Problem Set 3

Stephanie Kim

October 1, 2014

/raggedright

Part (a)

```
library(stringr)
library(XML)

## Loading required package:  methods

library(bitops)
library(RCurl)

## get the source code of the URL
html<-getURL("http://www.presidency.ucsb.edu/sou.php")

## this command gets rid of unnecessary spaces
clean<-htmlParse(html)

## grep the parts that starts with <td and ends with a> and that the class is doclist
vector<-xpathSApply(clean,"//td[@class='doclist']//a")

## grep the parts that come after href
url<-sapply(vector, xmlGetAttr,"href")

## get rid of duplicate elements
url<-unique(url)

## get rid of the first element since the it is not a url
url<-url[2:length(url)]
```

Part(b)˜(f)

```
speechinfo<-function(u){

## (b)
## returns the body and title of the speech

        htmlspeech<-getURL(u)
        cleanspeech<-htmlParse(str_replace_all(htmlspeech,"<p>", "\n\n"))
```

```r
    speechbody<-xpathSApply(cleanspeech,"//span[@class='displaytext']",xmlValue)
    speechtitle<-xpathSApply(cleanspeech, "//meta[@name='title']")
    speechtitle<-sapply(speechtitle,xmlGetAttr,"content")

## (c)
## this function returns the count of laughter and applause
## it also returns the body of the speech with laughter and applause eliminated

    laughter <- str_count(speechbody,ignore.case("\\[Laughter\\]"))
    applause <- str_count(speechbody,ignore.case("\\[Applause\\]"))
    stripout <- str_replace_all(speechbody,ignore.case("\\[Laughter\\][[:space:]]"),"\\1")
    stripout <- str_replace_all(speechbody,ignore.case("\\[Applause\\][[:space:]]"),"\\1")

## (d)
## this function returns a vector of which each element is the words of the speech
## it also returns the number of words

    extractwords <- unlist(strsplit(speechbody," "))
    countwords <- length(extractwords)

## (e)
## this function returns a vector of which each element is the sentences of the speech
  extractsentence <- unlist(strsplit(speechbody,"\\."))

## (f) i,ii
## this function returns the number of words and sentences
## it also returns the average length of sentences and words

    extractwords <- unlist(strsplit(speechbody," "))
    extractsentence <- unlist(strsplit(speechbody,"\\."))
    countwords <- length(extractwords)
    countsentence <- length(extractsentence)
    averagesentences <- countwords/countsentence
    averagewords <- sum(nchar(extractwords))/countwords

## (f) iii
## this function returns the number of the word stems we are looking for

    I <- str_count(speechbody,ignore.case("[^[:alpha:]]I[^[:alpha:]]"))
    we <- str_count(speechbody,"[^[:alpha:]][Ww]e[^[:alpha:]]")
    america <- str_count(speechbody,"[^[:alpha:]][Aa]merica[n]?[s]?[^[:alpha:]]")
    democra <- str_count(speechbody,"[^[:alpha:]][Dd]emocracy[^[:alpha:]]|[^[:alpha:]][Dd]emocratic
    republic <- str_count(speechbody,"[^[:alpha:]][Rr]epublic[^[:alpha:]]")
    democrat <- str_count(speechbody,"[^[:alpha:]][Dd]emocrat[s]?[^[:alpha:]]|[^[:alpha:]][Dd]emocra
    republican <- str_count(speechbody,"[^[:alpha:]][Rr]epublican[s]?[^[:alpha:]]")
    free <- str_count(speechbody,"[^[:alpha:]][Ff]ree[^[:alpha:]]|[^[:alpha:]][Ff]reedom[^[:alpha:]]
    war <- str_count(speechbody,"[^[:alpha:]][Ww]ar?[^[:alpha:]]")
    god <- str_count(speechbody,"[^[:alpha:]][Gg]od[^[:alpha:]][^bless]")
    godbless <- str_count(speechbody,"[^[:alpha:]][Gg]od[^[:alpha:]]bless")
    jcc <- str_count(speechbody,"[^[:alpha:]]Jesus[^[:alpha:]]|[^[:alpha:]]Christ[^[:alpha:]]|[^[:al
    income <- str_count(speechbody,"[^[:alpha:]][Ii]ncome[^[:alpha:]]")
    labor <- str_count(speechbody,"[^[:alpha:]][Ll]abor[^[:alpha:]]")
```

