STAT 243 PS 3

Junyuan Gao(SID:26484653)

September 30, 2017

1 Q2

1.1 a

```
# read play text from website
shakespeare <- readLines("http://www.gutenberg.org/cache/epub/100/pg100.txt")
shakespeare <- shakespeare[sapply(shakespeare, nchar) > 0]
# omit first sonnet and last play
front_index <- grep("1603", shakespeare)[1] #before are "The Sonnets" and info
end_index <- grep("1609", shakespeare)[3]-1 #after are "A Lover's complaint"
shakespeare_clean<- shakespeare[front_index : end_index]

# grep the index of year and 'THE END' of the play
year_index <- grep('^[0-9]{4}$', shakespeare_clean)
tail_index <- grep('THE END', shakespeare_clean)

# save the desired play into a list
play <- list()
for (i in 1:length(year_index)){
  play[[i]] <- shakespeare_clean[year_index[i]:tail_index[i]]
}
```

1.2 b

I choose not to generate a list in part(b) and create the list in part(d) see 2(d)iv.

```
# Get year and title of plays
years_of_play <- shakespeare_clean[year_index]
titles_of_play <- shakespeare_clean[year_index+1]
act_scene <- c()
scenes_of_play <- c()
acts_of_play <- c()
```

```
# To get number of acts, write a function to
# grep the last character of "ACT X", convert
# roman to numeric to count
get_numeric <- function(x){
  x= strsplit(x[length(x)], "")[[1]]
  x= x[length(x)]
  return (as.numeric(as.roman(x)))
}
# get number of scenes and acts of plays
for (i in 1:length(year_index)){
  act <- shakespeare_clean[year_index[i]:tail_index[i]
              ][grep('^ACT (I|II|III|IV|V).',
              shakespeare_clean[year_index[i]:tail_index[i]])]
  scene <- shakespeare_clean[year_index[i]:tail_index[i]
              ][grep('SCENE|Scene',
              shakespeare_clean[year_index[i]:tail_index[i]])]
  scenes_of_play[i] <- length(scene)-1

  unique_act <- unique(gsub("SCENE .*"," ", act))
  unique_act <- unique(gsub(" ", "", gsub("\\. .*","", unique_act)))
  acts_of_play[i] <- get_numeric(unique_act)
}
#by checking the result and real text, I find that the scenes/act info
#in 2nd play is mostly of the form ACT_1|SCENE_2, so can't be
#detected in this case, so manually set it as 5
acts_of_play[2]=5

# extract body of plays(for 2c and 2d)
body <- c()
for (i in 1:length(year_index)){
  begin <- grep('SCENE|Scene',  play[[i]])[1]
  body[i]<- paste(play[[i]][begin :length(play[[i]])],collapse = "\n")
}
```

1.3 c

```
# find spoken chunks by paste text between 2 speakers
spoken_text <- list()
length(spoken_text) <- length(year_index)
for (i in 1:length(year_index)){
  k=1
  tempvec<- c()
  #the pattern of speakers is "name."
  #NAME might be 1 or 2 words
  #(1) this line detect speaker pattern in Play 4

#pattern = "([[:upper:]]+\\. [[:upper:]][^A-Z])|
  #(^\\s{2,4}[A-Z]{1}[a-z]+\\. [[:upper:]][^A-Z])"(easier to read in pdf)
  for (j in 1: length(play[[i]])){
    pattern = "([[:upper:]]+\\. [[:upper:]][^A-Z])|(^\\s{2,4}[A-Z]{1}[a-z]+\\. [[:upper:]][^
    if (grepl(pattern, play[[i]][j])){ #(1)
      spoken_text[[i]][k] = tempvec
      tempvec <- c()
      k = k+1
    }
    tempvec <- paste(tempvec, play[[i]][j])
  }
}

for (i in 1:length(year_index)){
  spoken_text[[i]]= spoken_text[[i]][-1]
}

#get speakers and dailogues in plays
speaker_list <- list()
pure_spoken_text<- list()
length(speaker_list) <- length(year_index)
length(pure_spoken_text) <- length(year_index)
# extract word before first "." to get speaker
# extract word after first "." to get spoken text
for (i in 1:length(year_index)){
  for (j in 1:length(spoken_text[[i]])) {
    speaker_list[[i]][j]= gsub('\\. .*$', '', spoken_text[[i]][j])
    pure_spoken_text[[i]][j]=
      sub('.*? .+(\\.)', '', gsub("\\s{4}", "", spoken_text[[i]][j]))
  }
}
```

