1 Problem 2.a

```
setwd("D:/Berkeley//stat243ps/ps3")
library(stringr)
#library(Rcurl)
URL<-"http://www.gutenberg.org/cache/epub/100/pg100.txt"
originaltext<-readLines(URL)

textbeginning<-grep("^1[[:digit:]]{3}",originaltext)
textend<-grep("THE END",originaltext)

if (length(textbeginning)==length(textend))


play<-list()
for (i in 1:length(textbeginning)){
  play[[i]]<-originaltext[textbeginning[i]:textend[i]]
}

#update the dimension of play ,textbeginning and textend
#ignore the first and last piece
play[length(textbeginning)]<-NULL
play[1]<-NULL

textbeginning<-textbeginning[-(length(play)+2)]
textbeginning<-textbeginning[-1]

textend<-textend[-(length(play)+2)]
textend<-textend[-1]
```

2 Problem 2.b

The playyear is the line with four digitals and the title is the first line in the play that contains words.

To extract the play, I use the playyear to be the beginning part and ”THE END” to be the end part.

I then tried to extract the number of acts by grepping the pattern with ”ACT *. SCENE 1(or I).”. But it seems weird that this way failed in the first play, where ”SCENE 2” ”SCENE3”... are also grepped.

```
playyear<-list()
playtitle<-list()
playactnum<-list()
```

```
playscenenum<-list()

for (i in 1:length(play)){
   #grep the year of the play
    playyear[i]<-originaltext[textbeginning[i]]

    #grep the title
    titlelocation<-grep("^[a-zA-Z]",play[[i]])
    playtitle[i]<-play[[i]][titlelocation[1]]

    #number of acts
    #Using "SCENE 1." or "SCENE I." to detect the number of acts
    tmp<-grep("^ *Act ([A-Z]{1,2})|[0-9]{1,2}\\. ((Scene 1)|(Scene I))\\.$",
              play[[i]],ignore.case=TRUE)
    #tmp<-grep("^ *Act ([A-Z]{1,2})|([0-9]{1,2})\\. Scene I\\.£",play[[i]],ignore.case=TRUE)
    playactnum[i]<-length(tmp)

    #number of scenes
    tmp<-grep("scene ",play[[i]],ignore.case=TRUE)
    playscenenum[i]<-length(tmp)
}
playyear<-unlist(playyear)
playtitle<-unlist(playtitle)
playactnum<-unlist(playactnum)
playscenenum<-unlist(playscenenum)
playyear
```

```
##  [1] "1603" "1607" "1601" "1593" "1608" "1609" "1604" "1598" "1598" "1599"
## [11] "1592" "1591" "1591" "1611" "1597" "1599" "1606" "1595" "1606" "1605"
## [21] "1597" "1601" "1596" "1599" "1605" "1596" "1593" "1595" "1594" "1612"
## [31] "1608" "1594" "1602" "1602" "1595" "1611"
```

```
playtitle
```

```
##  [1] "ALLS WELL THAT ENDS WELL"
##  [2] "THE TRAGEDY OF ANTONY AND CLEOPATRA"
##  [3] "AS YOU LIKE IT"
##  [4] "THE COMEDY OF ERRORS"
##  [5] "THE TRAGEDY OF CORIOLANUS"
##  [6] "CYMBELINE"
##  [7] "THE TRAGEDY OF HAMLET, PRINCE OF DENMARK"
##  [8] "THE FIRST PART OF KING HENRY THE FOURTH"
##  [9] "SECOND PART OF KING HENRY IV"
## [10] "THE LIFE OF KING HENRY THE FIFTH"
## [11] "THE FIRST PART OF HENRY THE SIXTH"
## [12] "THE SECOND PART OF KING HENRY THE SIXTH"
## [13] "THE THIRD PART OF KING HENRY THE SIXTH"
## [14] "KING HENRY THE EIGHTH"
## [15] "KING JOHN"
## [16] "THE TRAGEDY OF JULIUS CAESAR"
## [17] "THE TRAGEDY OF KING LEAR"
## [18] "LOVE'S LABOUR'S LOST"
## [19] "THE TRAGEDY OF MACBETH"
## [20] "MEASURE FOR MEASURE"
## [21] "THE MERCHANT OF VENICE"
## [22] "THE MERRY WIVES OF WINDSOR"
## [23] "A MIDSUMMER NIGHT'S DREAM"
## [24] "MUCH ADO ABOUT NOTHING"
## [25] "THE TRAGEDY OF OTHELLO, MOOR OF VENICE"
## [26] "KING RICHARD THE SECOND"
## [27] "KING RICHARD III"
## [28] "THE TRAGEDY OF ROMEO AND JULIET"
## [29] "THE TAMING OF THE SHREW"
## [30] "THE TEMPEST"
## [31] "THE LIFE OF TIMON OF ATHENS"
## [32] "THE TRAGEDY OF TITUS ANDRONICUS"
## [33] "THE HISTORY OF TROILUS AND CRESSIDA"
## [34] "TWELFTH NIGHT; OR, WHAT YOU WILL"
## [35] "THE TWO GENTLEMEN OF VERONA"
## [36] "THE WINTER'S TALE"
```

```
playactnum
```

```
##  [1] 23  5  5  5  5  5  6  5  5  5  5  5  5 17  5  5  5  5  5  5  5  5  5
## [24]  5  5  5  5  5  5  5  5  5 24  5  5  5
```

```
playscenenum
```

```
##  [1] 23 42 23 11 29 27 21 20 20 25 27 24 29 17 16 19 26 11 29 17 20 23 11
## [24] 18 15 21 26 25 14  9 17 14 25 18 20 17
```