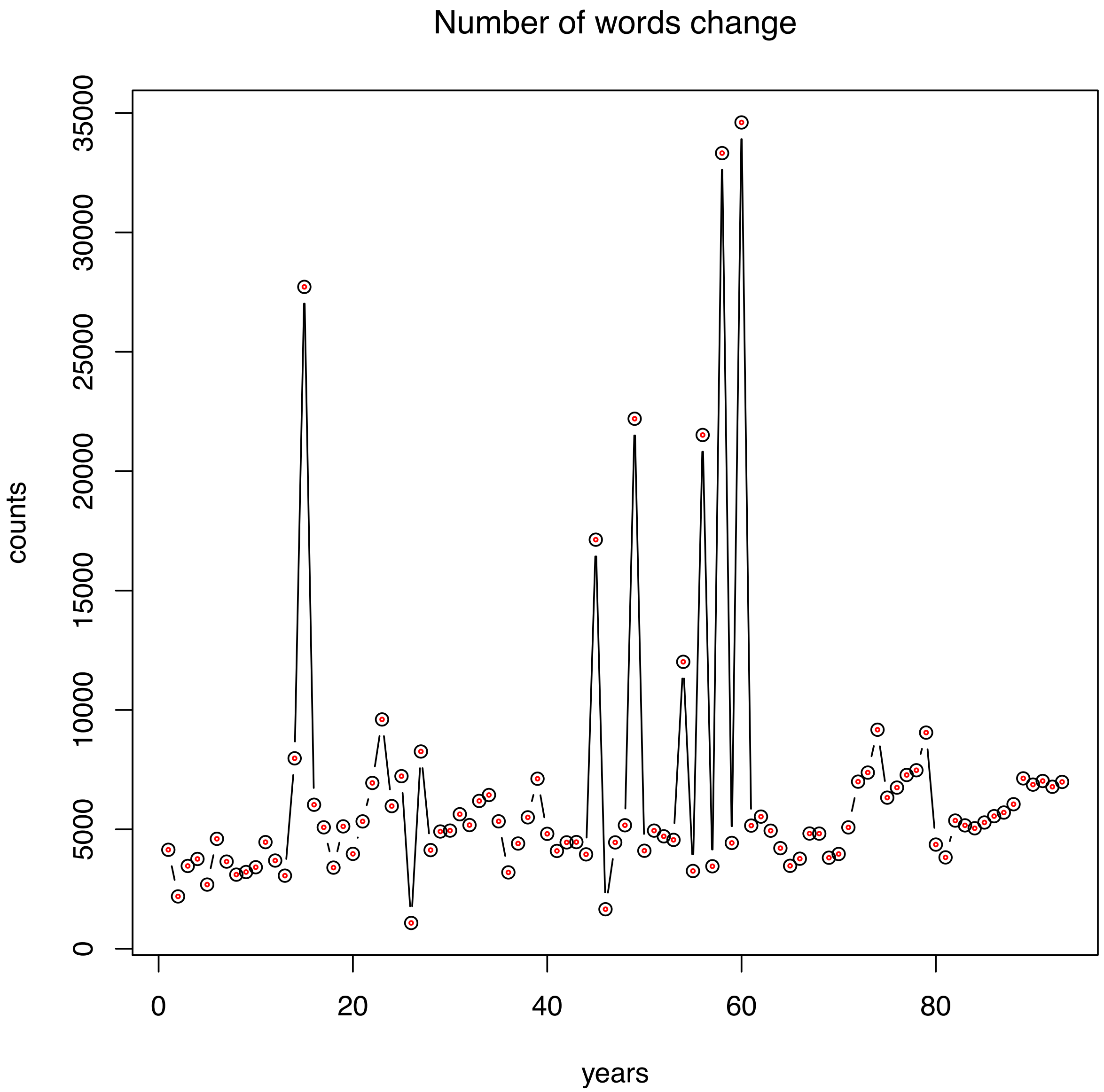
```
        list<-list(Body=speechbody,Title=speechtitle,Laugh=laughter,Appl=applause,Strip=stripout,
          Extractw=extractwords,Extracts=extractsentence,CW=countwords, CS=countsentence,Averages=averag
                  A=I, B=we, C=america, Demo=democra, Repu=republic, Democra=democrat, Republica=republi
                  Fr=free, Wa=war, Go=god, Godbl=godbless, Jccc=jcc, IC=income, LB=labor)

    return(list)

}
```

Part (h)

