth2015.pdf           lord       23          1                      2
## 14 Macbeth2015.pdf           hail       21          1                      2
## 15 Macbeth2015.pdf           thane      21          1                      2
## 16 Macbeth2020.pdf           macbeth   488          2                      1
## 17 Macbeth2020.pdf           ladi      199          1                      2
## 18 Macbeth2020.pdf           macduff   123          2                      1
## 19 Macbeth2020.pdf           ross       87          1                      2
## 20 Macbeth2020.pdf           banquo     84          2                       1
```

```
TopIncorrectWords <-LDA.tidy %>% group_by(term) %>% slice_max(n=5, beta, with_ties = FALSE) %>% slice_mi
TopCorrectWords <-LDA.tidy %>% group_by(topic) %>% slice_max(beta, n = 5, with_ties = FALSE)
```

```
ggplot(TopIncorrectWords, aes(beta, reorder(term,beta))) + geom_col() + facet_wrap(~topic, ncol = 2, sca
```



```
ggplot(TopCorrectWords, aes(beta, reorder(term,beta))) + geom_col() + facet_wrap(~topic, ncol = 2, scale
```


Top 5 Most Common Words in each Cluster
1
2
Word
ladi
murder
ross
king
lord
macbeth
macduff
banquo
time
hand
0.00
0.02
0.04
0.06
0.00
0.02
0.04
0.06
beta