STAT243: Problem Set 3

Zicheng Huang

09/29/2017

```r
library(knitr)
## required package
library(ggplot2)
## store the Shakeapeare txt as 'shakespeare' with readLines function
url <- "http://www.gutenberg.org/cache/epub/100/pg100.txt"
## shakespeare file contains all lines in the txt file and stores each
## line as a character string
shakespeare <- readLines(url)
```

Question 2 and 3 will be answered together with a newly defined reference class of "ShakespeareClass".

```r
## define a reference class
ShakespeareClass <- setRefClass(

  ## name of the new reference class
  "ShakespeareClass",

  ## define fields in the new reference class with
  ## corresponding type of the object
  fields = list(

    ## store the readin file
    Whole = "character",
    ## lines in each of the play
    Plays = "character",
    ## number of plays in the txt file
    PlaysNum = "integer",
    ## index for each play
    PlaysIndex = "integer",
    ## location of the year of each play
    YearsIndex = "integer",
    ## year of each play
    Years = "numeric",
    ## location of the author of each play
    AuthorsIndex = "integer",
    ## location of the title of each play
    TitlesIndex = "numeric",
    ## title of each play
    Titles = "character",
    ## location of the start of the body of each play
    StartsIndex = "integer",
    ## location of the end of each play
    EndsIndex = "integer",
    ## body text of each play
    Bodies = "character",
    ## number of acts in each play
    ActsNum = "numeric",
    ## number of scenes in each play
    ScenesNum = "integer",
    ## meta information of each play for part b, including the year of
    ## the play, the title, the number of acts, the number of scenes
    MetaB = "list",
    ## meta information of each play for part c, including infomration
    ## in Meta_B with additional chunks of spoken text with speaker
    MetaC = "list",
    ## number of unique speakers in each play
    SpeakersNum = "integer",
    ## number of chunks in each play
    ChunksNum = "integer",
    ## number of sentences in each play
    SentencesNum = "integer",
    ## number of words spoken in each play
    WordsNum = "integer",
    ## number of words spoken per chunk in each play
    WordsPerChunkNum = "numeric",
    ## number of unique words in each play
    WordsUniqueNum = "integer"

    ),

  ## define the methods in the new reference class
  methods = list(

    ## define the initial object in some fields and
    ## some methods that will be called initially
    initialize = function(Whole = shakespeare, ...) {
      ## required pacage for string processing
      require(stringr)
      ## store the readin file into 'Whole' field
      Whole <<- Whole
      ## extract the location of year of each play
      YearsIndex <<- grep('^[0-9]{4}$', Whole)[2:37]
      ## extract the year of each play
      Years <<- as.numeric(Whole[YearsIndex])
      ## extract the location of the author of each play
      AuthorsIndex <<- grep('William Shakespeare', Whole)[7:42]
      ## extract the location of the title of each play
      titles_index_tmp <- AuthorsIndex - 2
      titles_index_tmp[c(7,8,19,24,26)] <- titles_index_tmp[c(7,8,19,24,26)] - 1
      TitlesIndex <<- titles_index_tmp
      ## extract the title of each play
      Titles <<- Whole[TitlesIndex]
      ## extract the location of the scene information of each play,
      ## which is also where the play starts
      StartsIndex <<- grep('^SCENE:|^Scene:|^ +SCENE:|^SCENE\\.|^ +SCENE\\.', Whole)
      ## extract the locatino of the end of each play
      EndsIndex <<- grep('THE END', Whole)[2:37]
      ## extract the number of plays in the readin file
      PlaysNum <<- length(YearsIndex)
      # create a sequence containing the index of each play, 1 to 36
      PlaysIndex <<- seq(1:PlaysNum)
      ## method that extract the plays into a character vector
      extractEachPlay()
      ## method that extract the body of each play
      extractBodies()
      ## method that extract the number of acts in each play
      extractActsNum()
      ## method that extract the number of scenes in each play
      extractScenesNum()
      ## method that extract the meta information of each play
      extractMetaInformation()
      ## method that extract the spoken chunks in each play
      extractChunksAllPlays()