3 Problem 2.c

To find the location of each spoken chunks, I first identified the location of each line start with the speaker. Then any lines betweem two speaker-line is belongs to one spoken chunk.

```
text<-list()
chunktext<-list()
chunklocation<-list()
```

```r
#define a function to find the location of text chunk in each play
findlocation <- function(playindex){
  #location<-grep("^ {0,3}[A-Z]+[a-z]*( [A-Z]+[a-z]*){0,2}\\. [A-Z][a-z]+",play[[playindex]]
  location<-grep("^ {0,3}[A-Z]+[a-z]*( [A-Z]+[a-z]*){0,2}\\. [A-Z](([a-z]+)| |,)"
                 ,play[[playindex]])
  location[length(location)+1] <- length(play[[playindex]])-1
  return(location)
}

#Using for loop to find the chunklocation in all plays and assign them to an list element
chunklocation<-list()
for (i in 1:length(play)){
  chunklocation[[i]] <- findlocation(i)
}


#define a function to find the text chunk in each play
findtext<-function(playindex,locationindex1,locationindex2){
  text<-play[[playindex]][locationindex1:locationindex2]
  return(text)
}


chunktext<-list()
for (i in 1:length(play)){
  chunktext[[i]] <- list()
}

#Using for loop to find the chunktext in all plays and assign them to an list element
for (i in 1:length(play)){
  for (j in 1:(length(chunklocation[[i]])-1)){
    chunktext[[i]][[j]] <- findtext(i,chunklocation[[i]][j],(chunklocation[[i]][j+1]-1))
  }
}
```

4 Problem 2.d

4.1 The number of unique speakers

I grepped the part before ”.” on the speaker-lines I found in problem 2.c. That part is the speaker names. I then use ”unique” function to find the unique speaker names.

```
#The number of unique speakers
firstline<-list()
firstlinesplit<-list()
speakers<-list()
findspeakers<-function(inputplay){
  for (i in 1:length(inputplay)){
    firstline[[i]]<-inputplay[[i]][1]
    firstlinesplit[[i]]<-strsplit(firstline[[i]],"\\.")
    speakers[[i]]<-firstlinesplit[[i]][[1]][1]
    speakers[[i]] <- gsub(" +","",speakers[[i]])
  }
    return(speakers)
}
#findspeakers(chunktext[[2]])
speakerlist<-lapply(chunktext,findspeakers)
uniquespeakers<-lapply(speakerlist,unique)
unispeakernum<-unlist(lapply(uniquespeakers,length))
```

4.2 The number of spoken chunks

I used lapply to find the length of spoken chunks in each play.

```
#number of spoken chunks in each play
chunknum<-lapply(chunktext,length)
chunknum<-unlist(chunknum)
```