1.4 d

i. Number of Unique Speakers

```
library(stringr)

speakers <- c()
for(i in 1: length(year_index)){
  speakers[i]= length(unique(speaker_list[[i]]))
}
```

ii. Number of Spoken Chunks

```r
num_spoken_chunk <- list()
for (i in 1:length(year_index)){
  num_spoken_chunk[i] <- length(spoken_text[[i]])
  }
```

iii. For each play, calculate number of sentences, words spoken and average number of words per chunk.

```r
num_sentence <- list()
length(num_sentence) <- length(year_index)
num_word <- list()
length(num_word) <- length(year_index)

ave_word <-c()
num_sentences_play <- c()
word_spoken_play<- c()

for (i in 1:length(year_index)){
  for (j in 1: length(spoken_text[[i]])){
    num_sentence[[i]][j]=
      str_count(pure_spoken_text[[i]][j], "(\\.)|(\\;)|(\\?)")
    num_word[[i]][j]= str_count(pure_spoken_text[[i]][j], '\\w+')
                      -str_count(pure_spoken_text[[i]][j], "\\'")
  }
  # desired variables for 2(d) iii
  num_sentences_play[i]= sum(num_sentence[[i]])
  word_spoken_play[i]= sum(num_word[[i]])
  ave_word[i]= word_spoken_play[i]/num_spoken_chunk[[i]]
}
```

iv. The number of unique words.

```r
unique_words <- list()
length(unique_words) <- length(year_index)
for (i in 1:length(year_index)){
  unique_words[i] = length(unique(str_extract_all(toupper(body[i]),
                                                  "\\w+")[[1]]))
}

#create a data object(linked list) to save results in 2b 2c and 2d
shakespeare_list <- list()
length(shakespeare_list) <- length(year_index)
```

```
for (i in 1: length(year_index)){
  shakespeare_list[[i]] = list(Year = years_of_play[i],
                               Scenes= scenes_of_play[i],
                           Acts= acts_of_play[i], Body= body[i],
                           Unique_speakers= speakers[i],
                           Spoken_chunks= num_spoken_chunk[i],
                           Sentences= num_sentences_play[i],
                           Words_Spoken= word_spoken_play[i],
                           Ave_Word_Per_Chunk= ave_word[i],
                           Unique_words=unique_words[i])
}
names(shakespeare_list)=titles_of_play

attributes(shakespeare_list)

## $names
##  [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"
```