      ## method that extract the number of speakers in each play
      extractNumOfSpeakersAllPlays()
      ## method that extract the number of spoken chunks in each play
      extractNumOfChunksAllPlays()
      ## method that extract the number of sentences in each play
      extractNumOfSentencesAllPlays()
      ## method that extract the number of words in each play
      extractNumOfWordsAllPlays()
      ## method that extract the average number of words in each chunk,
      ## obtained by dividing the number of words in each play by the
      ## number of spoken chunks in each play
      WordsPerChunkNum <<- WordsNum / ChunksNum
      ## method that extract the number of unique words in each play
      extractUniqueWordsAllPlays()

    },

    extractEachPlay = function() {
      plays <- c()
      for (i in 1:PlaysNum) {
        play <- paste(Whole[YearsIndex[i]:EndsIndex[i]], collapse = '\n')
        ## vector that stored all plays
        plays <- c(plays, play)
      }
      ## define the field 'Plays'
      Plays <<- plays
    },

    extractBodies = function() {
      bodies <- c()
      for (i in 1:PlaysNum) {
        body <- paste(Whole[StartsIndex[i]:EndsIndex[i]], collapse = '\n')
        ## vector that stored all body tests
        bodies <- c(bodies, body)
      }
      ## define the field 'Bodies'
      Bodies <<- bodies
    },

    extractActsNum = function() {
      acts_num <- c()
      for (i in 1:PlaysNum) {
        play <- paste(Whole[StartsIndex[i]:EndsIndex[i]], collapse = '\n')
        if (str_detect(play, 'ACT V|ACE_5')) {
          act_num <- 5
          ## vector that stored number of acts
          acts_num <- c(acts_num, act_num)
        }
      }
      ## define the field 'ActsNum'
      ActsNum <<- acts_num
    },

    extractScenesNum = function() {
      scenes_num <- c()
      for (i in 1:PlaysNum) {
        scene_num <- length(grep('^ACT.+SCENE|SCENE |^ACT.+Scene|Scene ',
                                 Whole[StartsIndex[i]:EndsIndex[i]]))
        ## vector stored number of scenes
        scenes_num <- c(scenes_num, scene_num)
      }
      ## define the field 'ScenesNum'
      ScenesNum <<- scenes_num
    },

    extractMetaInformation = function() {
      meta <- list()
      for (i in 1:PlaysNum) {
        ## create list containing meta information
        meta[[Titles[i]]] <- list(Year = Years[i], NumOfActs = ActsNum[i],
                                  NumOfScenes = ScenesNum[i], BodyText = Bodies[i])
      }
      ## define the filed 'MetaB'
      MetaB <<- meta
    },

    ## methods that extract spoken chunks for one play
    extractChunksOnePlay = function(x) {
      ## the body text of play
      onePlay <- Whole[StartsIndex[x]:EndsIndex[x]]
      ## deal with the exceptiions in the fourth play
      if (x == 4) {
        ## index for all spoken text
        text_index <-
          grep('^[A-Z]{1,}[^a-z]{0,}\\.[[:space:]][A-Z]|^[[:space:]]{2,4}.+', onePlay)
        ## edge cases
        edge_text_index <- str_detect(onePlay[text_index], '^ACT')
        ## actual spoken text
        text_all <- (onePlay[text_index])[!edge_text_index]
        ## index for lines containing speaker's name
        text_speakers_index_tmp <-
          grep('^[A-Z]{1,}[^a-z]{0,}\\.[[:space:]][A-Z]', onePlay)
        ## edge cases
        edge_text_speakers_index <- str_detect(onePlay[text_speakers_index_tmp], '^ACT')
        ## actual locations containing speaker's name
        text_speakers_index <- text_speakers_index_tmp[!edge_text_speakers_index]
      }
      ## other plays follow similar pattern
      else {
        ## index for all spoken text
        text_index <- grep('^[[:space:]]{2,}.{1,}', onePlay)
        ## actuall spoken text
        text_all <- onePlay[text_index]
        ## index for lines containing speaker's name
        text_speakers_index <- grep('^[[:space:]]{2}[A-Za-z]{1,}[^a-z]{0,}\\.', onePlay)
      }
      ## extract chunks of the corresponding play to a character vector
      x <- str_replace(onePlay, '^[[:space:]]{2,4}', '')
      chunks_all <- c()
      lst <- list()
      for (i in 1:length(text_speakers_index)) {
        if (!(i == length(text_speakers_index))) {
          n <- text_speakers_index[i + 1] - text_speakers_index[i] - 1
          start <- text_speakers_index[i]
          end <- text_speakers_index[i] + n