4.3 THe number of sentences and words spoken and average number of words per chunk

I used ”.”,”?”,”!” to sperate every spoken chunks to find out the number of sentences; ” ” to separate every spoken chunks to find out the number of words. During the process, I also replace the irrelevant punctuation such as commas and periods but not apostrophes.

```
namenum<-unlist(lapply(speakerlist,length))
#number of sentences
sentences<-list()
countsentences<-function(inputchunk){
  inputchunk<-unlist(inputchunk)
  sentences <- unlist(strsplit(inputchunk,"\\.|\\?|!"))
  return(length(sentences))
}
```

```
#Using the sentences number calculated by conutsentences minus the number of speaker names.
chunksensnum<-unlist(lapply(chunktext,countsentences))
chunksensnum<-chunksensnum-namenum


#number of words
words<-list()
countwords<-function(inputchunk){
  sum<-0
  inputchunk<-unlist(inputchunk)
  inputchunk <- gsub(",|/.|/?|!|:|;","",inputchunk)
  words <- unlist(strsplit(inputchunk," {1,}"))
  return(length(words))
}

chunkwordsnum<-unlist(lapply(chunktext,countwords))
chunkwordsnum<-unlist(chunkwordsnum)-unlist(namenum)



#average number of words per chunk
avgwords<-chunkwordsnum/chunknum
```

4.4 The number of unique words

I used the same way as above to find out the words in each play and ”unique” them.

```
words<-list()
countuniquewords<-function(inputchunk){
  inputchunk<-unlist(inputchunk)
  inputchunk <- gsub(",|/.|/?|!|:|;","",inputchunk)
  words <- unlist(strsplit(inputchunk," {1,}"))
  words<-unique(tolower(words))
  words<-
  return(length(words))
}
uniwordsnum<-unlist(lapply(chunktext,countuniquewords))
uniwordsnum<-unlist(uniwordsnum)-unispeakernum
```

4.5 Result Summary

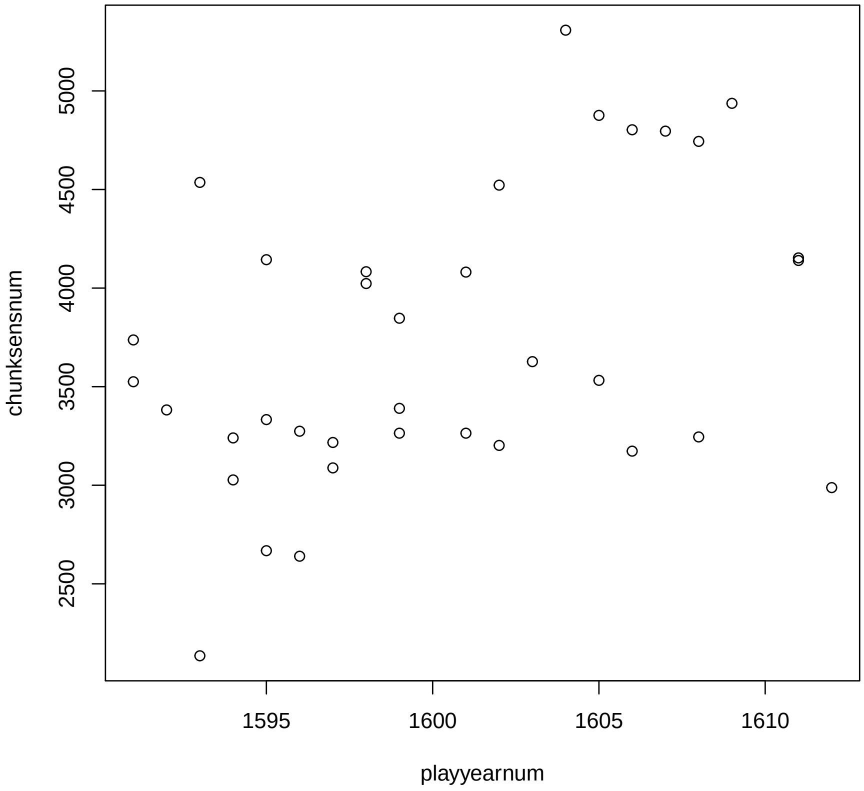
Below is the result of summary statistics of each play.

```
#number of sentences
chunksensnum

##  [1] 3627 4796 3264 2135 4744 4937 5308 4023 4083 3847 3382 3737 3525 4140
## [15] 3088 3264 4803 3333 3173 3532 3217 4081 2640 3390 4876 3274 4536 4144
## [29] 3240 2988 3245 3027 4522 3202 2668 4153

#number of words
chunkwordsnum

##  [1] 26512 28983 24793 16987 32035 31782 35445 28117 30090 30105 25190
## [12] 29309 28280 28336 23996 22856 30221 24861 20188 24947 24298 25609
## [23] 19113 24188 30747 25766 34107 28381 24265 19377 21590 23923 29881
## [34] 22922 19905 28711

#average number of words per chunk
avgwords

##  [1] 29.29503 25.99372 32.36684 29.59408 29.82775 39.67790 33.15716
##  [8] 37.74094 34.34932 41.35302 41.02606 38.71731 35.70707 42.48276
## [15] 44.19153 29.37789 29.80375 25.08678 32.71961 29.84091 39.12721
## [22] 32.91645 41.19181 26.00860 34.74237 47.53875 32.39031 35.65452
## [29] 28.75000 32.45729 29.94452 43.49636 27.26369 26.96706 24.36353
## [36] 39.98747

#number of unique words
uniwordsnum

##  [1] 4369 4947 4050 3074 5034 5352 6014 4867 5019 5396 4611 4927 4446 4542
## [15] 4240 3587 5342 4737 4054 4064 4013 4057 3581 3874 4760 4449 5143 4779
## [29] 4119 3897 4062 4156 5203 3916 3412 4845
```

