STAT 243 PS3

Riv Jenkins

1 c)

In the "Best Practices for Scientific Computing" article, the authors claim that writing code in the highest level language possible is preferred even when the final product will need to be written in a lower-level language. They claim this is because higher level languages require fewer lines of code, and the program can later be rewritten in a lower-level language. I do not agree that this will be faster overall because although you may spend more time coding in a lower-level language if you start with that language than you would if you were simply transposing from a higher level language, I don't think the time saved by transposing would equal the amount of time necessary to write the program in the higher level language. There are other arguments in favor of coding in a high level first, and only switching to a low level if necessary, but I do not think time savings is one of them.

2 a)

The following code downloads the full text and creates a character vector with each element corresponding to a different play. I skip the Comedy of Errors.

```r
#download the full text file
fulltext = readLines("http://www.gutenberg.org/cache/epub/100/pg100.txt")
```

```r
#extract overall start and end points for the plays
start_point = which(!is.na(str_extract(fulltext, "1603")))
end_point = which(!is.na(str_extract(fulltext, "We were dissever'd"))) + 2

#create char vector with only plays
all_plays = fulltext[start_point:end_point]

#mark characters' speeches with '##'
all_plays = str_replace_all(all_plays, "^ {2}([A-Z])", "##\\1")
#remove many stage directions
all_plays = str_replace_all(all_plays, " {2}((Enter)|(Exit)|(Re-enter)|(Exeunt)).*$", "")

#separate individual plays using "THE END"
Play_ends = c(0, which(!is.na(str_extract(all_plays, "THE END"))))

Create_Play_Vec <- function(all_plays, Play_ends){
  #this function takes a char vector with evey line a row (all_plays) and returns a char
  #vector with every row a separate play using the endpoints denoted by Play_ends
  char_vec = character(length(Play_ends) - 1)
  for(i in 1:(length(Play_ends)-1)){
    if(i==1){
      char_vec[i] = paste0(all_plays[Play_ends[i]:Play_ends[i+1]], collapse = ' ')
    }
    else if(i==4){
      #for some reason this play is formatted differently
      char_vec[i] = paste0(all_plays[(Play_ends[i]+2):Play_ends[i+1]], collapse = ' ')
    }
    else{
      char_vec[i] = paste0(all_plays[(Play_ends[i]+13):Play_ends[i+1]], collapse = ' ')
    }
  }
  #remove copyright tags
  char_vec = str_replace_all(char_vec, "<<TH[^>]+IP.>>", "")
  return(char_vec)
}

play_vec = Create_Play_Vec(all_plays, Play_ends)
rm(fulltext)
#remove comedy of errors
play_vec = play_vec[-4]
substring(play_vec, 1, 55)
```

```
##  [1] "1603  ALLS WELL THAT ENDS WELL  by William Shakespeare "
##  [2] "    1607  THE TRAGEDY OF ANTONY AND CLEOPATRA  by Willi"
##  [3] "    1601  AS YOU LIKE IT  by William Shakespeare    DRA"
##  [4] "   1608  THE TRAGEDY OF CORIOLANUS  by William Shakespe"
##  [5] "    1609  CYMBELINE  by William Shakespeare    Dramatis"
##  [6] "    1604   THE TRAGEDY OF HAMLET, PRINCE OF DENMARK   b"
##  [7] "    1598  THE FIRST PART OF KING HENRY THE FOURTH   by "
##  [8] "    1598   SECOND PART OF KING HENRY IV  by William Sha"
##  [9] "    1599  THE LIFE OF KING HENRY THE FIFTH  by William "
## [10] "    1592  THE FIRST PART OF HENRY THE SIXTH  by William"
## [11] "    1591  THE SECOND PART OF KING HENRY THE SIXTH  by W"
## [12] "    1591  THE THIRD PART OF KING HENRY THE SIXTH  by Wi"
## [13] "    1611  KING HENRY THE EIGHTH  by William Shakespeare"
## [14] "    1597  KING JOHN  by William Shakespeare    DRAMATIS"
## [15] "     1599   THE TRAGEDY OF JULIUS CAESAR  by William Sh"
## [16] "     1606   THE TRAGEDY OF KING LEAR  by William Shakes"
## [17] "     1595  LOVE'S LABOUR'S LOST  by William Shakespeare"
## [18] "     1606  THE TRAGEDY OF MACBETH   by William Shakespe"
## [19] "    1605   MEASURE FOR MEASURE  by William Shakespeare "
## [20] "    1597  THE MERCHANT OF VENICE  by William Shakespear"
## [21] "    1601  THE MERRY WIVES OF WINDSOR  by William Shakes"
## [22] "    1596  A MIDSUMMER NIGHT'S DREAM  by William Shakesp"
## [23] "    1599   MUCH ADO ABOUT NOTHING   by William Shakespe"
## [24] "    1605   THE TRAGEDY OF OTHELLO, MOOR OF VENICE  by W"
## [25] "    1596   KING RICHARD THE SECOND   by William Shakesp"
## [26] "    1593  KING RICHARD III  by William Shakespeare   Dr"
## [27] "    1595   THE TRAGEDY OF ROMEO AND JULIET  by William "
## [28] "1594      THE TAMING OF THE SHREW  by William Shakespea"
## [29] "    1612  THE TEMPEST  by William Shakespeare    DRAMAT"
## [30] "    1608  THE LIFE OF TIMON OF ATHENS  by William Shake"
## [31] "    1594  THE TRAGEDY OF TITUS ANDRONICUS  by William S"
## [32] "    1602  THE HISTORY OF TROILUS AND CRESSIDA  by Willi"
## [33] "    1602   TWELFTH NIGHT; OR, WHAT YOU WILL  by William"
## [34] "  1595  THE TWO GENTLEMEN OF VERONA  by William Shakesp"
## [35] "    1611  THE WINTER'S TALE  by William Shakespeare    "
```