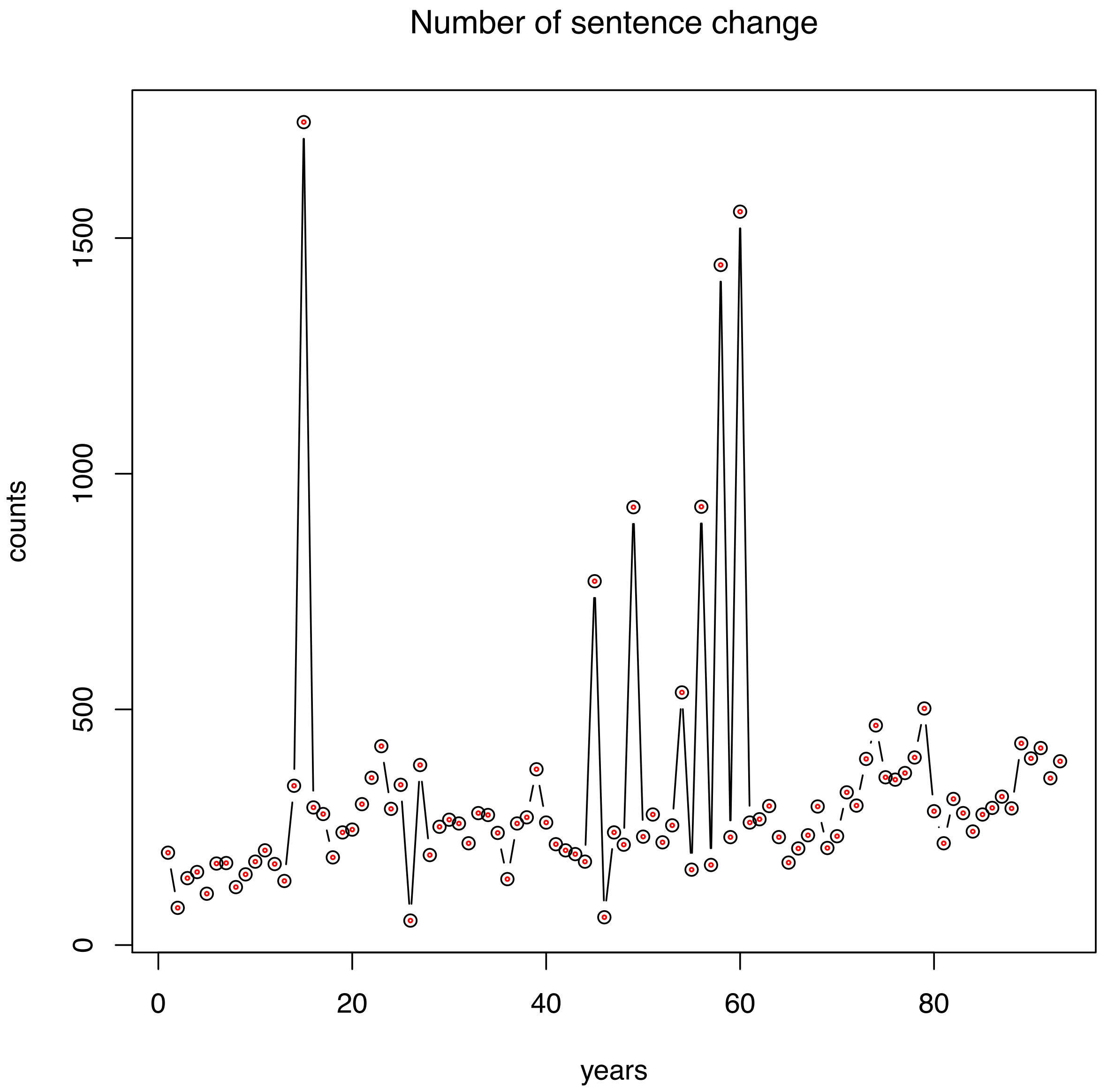
```
## make a list of every information of the speech
summary<-lapply(url,speechinfo)

## now we want to line up the years by orders so that this becomes x-axis
## first we get the vector of titles

titles<-sapply(1:length(url),function(x){summary[[x]]$Title})

## then extract the years from the titles and order them

years<-str_extract(titles,"[[:digit:]]{4}")
yearsindex<-order(years)
range<-(144:236)
len<-length(range)

## then summarize the speeches again by order of years

speechsummary<-lapply(yearsindex,function(x){summary[[x]]})
speechsummary<-lapply(range,function(x){speechsummary[[x]]})
```

Part(h) plotting

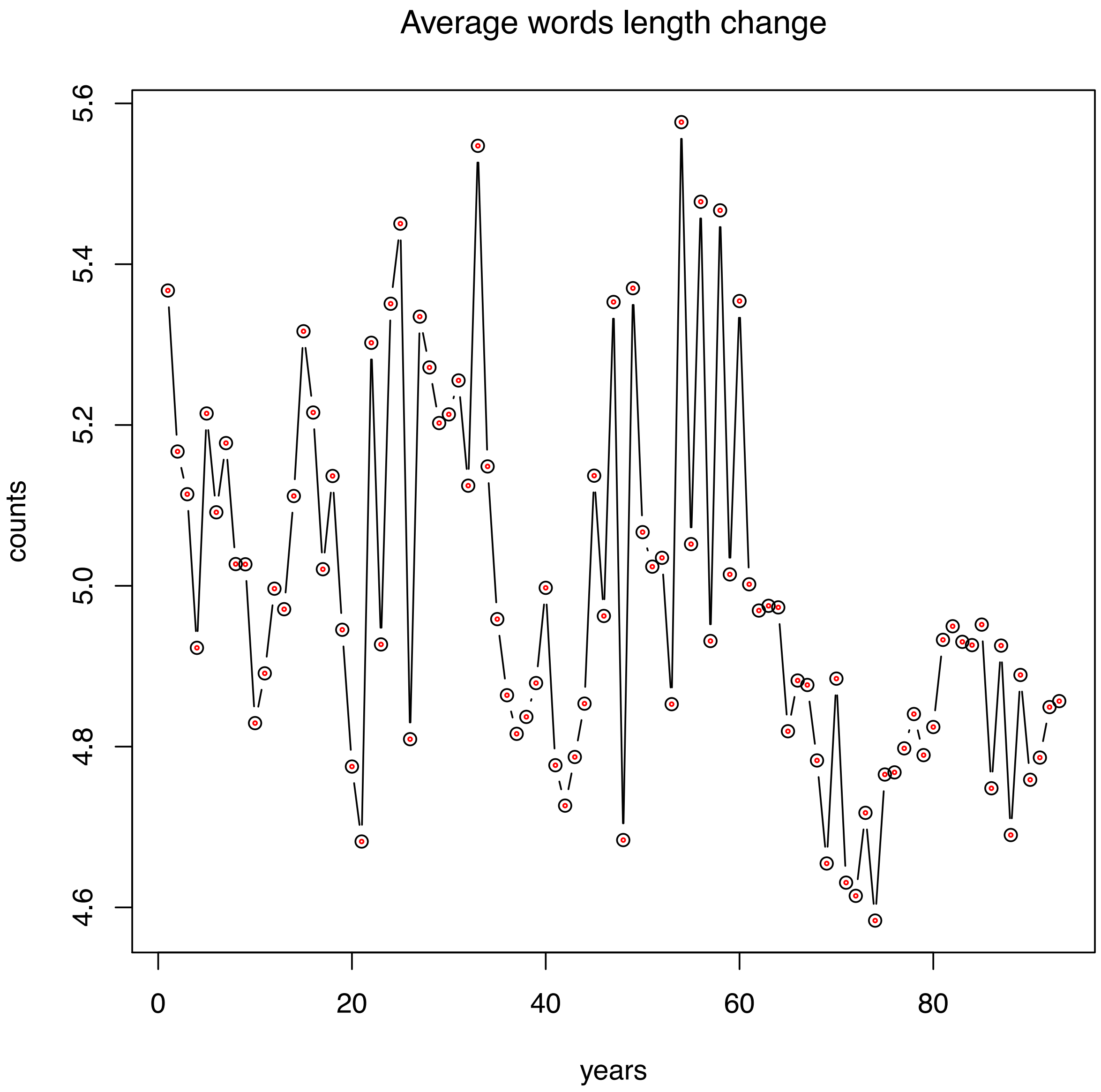
```
## Number of words
numberofwords<-sapply(1:len,function(x){speechsummary[[x]]$CW})
plot(numberofwords,type="b",main="Number of words change",font.main=1,xlab="years",ylab="counts")
points(numberofwords, cex = .3, col = "red")
```