          chunk <- paste(x[start:end], collapse = ' ')
          chunks_all <- c(chunks_all, chunk)
        }
        else {
          n <- text_index[length(text_index)] - text_speakers_index[i]
          start <- text_speakers_index[i]
          end <- text_speakers_index[i] + n
          chunk <- paste(x[start:end], collapse = ' ')
          chunks_all <- c(chunks_all, chunk)
        }
      }
      ## stored the chunks in a list
      for (i in 1:length(text_speakers_index)) {
        lst[i] <- chunks_all[i]
      }
      ## the method will return the list of chunks
      lst
    },

    extractChunksAllPlays = function() {
      META <- MetaB
      ## call the previous method on each play
      for (i in 1:PlaysNum) {
        ## list stored the meta information including spoken chunks
        META[[Titles[i]]] <- append(
          MetaB[[Titles[i]]],
```

```r
      list(AllSpokenText = extractChunksOnePlay(i))
    )
  }
  ## define the field 'MetaC'
  MetaC <<- META
},

# method that extract number of speakers in one play
extractNumOfSpeakersOnePlay = function(i) {
  ## the body text of play
  onePlay <- Whole[StartsIndex[i]:EndsIndex[i]]
  ## deal with the exceptiions in the fourth play
  if (i == 4) {
    ## index for lines containing speaker's name
    text_speakers_index_tmp <-
      grep('^[A-Z]{1,}[^a-z]{0,}\\.[[:space:]][A-Z]', onePlay)
    ## edge cases
    edge_text_speakers_index <-
      str_detect(onePlay[text_speakers_index_tmp], '^ACT')
    ## actual locations containing speaker's name
    text_speakers_index <- text_speakers_index_tmp[!edge_text_speakers_index]
    ## all speakers appeared
    speakers <- na.omit(str_extract(onePlay[text_speakers_index],
                                    '^[A-Za-z]{1,}[^a-z]{0,}\\.'))
    ## unique speakers appeared
    speakers_unique <- unique(str_replace(speakers, '\\.', ''))
    ## number of unique speakers
    num <- length(speakers_unique)
  }
  ## all other plays follow similar pattern
  else {
    ## all speakers appeared
    speakers <- na.omit(str_extract(onePlay,
                                    '^[[:space:]]{2}[A-Za-z]{1,}[^a-z]{0,}\\.'))
    ## unique speakers appeared
    speakers_unique <-
      unique(str_replace(str_replace(speakers, '[[:space:]]{2}', ''), '\\.', ''))
    ## number of unique speakers
    num <- length(speakers_unique)
  }
},

extractNumOfSpeakersAllPlays = function() {
  speakers_num <- c()
  ## call the previous method on each play
  for(i in 1:PlaysNum) {
    num <- extractNumOfSpeakersOnePlay(i)
    ## vector containing number of speakers
    speakers_num = c(speakers_num, num)
  }
  ## define the field 'SpeakersNum'
  SpeakersNum <<- speakers_num
},
## method that extract the number of chunks in one play
extractNumOfChunksOnePlay = function(i) {
  ## body text
  onePlay <- Whole[StartsIndex[i]:EndsIndex[i]]
  ## deal with exceptions in the fourth play
  if (i == 4) {
    ## index for lines containing speaker's name
    text_speakers_index_tmp <-
      grep('^[A-Z]{1,}[^a-z]{0,}\\.[[:space:]][A-Z]', onePlay)
    ## edge cases
    edge_text_speakers_index <- str_detect(onePlay[text_speakers_index_tmp], '^ACT')
    ## actual locations containing speaker's name
    text_speakers_index <- text_speakers_index_tmp[!edge_text_speakers_index]
    ## the number of lines containing speaker's name,
    ## same as number of spoken chunks
    num <- length(text_speakers_index)
  }

  else {
    # index for lines containing speaker's name
    text_speakers_index <- grep('^[[:space:]]{2}[A-Za-z]{1,}[^a-z]{0,}\\.', onePlay)
    ## number of spoken chunks
    num <- length(text_speakers_index)
  }
},

extractNumOfChunksAllPlays = function() {
  chunks_num <- c()
  ## call the previous method on each play
  for (i in PlaysIndex) {
    num <- extractNumOfChunksOnePlay(i)
    ## vector containing number of chunks
    chunks_num = c(chunks_num, num)
  }
  ## define the field 'ChunksNum'
  ChunksNum <<- chunks_num
},

## method that extract number of sentences in one play
extractNumOfSentencesOnePlay = function(i) {
  ## body text
  onePlay <- Whole[StartsIndex[i]:EndsIndex[i]]
  ## deal with the fourth play
  if (i == 4) {
    ## index for lines containing speaker's name
    text_speakers_index_tmp <-
      grep('^[A-Z]{1,}[^a-z]{0,}\\.[[:space:]][A-Z]', onePlay)
    ## edge cases