b)

The following code takes the character vector constructed above and turns each play into an object which has the title, number of acts, scenes, year produced, and the text of the body.

```
Play <- function(Title = NULL, Acts = 5, Scenes = NULL, Characters = NULL, Year = NULL, Body = NULL){
  #defines a class for a play (all Shakespeare plays should have 5 acts)
  obj = list(Title = Title, Acts = Acts, Scenes = Scenes, Characters = Characters, Year = Year, Body = 
  class(obj) <- 'Play'
  return(obj)
}

#extract years of plays
years = str_extract(play_vec, "[0-9]{4} ")
#extract titles of plays
titles = str_extract(play_vec, "[0-9] [A-Z ,';]+")
titles = str_replace(titles, "[0-9 ]+", "")
#extract number of scenes
scenes = str_count(play_vec, "S[cC][eE][nN][eE]")
#extract body of each play
body = str_extract(play_vec, "ACT.*THE END")
body = sapply(body, str_replace, "THE END", "#THE END")

#construct a list of all plays
plays = list()
for(i in 1:length(play_vec)){
  plays[[i]] <- Play(Title=titles[i], Scenes = scenes[i], Year=years[i], Body = body[i])
}
plays[[2]]
```

```
## $Title
## [1] "THE TRAGEDY OF ANTONY AND CLEOPATRA  "
##
## $Acts
## [1] 5
##
## $Scenes
## [1] 43
##
## $Characters
## NULL
##
## $Year
## [1] "1607 "
##
## $Body
##    ACT I. SCENE I. Alexandria. CLEOPATRA'S palace  Enter DEMETRIUS and PHILO  ##PHILO. Nay, but this
```

c)

The following code takes the objects created above and adds to them by extracting each chunk of spoken text.

```r
Get_speech <- function(play){
  #this function takes a play object and returns the chunks of speech text

  #use the '#' markers to extract the chunks from the body string
  chunks = str_extract_all(play$Body, "#[A-Z a-z]+\\..*?#")
  #get the speaker for each chunk
  chunknames = str_extract_all(chunks, "#[A-Z a-z]+\\.")

  #remove extraneous symbols
  chunks = sapply(chunks, str_replace_all, "#[A-Z a-z]+\\.", "")
  chunknames = sapply(chunknames, str_replace_all, "#", "")
  chunknames = str_replace_all(chunknames, "\\.", "")

  #remove more stage directions
  chunks = str_replace_all(chunks, "\\[[^\\]]+\\]", "")

  return(list(names = chunknames, text = chunks))
}

#extract the spoken chunks for each play
for(i in 1:length(plays)){
  plays[[i]]$Chunks = Get_speech(plays[[i]])
}
print(c(plays[[4]]$Chunks$names[50], plays[[4]]$Chunks$text[50]))
```

```
## [1] "MARCIUS"
## [2] " He that will give good words to thee will flatter     Beneath abhorring. What would you have,
```

d)