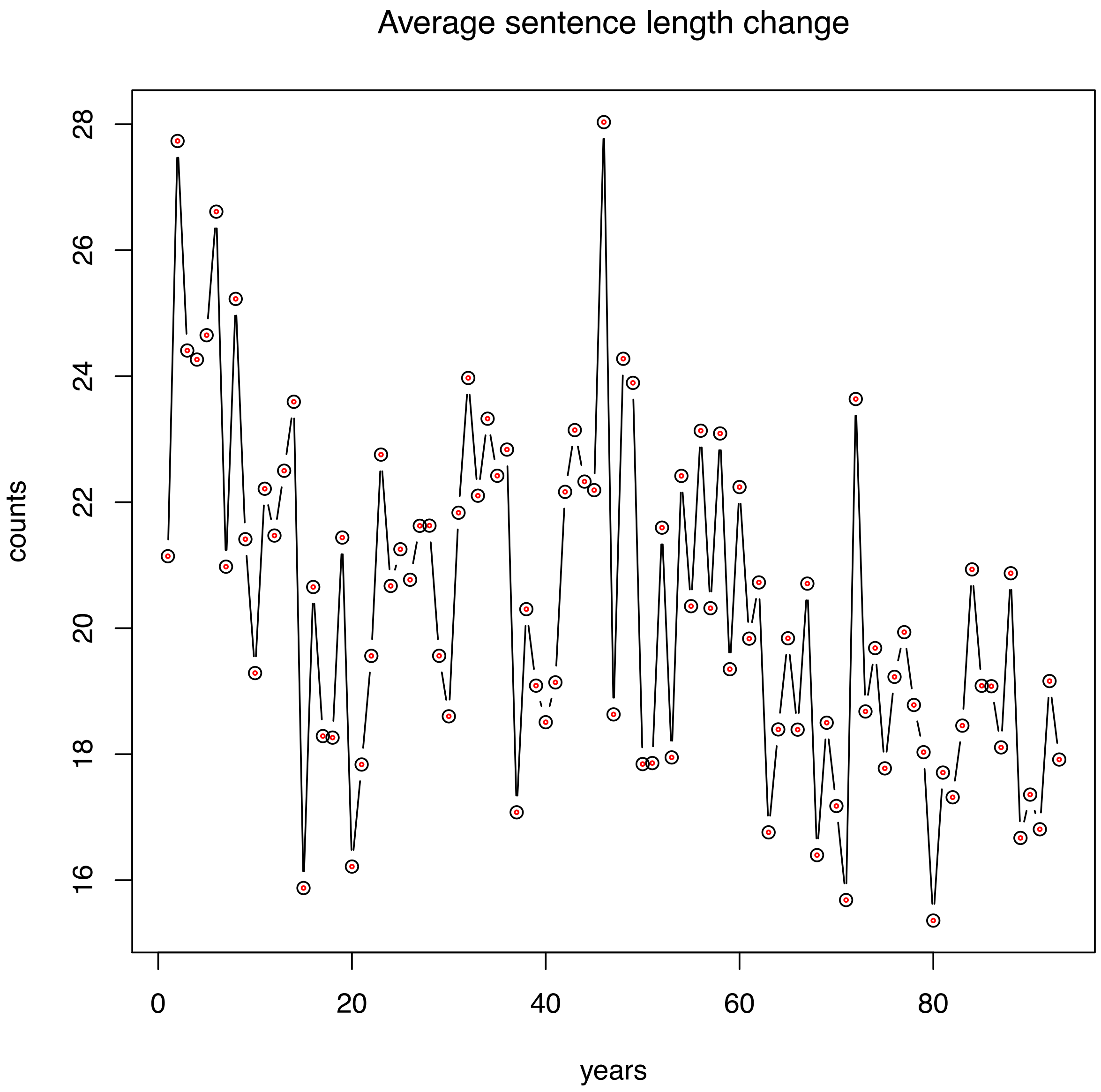
```
## Number of sentences
numberofsentences<-sapply(1:len,function(x){speechsummary[[x]]$CS})
plot(numberofsentences,type="b",main="Number of sentence change",font.main=1,xlab="years",ylab="counts")
points(numberofsentences, cex = .3, col = "red")
```