    edge_text_speakers_index <- str_detect(onePlay[text_speakers_index_tmp], '^ACT')
    ## actual locations containing speaker's name
    text_speakers_index <- text_speakers_index_tmp[!edge_text_speakers_index]
    ## number of sentences in one play
    num <- sum(str_count(onePlay, '\\!|\\.|\\?')) - length(text_speakers_index)
  }

  else {
    ## index for lines containing speaker's name
    text_speakers_index <- grep('^[[:space:]]{2}[A-Za-z]{1,}[^a-z]{0,}\\.', onePlay)
    ## number of sentences in one play
    num <- sum(str_count(onePlay, '\\!|\\.|\\?')) - length(text_speakers_index)
  }
},

extractNumOfSentencesAllPlays = function() {
  sentences_num <- c()
  ## call the previous method on each play
  for (i in PlaysIndex) {
    num <- extractNumOfSentencesOnePlay(i)
    ## vector containing number of sentences of all plays
    sentences_num <- c(sentences_num, num)
  }
  ## define the field 'SentencesNum'
  SentencesNum <<- sentences_num
},

## method that extract number of words in one play
extractNumOfWordsOnePlay = function(i) {
  ## body text
  onePlay <- Whole[StartsIndex[i]:EndsIndex[i]]
  ## deal with exceptions in patterns in the fourth play
  if (i == 4) {
    # index for all spoken text
    text_index <-
      grep('^[A-Z]{1,}[^a-z]{0,}\\.[[:space:]][A-Z]|^[[:space:]]{2,4}.+', onePlay)
    # edge cases
    edge_text_index <- str_detect(onePlay[text_index], '^ACT')
    # actual spoken text
    text_all <- (onePlay[text_index])[!edge_text_index]
  }
  else {
    # index for all spoken text
    text_index <- grep('^[[:space:]]{2,}.{1,}', onePlay)
    # actuall spoken text
    text_all <- onePlay[text_index]
  }
  ## clean up the extracted spoken text in 'text_all'
  a <- str_replace_all(text_all, "[[:space:]]{2,}", ' ')
  b <- str_replace_all(a, '^ ', '')
  c <- str_replace_all(b, "[^[A-Za-z]' ]", '')
  ## split the spoken text which is delimited by single
  ## space after the clean up
  words_all <- unlist(strsplit(c, ' '))
  ## number of words in one play
  words_all_num <- length(words_all)
  ## return the number of words
  words_all_num
```

```r
  },

  extractNumOfWordsAllPlays = function() {
    words_num <- c()
    ## call previous method on all plays
    for (i in PlaysIndex) {
      num <- extractNumOfWordsOnePlay(i)
      ## vector containing number of words in all plays
      words_num <- c(words_num, num)
    }
    ## define the field 'WordsNum'
    WordsNum <<- words_num
  },

  ## method that extract the unique words in one play
  extractUniqueWordsOnePlay = function(i) {
    ## body text
    onePlay <- Whole[StartsIndex[i]:EndsIndex[i]]
    ## deal with exceptinos in the fourth play
    if (i == 4) {
      # index for all spoken text
      text_index <-
        grep('^[A-Z]{1,}[^a-z]{0,}\\.[[:space:]][A-Z]|^[[:space:]]{2,4}.+', onePlay)
      # edge cases
      edge_text_index <- str_detect(onePlay[text_index], '^ACT')
      # actual spoken text
      text_all <- (onePlay[text_index])[!edge_text_index]
    }
    else {
      # index for all spoken text
      text_index <- grep('^[[:space:]]{2,}.{1,}', onePlay)
      # actuall spoken text
      text_all <- onePlay[text_index]
    }
    ## clean up the extracted spoken text in 'text_all'
    a <- str_replace_all(text_all, "[[:space:]]{2,}", ' ')
    b <- str_replace_all(a, '^ ', '')
    c <- str_replace_all(b, "[^[A-Za-z]' ]", '')
    ## split the spoken text intwo words
    words_all <- unlist(strsplit(c, ' '))
    ## convert to all lower case and keep unique words
    words_unique <- unique(tolower(words_all))
    ## number of unique words in one play
    words_unique_num <- length(words_unique)
    ## reutrn the number of unique words
    words_unique_num
  },

  extractUniqueWordsAllPlays = function() {
    words_unique_num = c()
    ## call the previous method on all plays
    for (i in PlaysIndex) {
      num <- extractUniqueWordsOnePlay(i)
```

```
        ## vector containing number of uniques words in all plays
        words_unique_num = c(words_unique_num, num)
      }
      ## define the field 'WordsUniqueNum'
      WordsUniqueNum <<- words_unique_num
    },

    ## information about objct with the calss 'ShakespeareClass'
    show = function() {
      cat("Object of class 'ShakespeareClass' with ",
          PlaysNum, " plays by William Shakespeare.\n", sep = '')
    }
  )
)
```