The following code calculates many different statistics for each play and displays a summary of some of these statistics.

```r
Calculate_Stats <- function(play){
  #This function takes a play object and calculates many statistics for that play and then
  #returns the modified object

  #extract all speakers from play
  play$Characters = unique(play$Chunks$names)
  #count the number of unique speakers
  num_char = length(play$Characters)

  num_chunks = length(play$Chunks$text)
  #calculate number of sentences
  num_sent = sapply(str_extract_all(play$Chunks$text, "[\\.\\?\\!]+"), length)
  num_sent = sum(num_sent)

  #count number of words, and get average number per chunk
  words = str_extract_all(play$Chunks$text, "[\\w']+")
  num_words = sum(sapply(words, length))
  avg_words = num_words/num_chunks

  #count the number of unique words
  num_un_words = length(unique(unlist(words)))

  #store all the statistics in the Play object
  play$Stats = list(Num_Speakers = num_char, Num_Chunks = num_chunks, Num_Sentences =
                    num_sent, Num_Words = num_words, Avg_Words_Chunk = avg_words,
                    Num_Unique_Words = num_un_words)

  return(play)
}

#calculate statistics for each play
plays = lapply(plays, Calculate_Stats)
```

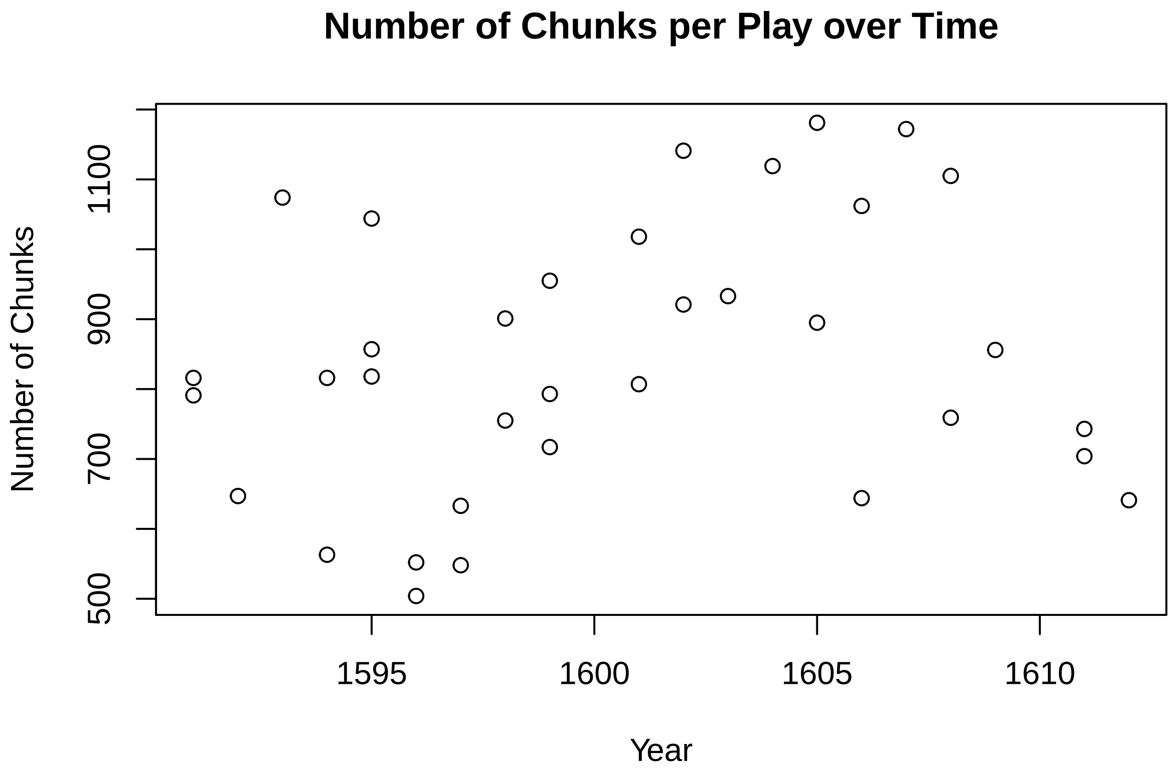
```r
summary.Play <- function(play){
  #this function defines what should be printed when a summary of a Play is requested
  print.noquote(play$Title)
  print.noquote(paste("   Number of Acts:", play$Acts," Number of Scenes:", play$Scenes))
  print.noquote(paste("   Number of unique speakers:", play$Stats$Num_Speakers))
  print.noquote(paste("   Number of spoken chunks:", play$Stats$Num_Chunks))
}
#print the summary statistics for each play
invisible(lapply(plays, summary))
```

```
## [1] ALLS WELL THAT ENDS WELL
## [1]    Number of Acts: 5  Number of Scenes: 24
## [1]    Number of unique speakers: 23
## [1]    Number of spoken chunks: 933
## [1] THE TRAGEDY OF ANTONY AND CLEOPATRA
## [1]    Number of Acts: 5  Number of Scenes: 43
## [1]    Number of unique speakers: 59
## [1]    Number of spoken chunks: 1172
## [1] AS YOU LIKE IT
## [1]    Number of Acts: 5  Number of Scenes: 23
## [1]    Number of unique speakers: 27
## [1]    Number of spoken chunks: 807
## [1] THE TRAGEDY OF CORIOLANUS
## [1]    Number of Acts: 5  Number of Scenes: 30
## [1]    Number of unique speakers: 62
## [1]    Number of spoken chunks: 1105
## [1] CYMBELINE
## [1]    Number of Acts: 5  Number of Scenes: 28
## [1]    Number of unique speakers: 40
## [1]    Number of spoken chunks: 856
## [1] THE TRAGEDY OF HAMLET, PRINCE OF DENMARK
## [1]    Number of Acts: 5  Number of Scenes: 21
## [1]    Number of unique speakers: 33
## [1]    Number of spoken chunks: 1119
## [1] THE FIRST PART OF KING HENRY THE FOURTH
## [1]    Number of Acts: 5  Number of Scenes: 20
## [1]    Number of unique speakers: 35
## [1]    Number of spoken chunks: 755
## [1] SECOND PART OF KING HENRY IV
## [1]    Number of Acts: 5  Number of Scenes: 20
## [1]    Number of unique speakers: 49
## [1]    Number of spoken chunks: 901
## [1] THE LIFE OF KING HENRY THE FIFTH
## [1]    Number of Acts: 5  Number of Scenes: 24
## [1]    Number of unique speakers: 48
## [1]    Number of spoken chunks: 717
## [1] THE FIRST PART OF HENRY THE SIXTH
## [1]    Number of Acts: 5  Number of Scenes: 28
## [1]    Number of unique speakers: 53
## [1]    Number of spoken chunks: 647
## [1] THE SECOND PART OF KING HENRY THE SIXTH
## [1]    Number of Acts: 5  Number of Scenes: 25
## [1]    Number of unique speakers: 67
## [1]    Number of spoken chunks: 791
## [1] THE THIRD PART OF KING HENRY THE SIXTH
## [1]    Number of Acts: 5  Number of Scenes: 29
## [1]    Number of unique speakers: 