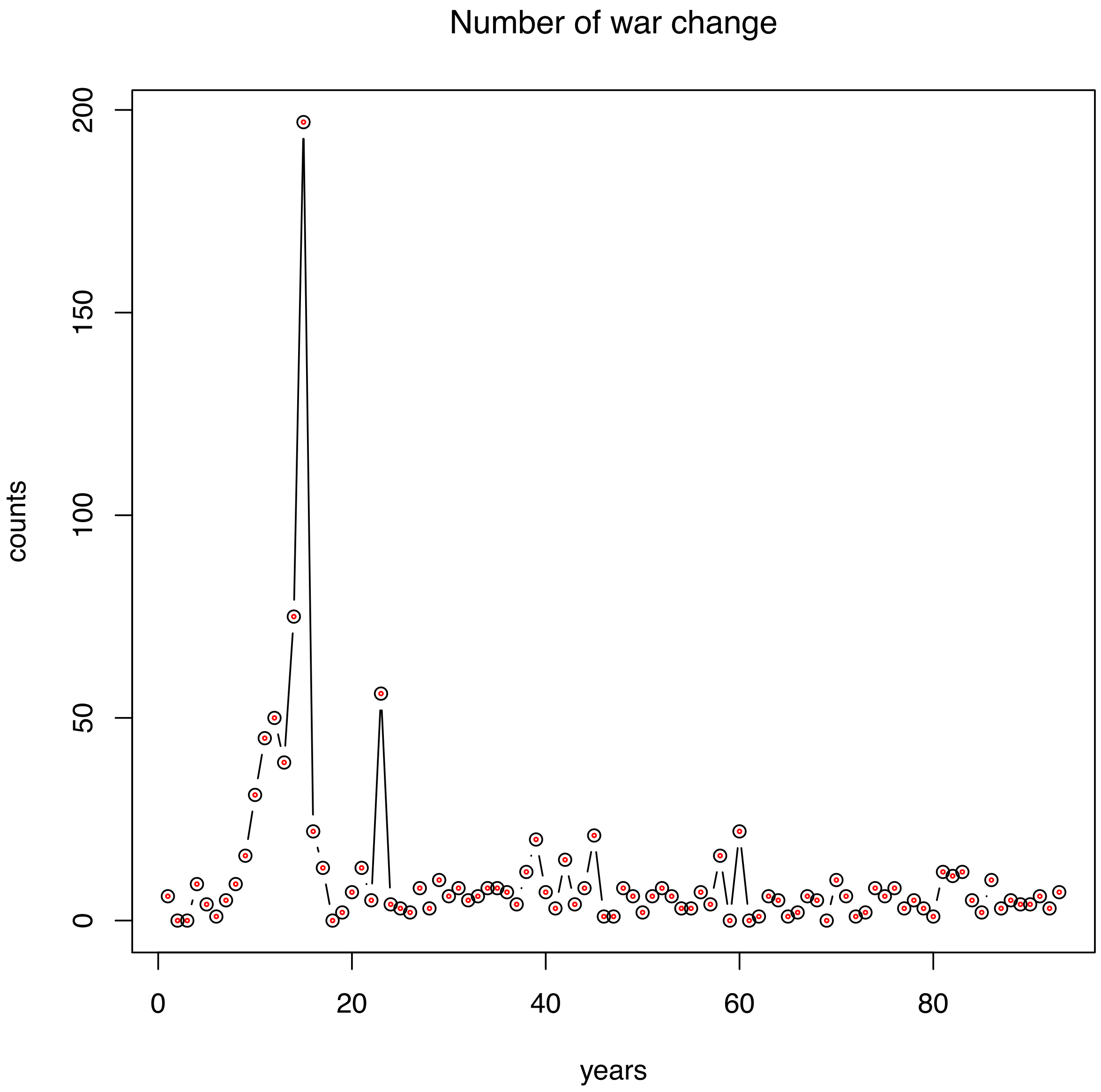
```
## Average words length
averagewords<-sapply(1:len,function(x){speechsummary[[x]]$Averagew})
plot(averagewords,type="b",main="Average words length change",font.main=1,xlab="years",ylab="counts")
points(averagewords, cex = .3, col = "red")
```