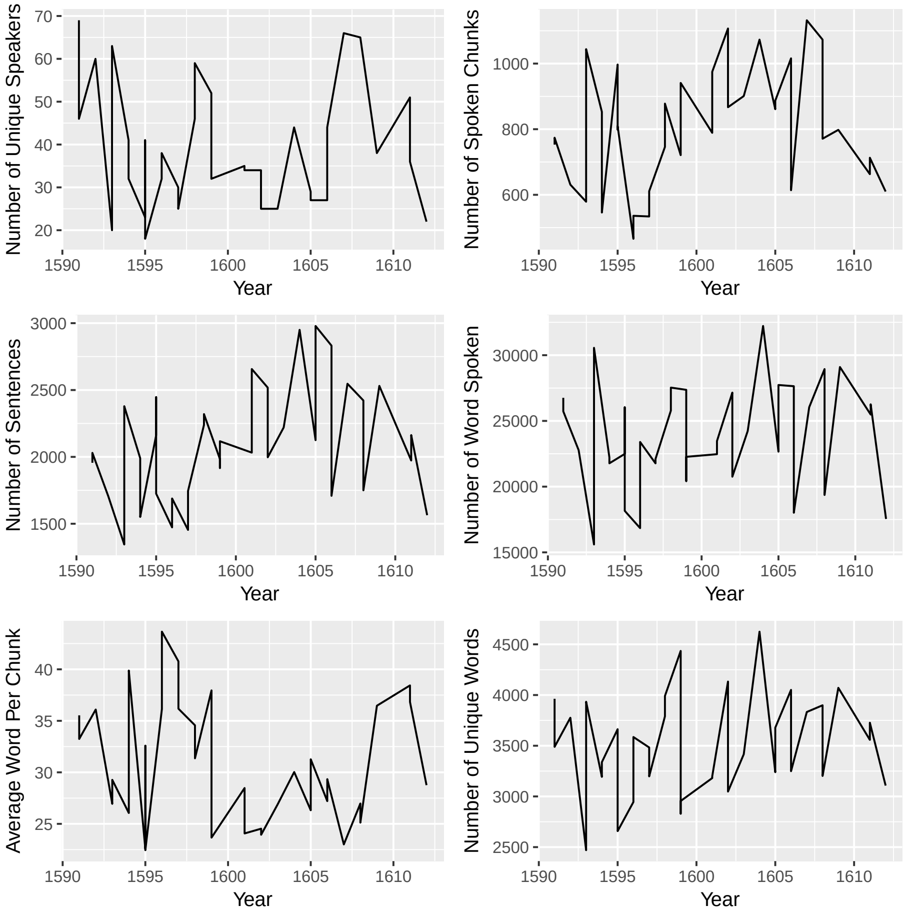
1.5 e

```
library(ggplot2)
#create data frame df_2e for report and ggplot
df_2e <- data.frame(Year = as.numeric(years_of_play),
                    Play_Name= titles_of_play,
                    Number_Acts= acts_of_play,
                    Number_Scenes= scenes_of_play,
                    Unique_speakers= as.numeric(speakers),
                 Spoken_chunks= as.numeric(num_spoken_chunk),
                 Sentences= as.numeric(num_sentences_play),
                 Words_Spoken= as.numeric(word_spoken_play),
                 Ave_Word_Per_Chunk= as.numeric(ave_word),
                 Unique_words= as.numeric(unique_words))

#report summary
summary(df_2e)
```

```
##       Year                            Play_Name   Number_Acts
##  Min.   :1591   A MIDSUMMER NIGHT'S DREAM: 1   Min.   :5
##  1st Qu.:1595   ALLS WELL THAT ENDS WELL : 1   1st Qu.:5
##  Median :1599   AS YOU LIKE IT           : 1   Median :5
##  Mean   :1600   CYMBELINE                : 1   Mean   :5
##  3rd Qu.:1605   KING HENRY THE EIGHTH    : 1   3rd Qu.:5
##  Max.   :1612   KING JOHN                : 1   Max.   :5
##                 (Other)                  :30
##  Number_Scenes   Unique_speakers Spoken_chunks      Sentences
##  Min.   : 9.00   Min.   :18.00   Min.   : 466.0   Min.   :1345
##  1st Qu.:16.75   1st Qu.:28.50   1st Qu.: 655.0   1st Qu.:1740
##  Median :19.50   Median :37.00   Median : 793.5   Median :2031
##  Mean   :20.25   Mean   :40.11   Mean   : 801.3   Mean   :2089
##  3rd Qu.:24.00   3rd Qu.:50.25   3rd Qu.: 911.0   3rd Qu.:2390
##  Max.   :42.00   Max.   :69.00   Max.   :1132.0   Max.   :2979
##
##   Words_Spoken   Ave_Word_Per_Chunk  Unique_words
##  Min.   :15600   Min.   :22.44      Min.   :2470
##  1st Qu.:21774   1st Qu.:26.20      1st Qu.:3197
##  Median :23434   Median :29.31      Median :3524
##  Mean   :23853   Mean   :30.63      Mean   :3515
##  3rd Qu.:26856   3rd Qu.:36.11      3rd Qu.:3850
##  Max.   :32223   Max.   :43.65      Max.   :4625
##

#############################
##############################
#
#Report statistics
df_2e

##    Year                                Play_Name Number_Acts Number_Scenes
## 1  1603                 ALLS WELL THAT ENDS WELL           5            23
## 2  1607      THE TRAGEDY OF ANTONY AND CLEOPATRA           5            42
## 3  1601                           AS YOU LIKE IT           5            22
## 4  1593                     THE COMEDY OF ERRORS           5            11
## 5  1608                THE TRAGEDY OF CORIOLANUS           5            29
## 6  1609                                CYMBELINE           5            27
## 7  1604 THE TRAGEDY OF HAMLET, PRINCE OF DENMARK           5            20
## 8  1598  THE FIRST PART OF KING HENRY THE FOURTH           5            19
## 9  1598             SECOND PART OF KING HENRY IV           5            19
## 10 1599         THE LIFE OF KING HENRY THE FIFTH           5            23
## 11 1592        THE FIRST PART OF HENRY THE SIXTH           5            27
## 12 1591  THE SECOND PART OF KING HENRY THE SIXTH           5            24
## 13 1591   THE THIRD PART OF KING HENRY THE SIXTH           5            28
## 14 1611                    KING HENRY THE EIGHTH           5            17
## 15 1597                                KING JOHN           5            16
## 16 1599             THE TRAGEDY OF JULIUS CAESAR           5            18
## 17 1606                 THE TRAGEDY OF KING LEAR           5            26
## 18 1595                     LOVE'S LABOUR'S LOST           5             9
## 19 1606                   THE TRAGEDY OF MACBETH           5            29
## 20 1605                      MEASURE FOR MEASURE           5            17
## 21 1597                   THE MERCHANT OF VENICE           5            20
## 22 1601               THE MERRY WIVES OF WINDSOR           5            23
## 23 1596                A MIDSUMMER NIGHT'S DREAM           5             9
## 24 1599                   MUCH ADO ABOUT NOTHING           5            17
## 25 1605   THE TRAGEDY OF OTHELLO, MOOR OF VENICE           5            15
## 26 1596                  KING RICHARD THE SECOND           5            19
## 27 1593                         KING RICHARD III           5            25
## 28 1595          THE TRAGEDY OF ROMEO AND JULIET           5            24
## 29 1594                  THE TAMING OF THE SHREW           5            14
## 30 1612                              THE TEMPEST           5             9
## 31 1608              THE LIFE OF TIMON OF ATHENS           5            17
## 32 1594          THE TRAGEDY OF TITUS ANDRONICUS           5            14