2.(a)

```
## create a object 'S' with the class of 'ShakespeareClass'
S <- ShakespeareClass$new(shakespeare)

## Loading required package:  stringr
```

The body text of each play can be viewed by the following command:

```
## body texts of the plays
S$Bodies
```

2.(b) The meta information of each play can be view by the following command:

```
## meta stored in the field 'MetaB'
S$MetaB
```

The attributes

```
## attributed
attributes(S$MetaB[[1]])

## $names
## [1] "Year"        "NumOfActs"   "NumOfScenes" "BodyText"
```

2.(c) The updated meta information of each play containing spoken chunks can be view by the following command:

```
## meta stored in the field 'MetaC'
S$MetaC
```

The attributes

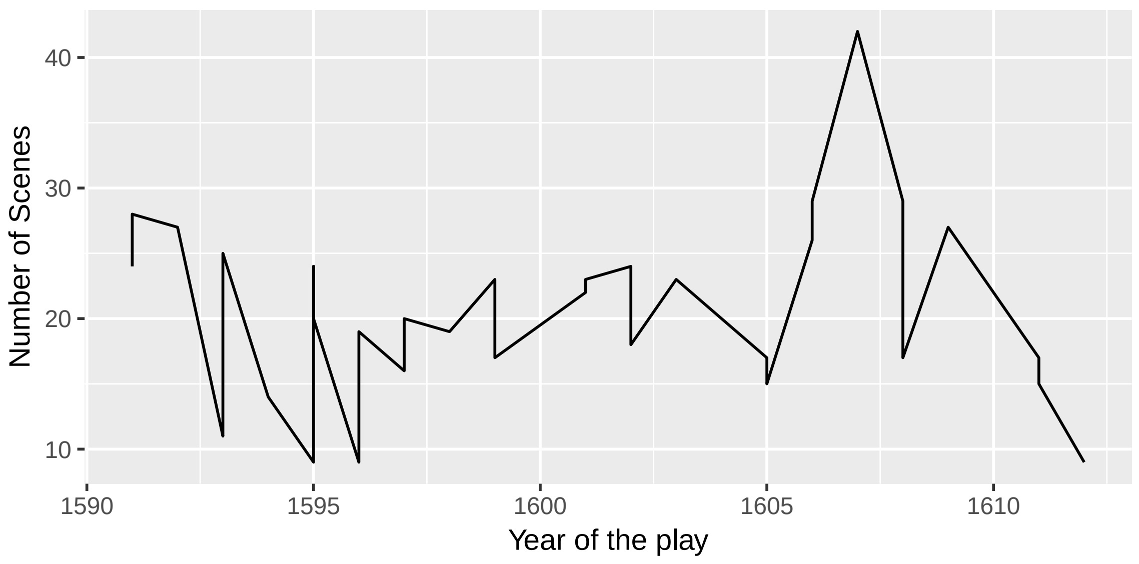
```
## attributed
attributes(S$MetaC[[1]])

## $names
## [1] "Year"          "NumOfActs"     "NumOfScenes"   "BodyText"
## [5] "AllSpokenText"
```