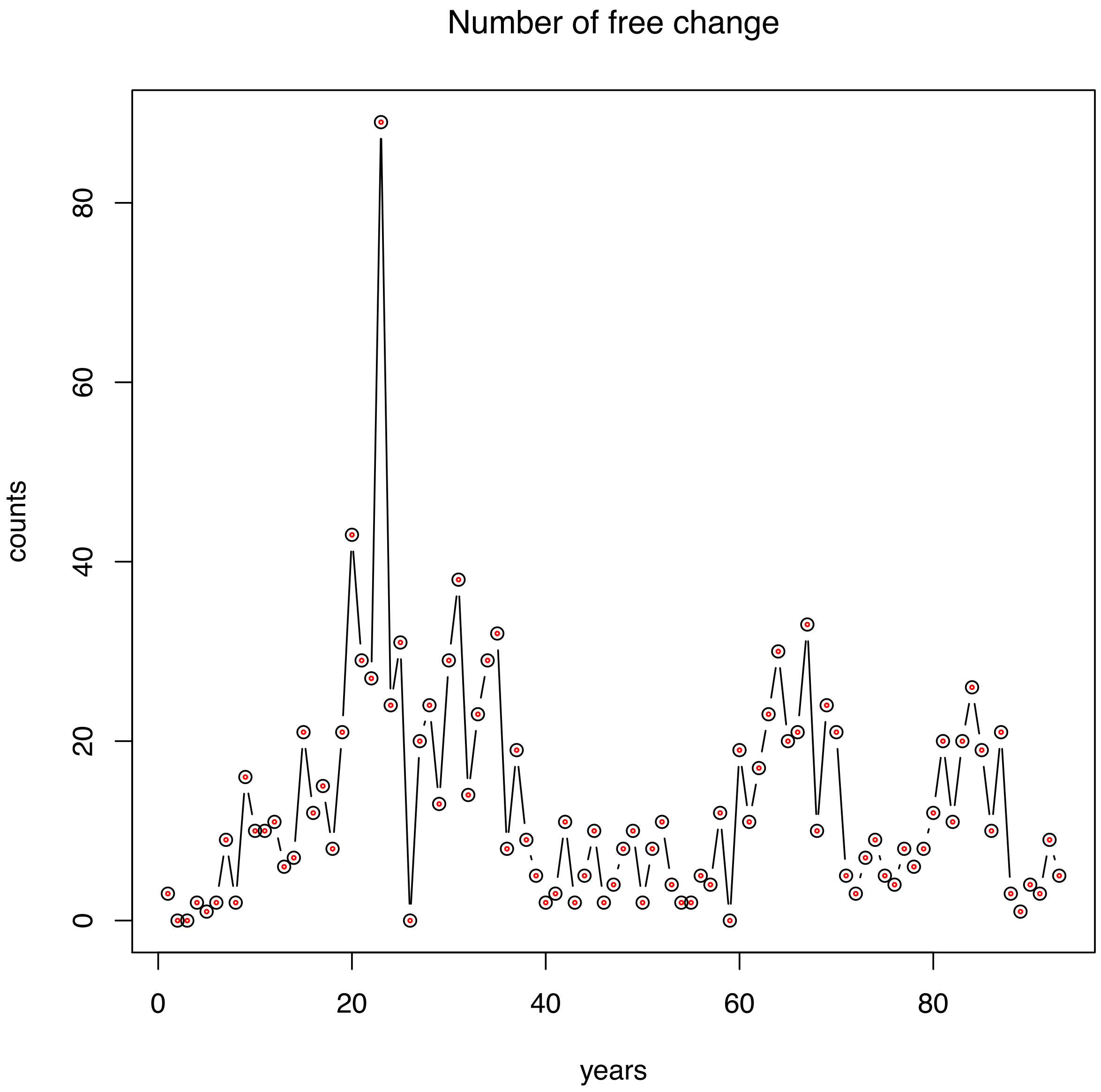
```
## Average sentence length
averagesentence<-sapply(1:len,function(x){speechsummary[[x]]$Averages})
plot(averagesentence,type="b",main="Average sentence length change",font.main=1,
xlab="years",ylab="counts")
points(averagesentence, cex = .3, col = "red")
```



```
## Number of "war"
numberofwar<-sapply(1:len,function(x){speechsummary[[x]]$Wa})
plot(numberofwar,type="b",main="Number of war change",font.main=1,xlab="years",ylab="counts")
points(numberofwar, cex = .3, col = "red")
```