## 33 1602      THE HISTORY OF TROILUS AND CRESSIDA           5            24
## 34 1602         TWELFTH NIGHT; OR, WHAT YOU WILL           5            18
## 35 1595              THE TWO GENTLEMEN OF VERONA           5            20
## 36 1611                        THE WINTER'S TALE           5            15
##    Unique_speakers Spoken_chunks Sentences Words_Spoken Ave_Word_Per_Chunk
## 1               25           901      2220        24232           26.89456
## 2               66          1132      2547        26041           23.00442
## 3               35           789      2032        22468           28.47655
## 4               20           579      1345        15600           26.94301
## 5               65          1073      2422        28943           26.97390
## 6               38           798      2531        29100           36.46617
## 7               44          1073      2950        32223           30.03075
## 8               46           746      2239        25786           34.56568
## 9               59           878      2320        27535           31.36105
## 10              52           721      1984        27366           37.95562
## 11              60           631      1705        22776           36.09509
## 12              69           753      1956        26757           35.53386
## 13              46           774      2030        25727           33.23902
## 14              51           663      1973        25484           38.43741
## 15              30           534      1454        21772           40.77154
## 16              50           778      1917        20424           26.25193
## 17              27          1016      2832        27641           27.20571
## 18              23           997      2162        22491           22.55868
## 19              44           614      1709        18014           29.33876
## 20              29           861      2125        22661           26.31940
## 21              25           611      1745        22104           36.17676
## 22              34           975      2657        23468           24.06974
## 23              32           466      1473        16844           36.14592
## 24              32           941      2117        22269           23.66525
## 25              27           887      2979        27733           31.26607
## 26              38           536      1689        23399           43.65485
## 27              63          1044      2379        30561           29.27299
## 28              41           799      2447        26035           32.58448
## 29              41           853      1990        22225           26.05510
## 30              22           610      1564        17542           28.75738
## 31              65           771      1750        19366           25.11803
## 32              32           546      1551        21775           39.88095
## 33              34          1107      2518        27154           24.52936
## 34              25           867      1997        20757           23.94118
## 35              18           809      1725        18158           22.44499
## 36              36           713      2163        26262           36.83310
##    Unique_words
## 1          3416
## 2          3833
## 3          3181
## 4          2470
## 5          3899
## 6          4071
## 7          4625
## 8          3790
## 9          3991
## 10         4435
## 11         3776
## 12         3963
## 13         3489
## 14         3558
## 15         3483
## 16         2830
## 17         4051
## 18         3663
## 19         3249
## 20         3239
## 21         3198
## 22         3198
## 23         2945
## 24         2955
## 25         3678
## 26         3586
## 27         3934
## 28         3628
## 29         3193
## 30         3107
## 31         3202
## 32         3338
## 33         4133
## 34         3048
## 35         2658
## 36         3727
```

```
#produce plots of summary statistics in 2(d)
p1 <- ggplot(data=df_2e, aes(y=Unique_speakers, x=Year))+
  ylab("Number of Unique Speakers")+
  geom_line()

p2 <- ggplot(data=df_2e, aes(y=Spoken_chunks, x=Year))+
  ylab("Number of Spoken Chunks")+
  geom_line()

p3 <- ggplot(data=df_2e, aes(y=Sentences, x=Year))+
  ylab("Number of Sentences")+
  geom_line()
```

```r
p4 <- ggplot(data=df_2e, aes(y=Words_Spoken, x=Year))+
  ylab("Number of Word Spoken")+
  geom_line()

p5 <- ggplot(data=df_2e, aes(y=Ave_Word_Per_Chunk, x=Year))+
  ylab("Average Word Per Chunk")+
  geom_line()

p6 <- ggplot(data=df_2e, aes(y=Unique_words, x=Year))+
  ylab("Number of Unique Words")+
  geom_line()

# print 6 plots together
library(gridExtra)
grid.arrange(p1,p2,p3,p4,p5,p6, ncol=2)
```