47
## [1]    Number of spoken chunks: 816
## [1] KING HENRY THE EIGHTH
## [1]    Number of Acts: 5  Number of Scenes: 18
## [1]    Number of unique speakers: 48
## [1]    Number of spoken chunks: 704
## [1] KING JOHN
## [1]    Number of Acts: 5  Number of Scenes: 17
## [1]    Number of unique speakers: 27
## [1]    Number of spoken chunks: 548
## [1] THE TRAGEDY OF JULIUS CAESAR
## [1]    Number of Acts: 5  Number of Scenes: 19
## [1]    Number of unique speakers: 48
## [1]    Number of spoken chunks: 793
## [1] THE TRAGEDY OF KING LEAR
## [1]    Number of Acts: 5  Number of Scenes: 27
## [1]    Number of unique speakers: 23
## [1]    Number of spoken chunks: 1062
## [1] LOVE'S LABOUR'S LOST
## [1]    Number of Acts: 5  Number of Scenes: 10
## [1]    Number of unique speakers: 19
## [1]    Number of spoken chunks: 1044
## [1] THE TRAGEDY OF MACBETH
## [1]    Number of Acts: 5  Number of Scenes: 30
## [1]    Number of unique speakers: 44
## [1]    Number of spoken chunks: 644
## [1] MEASURE FOR MEASURE
## [1]    Number of Acts: 5  Number of Scenes: 18
## [1]    Number of unique speakers: 23
## [1]    Number of spoken chunks: 895
## [1] THE MERCHANT OF VENICE
## [1]    Number of Acts: 5  Number of Scenes: 21
## [1]    Number of unique speakers: 25
## [1]    Number of spoken chunks: 633
## [1] THE MERRY WIVES OF WINDSOR
## [1]    Number of Acts: 5  Number of Scenes: 24
## [1]    Number of unique speakers: 28
## [1]    Number of spoken chunks: 1018
## [1] A MIDSUMMER NIGHT'S DREAM
## [1]    Number of Acts: 5  Number of Scenes: 10
## [1]    Number of unique speakers: 33
## [1]    Number of spoken chunks: 504
## [1] MUCH ADO ABOUT NOTHING
## [1]    Number of Acts: 5  Number of Scenes: 18
## [1]    Number of unique speakers: 23
## [1]    Number of spoken chunks: 955
## [1] THE TRAGEDY OF OTHELLO, MOOR OF VENICE
## [1]    Number of Acts: 5  Number of Scenes: 16
## [1]    Number of unique speakers: 27
## [1]    Number of spoken chunks: 1181
## [1] KING RICHARD THE SECOND
## [1]    Number of Acts: 5  Number of Scenes: 20
## [1]    Number of unique speakers: 36
## [1]    Number of spoken chunks: 552
## [1] KING RICHARD III
## [1]    Number of Acts: 5  Number of Scenes: 26
## [1]    Number of unique speakers: 64
## [1]    Number of spoken chunks: 1074
## [1] THE TRAGEDY OF ROMEO AND JULIET
## [1]    Number of Acts: 5  Number of Scenes: 25
## [1]    Number of unique speakers: 35
## [1]    Number of spoken chunks: 818
## [1] THE TAMING OF THE SHREW
## [1]    Number of Acts: 5  Number of Scenes: 15
## [1]    Number of unique speakers: 27
## [1]    Number of spoken chunks: 816
## [1] THE TEMPEST
## [1]    Number of Acts: 5  Number of Scenes: 10
## [1]    Number of unique speakers: 19
## [1]    Number of spoken chunks: 641
## [1] THE LIFE OF TIMON OF ATHENS
## [1]    Number of Acts: 5  Number of Scenes: 18
## [1]    Number of unique speakers: 58
## [1]    Number of spoken chunks: 759
## [1] THE TRAGEDY OF TITUS ANDRONICUS
## [1]    Number of Acts: 5  Number of Scenes: 15
## [1]    Number of unique speakers: 27
## [1]    Number of spoken chunks: 563
## [1] THE HISTORY OF TROILUS AND CRESSIDA
## [1]    Number of Acts: 5  Number of Scenes: 25
## [1]    Number of unique speakers: 29
## [1]    Number of spoken chunks: 1141
## [1] TWELFTH NIGHT; OR, WHAT YOU WILL
## [1]    Number of Acts: 5  Number of Scenes: 19
## [1]    Number of unique speakers: 21
## [1]    Number of spoken chunks: 921
## [1] THE TWO GENTLEMEN OF VERONA
## [1]    Number of Acts: 5  Number of Scenes: 21
## [1]    Number of unique speakers: 17
## [1]    Number of spoken chunks: 857
## [1] THE WINTER'S TALE
## [1]    Number of Acts: 5  Number of Scenes: 16
## [1]    Number of unique speakers: 34
## [1]    Number of spoken chunks: 743
```