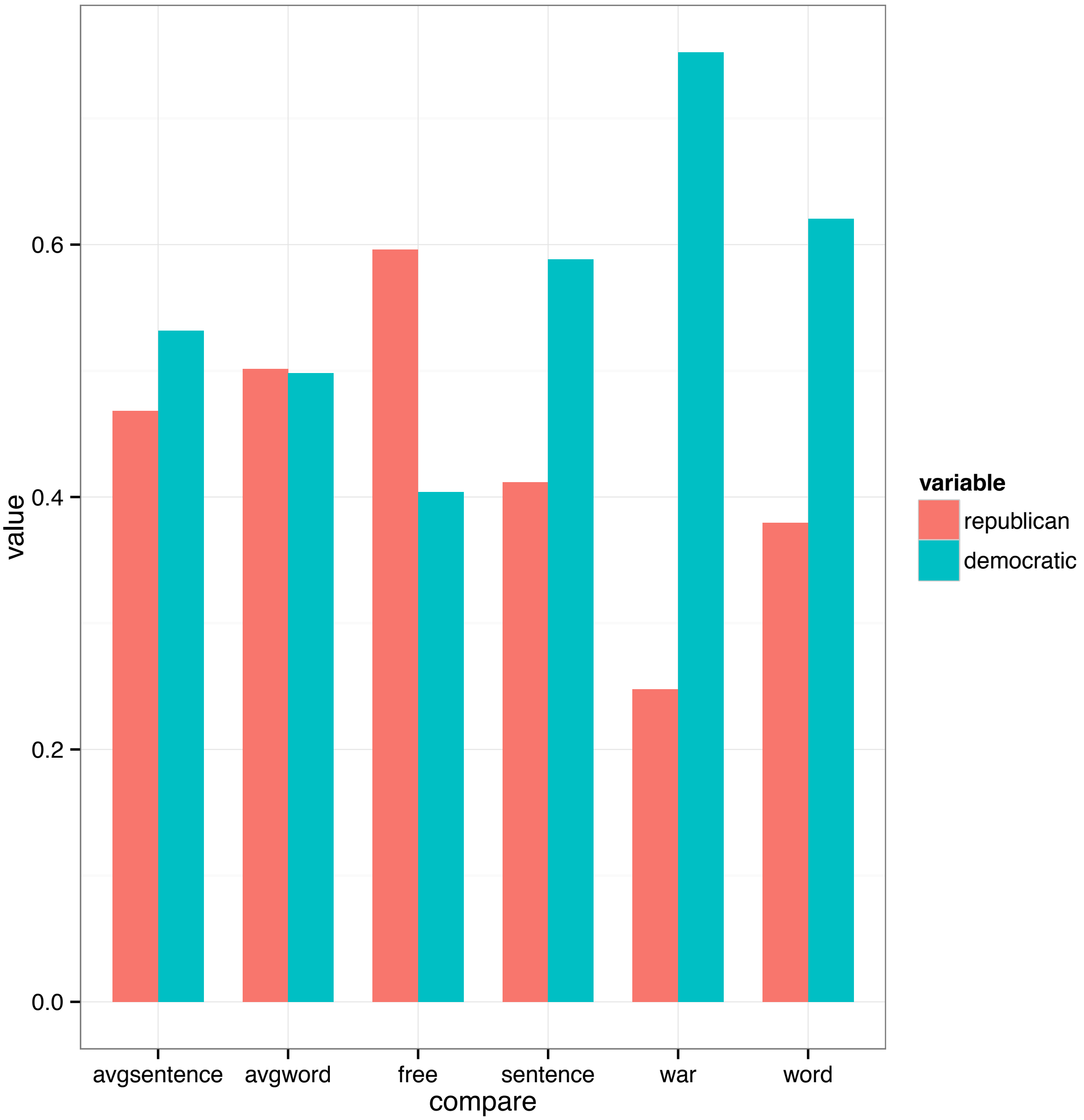
```
## Number of "free"
numberoffree<-sapply(1:len,function(x){speechsummary[[x]]$Fr})
plot(numberoffree,type="b",main="Number of free change",font.main=1,xlab="years",ylab="counts")
points(numberoffree, cex = .3, col = "red")
```