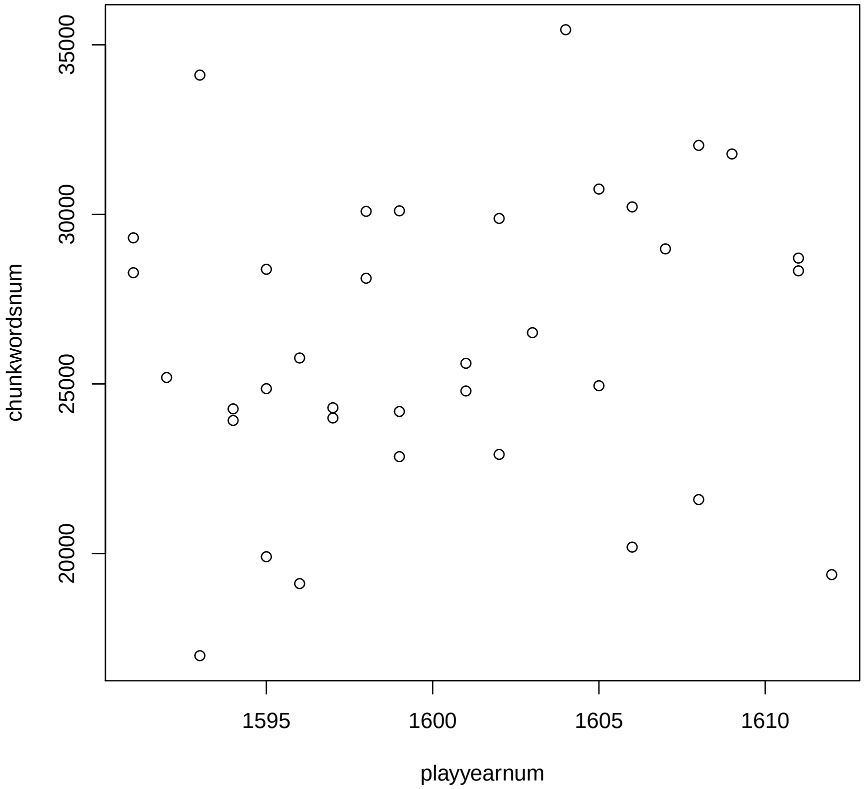
5 Prblem 2.e

Based on the following picutures, we can tell that as tiime pass by, the sentences per chunk tends to increase, words per chunk mainly remians the same, average words per chunk tends to decrease then increase a bit, almost remain the same.

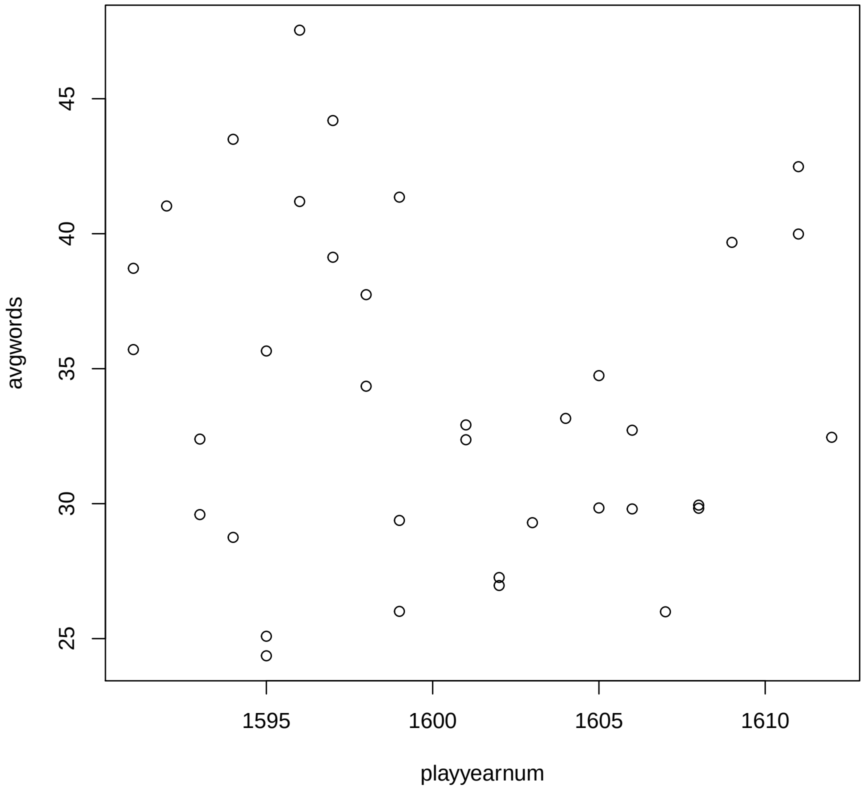
```
playyearnum<-unlist(playyear)
plot(playyearnum,chunksensnum)
```