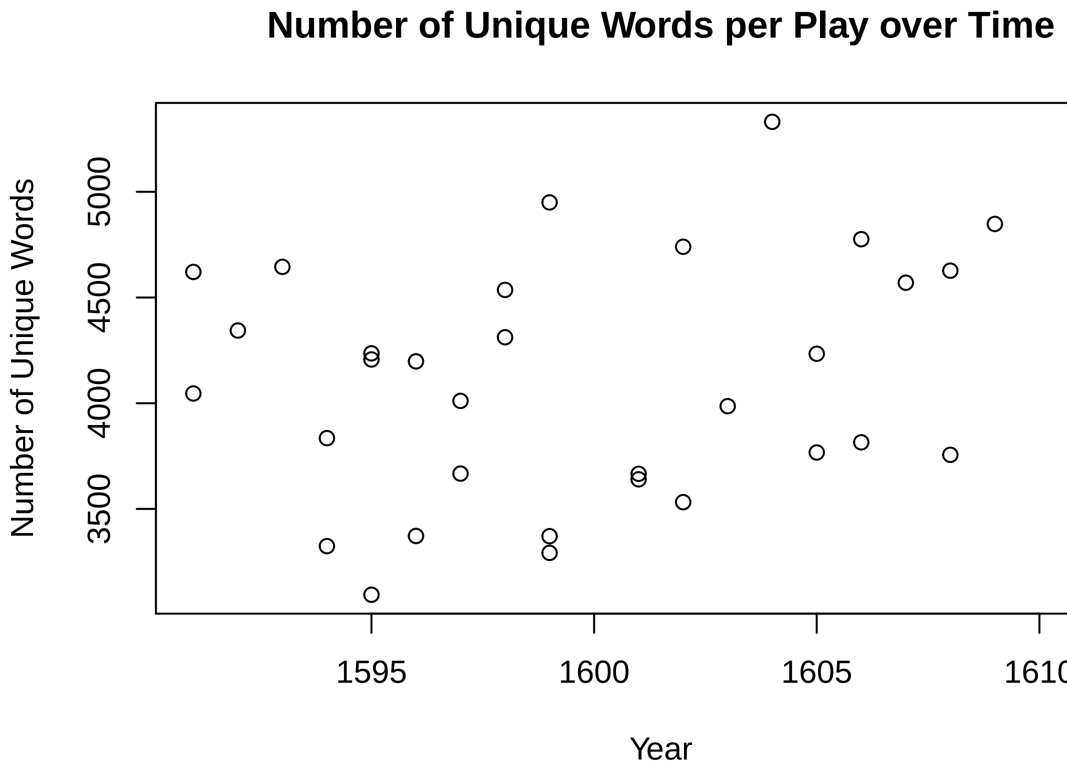
e)