2.(d) The desired information of each play can be view by the following commands:

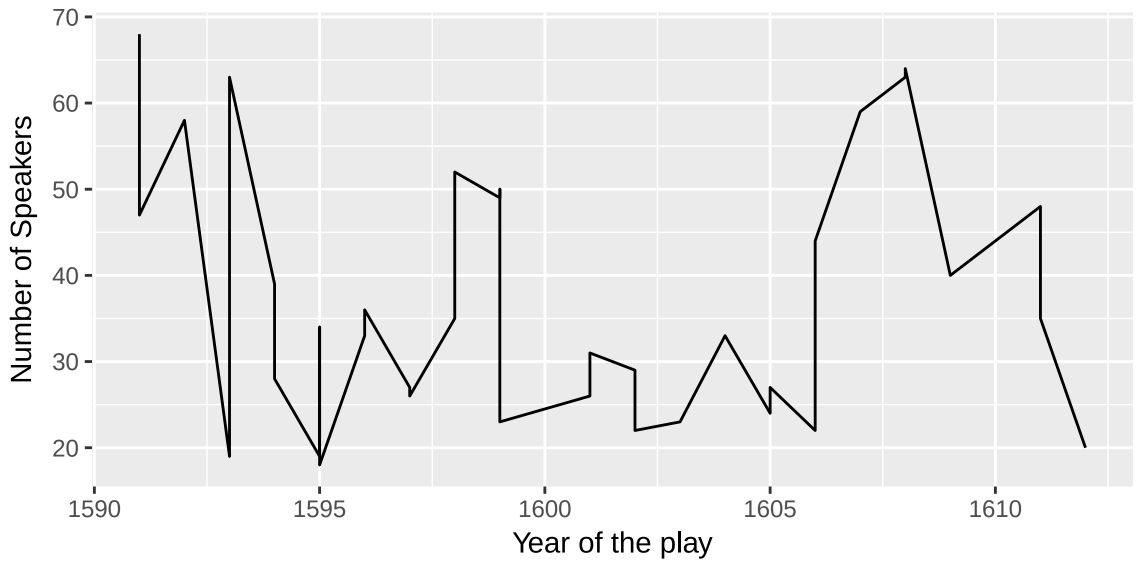
```
## number of speakers
S$SpeakersNum

##  [1] 23 59 26 19 63 40 33 35 52 49 58 68 47 48 27 50 22 19 44 24 26 31 33
## [24] 23 27 36 63 34 39 20 64 28 29 22 18 35

## number of spoken chunks
S$ChunksNum

##  [1]  933 1172  806  582 1106  855 1119  755  904  718  659  793  816  704
## [15]  548  793 1053 1044  643  896  632 1019  504  955 1181  552 1073  813
## [29]  891  642  800  564 1141  920  857  744

## number of sentences
S$SentencesNum

##  [1] 1755 2374 1570 1114 2145 2208 2846 2173 1995 1546 1486 1680 1706 1641
## [15] 1251 1859 2887 1804 1692 1701 1390 2142 1192 2028 2699 1362 2155 2539
## [29] 1684 1290 1640 1348 2315 1685 1406 1748

## number of words
S$WordsNum

##  [1] 23892 26114 22437 15397 28611 28234 31537 25281 26870 26616 22372
## [12] 26118 25165 25011 21314 20387 27041 22648 17676 22565 21864 22985
## [23] 16961 22070 27580 22822 30400 25366 21879 17180 19237 21154 27063
## [34] 20833 17980 25649

## number of words per chunk
S$WordsPerChunkNum

##  [1] 25.60772 22.28157 27.83747 26.45533 25.86890 33.02222 28.18320
##  [8] 33.48477 29.72345 37.06964 33.94841 32.93569 30.83946 35.52699
## [15] 38.89416 25.70870 25.67996 21.69349 27.48989 25.18415 34.59494
## [22] 22.55643 33.65278 23.10995 23.35309 41.34420 28.33178 31.20049
## [29] 24.55556 26.76012 24.04625 37.50709 23.71867 22.64457 20.98016
## [36] 34.47446

## number of unique words
S$WordsUniqueNum

##  [1] 3632 4068 3350 2544 4139 4356 4906 3970 4202 4639 3939 4162 3683 3783
## [15] 3660 2898 4259 3897 3366 3377 3364 3374 3048 3103 3810 3778 4217 3844
## [29] 3365 3302 3352 3469 4397 3216 2799 3994
```