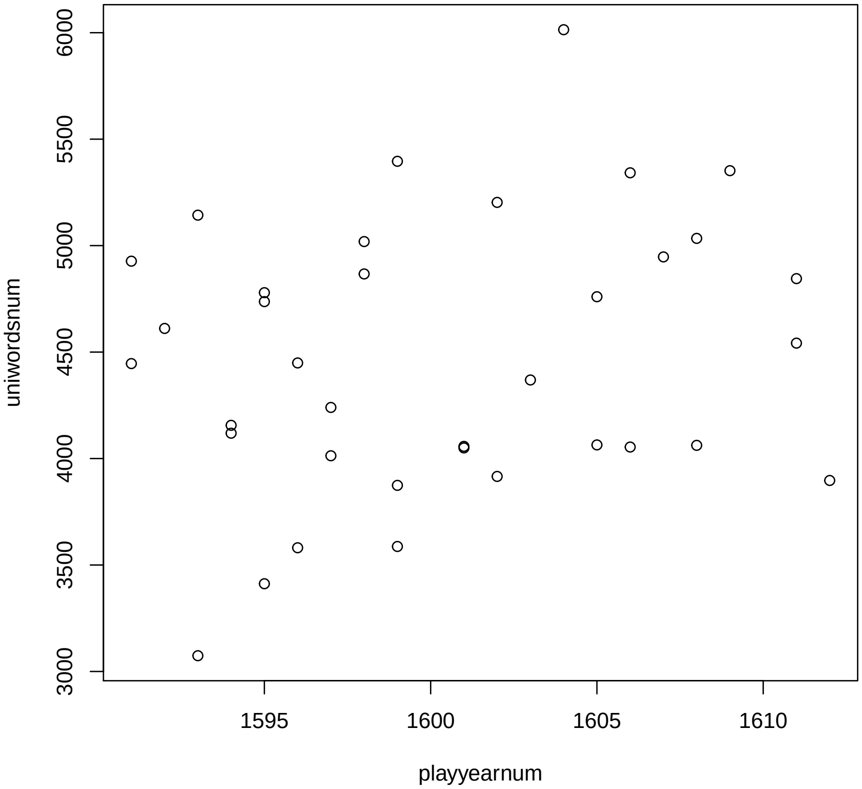
```
plot(playyearnum,chunkwordsnum)
```



```
plot(playyearnum,avgwords)
```



```
plot(playyearnum,uniwordsnum)
```



6 Problem 2.f

I include the fourth play in my analysis.

7 Problem 3.a

The fields for the class are playyear(named characteric vector), playtitle(named characteric vector),number of acts(named numberic vector), number of scenes(named numberic vector), spokenchunks(named characteric lists), speaker for each chunks(named characteric lists), number of unique speakers (named numberic vector), number of spoken chunks(named numberic vector), number of unique words(named numberic vector) .etc.

8 Problem 3.b

There are methods for the class, like to output part of a play, show the basics information of a play, etc.

showplay:showing part of the play; input: number of play; create fields: playyear, playtitle, spoken chunks; output: playyear, playtitle, spoken chunks

playstatistics: showing the relative play statistics; input: number of play; create fields: number of acts, number of scenes, number of unique speakers, number of spoken chunks, number of sentences, and words spoken and average number of words per chunk, number of unique words; output: the value of created fields

playplot: showing the plot of summary statistics; input: number of play, play statistics; output: plot of the play statistics