The following plots show number of chunks per play over time, and the number of unique words per play over time. There do not seem to be any obvious trends to me. The strongest trend I saw was that the number of chunks seems to rise and then fall right at the end.

```
y = sapply(plays, function(play) return(play$Stats$Num_Chunks))
x = sapply(plays, function(play) return(as.numeric(play$Year)))
plot(x,y, main="Number of Chunks per Play over Time", xlab="Year", ylab="Number of Chunks")
```



```
y = sapply(plays, function(play) return(play$Stats$Num_Unique_Words))
x = sapply(plays, function(play) return(as.numeric(play$Year)))
plot(x,y, main="Number of Unique Words per Play over Time", xlab="Year", ylab="Number of Unique Words")
```



3 a)

I did somewhat use an object-oriented approach in terms of defining my own class for a play. The fields I used are title, year, acts, scenes, characters, chunks, full body, and statistics. The title is a character string. Acts and scenes simply contain an integer for the number of acts and scenes. Chunks contains the spoken chunks and speaker of each chunk in a list. The full body is one long character string of the body of the play. The statistics are a list of different statistics calculated for the play.

b)

The methods would be the functions I defined in my code: initialization function (Play), Get_Speech(), Calculate_Stats(), and summary(). Finally, I would also implement the plotting as a method as well.