2.(e) We plot some of the summary statistics calculated above over time.

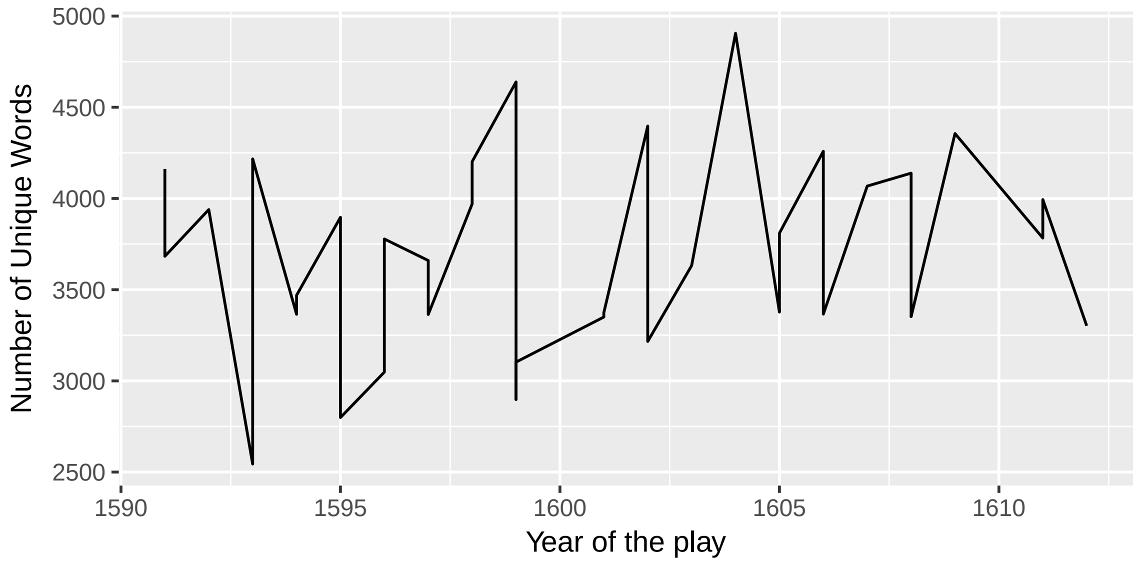
```r
## create a data.frame with information of years, number of acts, number of scenes,
## number of speakers, number of spoken chunks, number of sentences, number of
## words, number of word per chunk, number of unique words
df <- data.frame(years = S$Years, acts = S$ActsNum, scenes = S$ScenesNum,
                 speakers = S$SpeakersNum, chunks = S$ChunksNum,
                 sentences = S$SentencesNum, words = S$WordsNum,
                 avgwords = S$WordsPerChunkNum, uniquewords = S$WordsUniqueNum)
## make some plots
## number of scenes over years
g1 <- ggplot(df) + geom_line(aes(x = years, y = scenes)) +
  xlab("Year of the play") +
  ylab("Number of Scenes")
## number of speakers over years
g2 <- ggplot(df) + geom_line(aes(x = years, y = speakers)) +
  xlab("Year of the play") +
  ylab("Number of Speakers")
## number of unique words over years
g3 <- ggplot(df) + geom_line(aes(x = years, y = uniquewords)) +
  xlab("Year of the play") +
  ylab("Number of Unique Words")
## number of scenes over years
g1
```



```r
## number of speakers over years
g2
```



```
## number of unique words over years
g3
```



According to the plots above, we see that there does not exist an obvious trends.