There is no significant trend in plots except plot 1. In plot 1, the number of unique speakers in plays seems to have a period of 5 year. Possible explanation might be the small size of data(only 36). If more observation is available, there might be some trend detected.

2 Q3

2.1 a

Following pseudocode show the fields and methods of the class ”shakespeare”

```
library(methods)
setClass("shakespeare",
         representation(
           year = "numeric",  #requirements for 2b
           title = "character",
           number_of_acts = "numeric",
           number_of_scenes = "numeric",
           body = "character",

           spokenText = "matrix",  #requirements for 2c
           speaker="list"

           number_unique_speakers = "numeric",  #requirements for 2d (i)
           number_of_chunks = "numeric",   #requirements for 2d (ii)
           number_of_sentence = "numeric",  #requirements for 2d (iii)
           number_of_word = "numeric",  #requirements for 2d (iii)
           ave_word = "numeric", #requirements for 2d (iii)
           numer_of_unique_word = "numeric" #requirements for 2d (iv)
         )
        methods=list(
          get_title = function()(x),
          count_scene = function()(x),
          count_act = function()(x),
          get_SpokenText = function()(x),
          get_speaker = function()(x),
          count_speaker = function()(x),
        )
)
```

Just as it’s illustrated above, those fields indicates to desired variables and those methods works similar to the functions to get those desired variables.

2.2 b

```
# 1. "get_title()" is designed to get the title of play. It is a method
# processing to play which takes a the whole text file as input(many strings)
# and creates the "title" field. Its output is a string of characters.

# 2. "count_scene()" is designed to count the number of scene of plays.
# It is a method providing play info which takes a the whole text file as
# input(many strings) and creates the " number_of_scenes" field. Its output
# is a vector of numerics.

# 3. "count_act()" is designed to count the number of acts of plays. It is a
# method providing play info which takes a the whole text file as
# input(many strings) and creates the " number_of_acts" field. Its output
# is a vector of numerics.

# 4. "get_SpokenText()" is designed to get the spoken text of play.
# It is a method processing to play which takes a the a list of
# character strings as input and modifies the "SpokenText" field.
# Its output is a character matrix.

# 5. "get_speaker()" is designed to get the speaker of play. It is a
# method processing to play which takes the body of plays(a list of large
# string of characters) as input and modifies the "speaker" field. Its
# output is a list of characterstrings.

# 6. "count_speaker()" is designed to count the number of unique speakers
# of plays.  It is a method providing play info which takes field "speaker"
# as input and creates the " number_unique_speakers" field. Its output is
# a vector of numerics.
```