Part (h) Dem vs. Rep

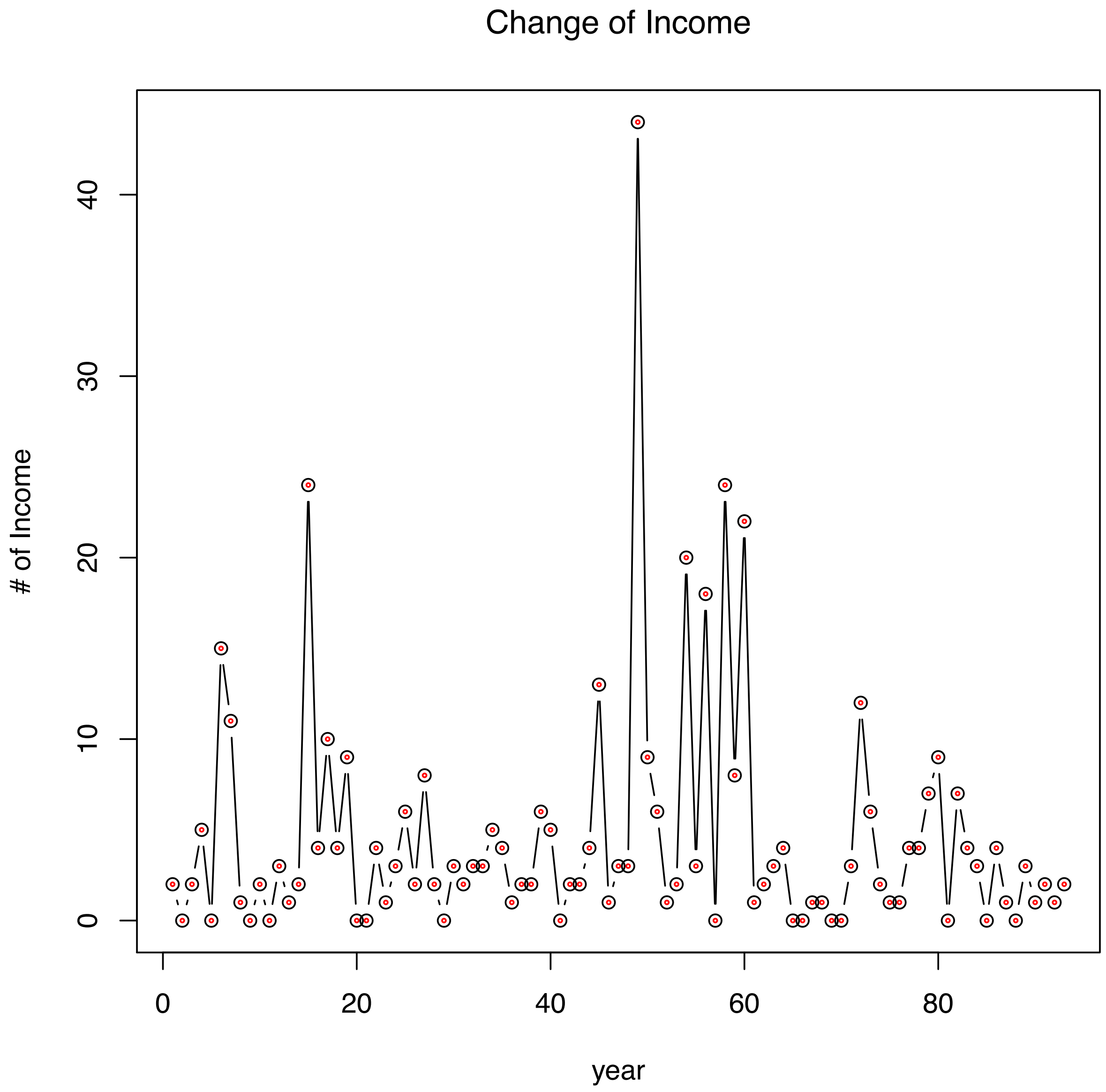
```
## We are comparing the average figures of variables of democratics and republican.

## First, we make a list of democratic speeches.

democratics<-grep("Roosevelt|Truman|Kennedy|Johnson|Carter|Clinton|Obama",titles)
lengthdemocratics<-length(democratics)
democraticspeech<-lapply(democratics,function(x){summary[[x]]})

## Then we calculate the mean of # of variables.
```

```
word1<-mean(sapply(1:lengthdemocratics,function(x){democraticspeech[[x]]$CW}))
sentence1<-mean(sapply(1:lengthdemocratics,function(x){democraticspeech[[x]]$CS}))
avgword1<-mean(sapply(1:lengthdemocratics,function(x){democraticspeech[[x]]$Averagew}))
avgsentence1<-mean(sapply(1:lengthdemocratics,function(x){democraticspeech[[x]]$Averages}))
war1<-mean(sapply(1:lengthdemocratics,function(x){democraticspeech[[x]]$Wa}))
free1<-mean(sapply(1:lengthdemocratics,function(x){democraticspeech[[x]]$Fr}))

## To make it easy to see, we make a vector consisted of the means we calculated above.

dem<-c(word1, sentence1, avgword1, avgsentence1, war1, free1)
dem

## [1] 9045.000  395.127    4.977   22.645   17.270   10.841

## We do the same thing for republican.

republican<-grep("Eisenhower|Nixon|Ford|Reagan|George W. Bush|George Bush",titles)
lengthrepublican<-length(republican)
republicanspeech<-lapply(republican,function(x){summary[[x]]})

word2<-mean(sapply(1:lengthrepublican,function(x){republicanspeech[[x]]$CW}))
sentence2<-mean(sapply(1:lengthrepublican,function(x){republicanspeech[[x]]$CS}))
avgword2<-mean(sapply(1:lengthrepublican,function(x){republicanspeech[[x]]$Averagew}))
avgsentence2<-mean(sapply(1:lengthrepublican,function(x){republicanspeech[[x]]$Averages}))
war2<-mean(sapply(1:lengthrepublican,function(x){republicanspeech[[x]]$Wa}))
free2<-mean(sapply(1:lengthrepublican,function(x){republicanspeech[[x]]$Fr}))

rep<-c(word2, sentence2, avgword2, avgsentence2, war2, free2)
rep

## [1] 5533.512  276.439    5.010   19.936    5.683   16.000

## When comparing the values, we use the ratio since the real value is too huge.

republican<-rep/(rep+dem)
republican

## [1] 0.3796 0.4116 0.5017 0.4682 0.2476 0.5961

democratic<-dem/(rep+dem)
democratic

## [1] 0.6204 0.5884 0.4983 0.5318 0.7524 0.4039

## Plot a graph: The two parties differed greatly in usage of "war".
## Democrats use the word more often.
## Also, it seems that democrats make longer speech with more sentences and words.

library(ggplot2)
library(reshape2)

compare<-c("word","sentence","avgword","avgsentence","war","free")
dataframe<-data.frame(compare,republican,democratic)
melt.var<-melt(dataframe,id=c("compare"))
ggplot(melt.var) +
        geom_bar(aes(x=compare, y=value, fill=variable),
        stat="identity",position="dodge",width=0.7) +
        theme_bw()
```

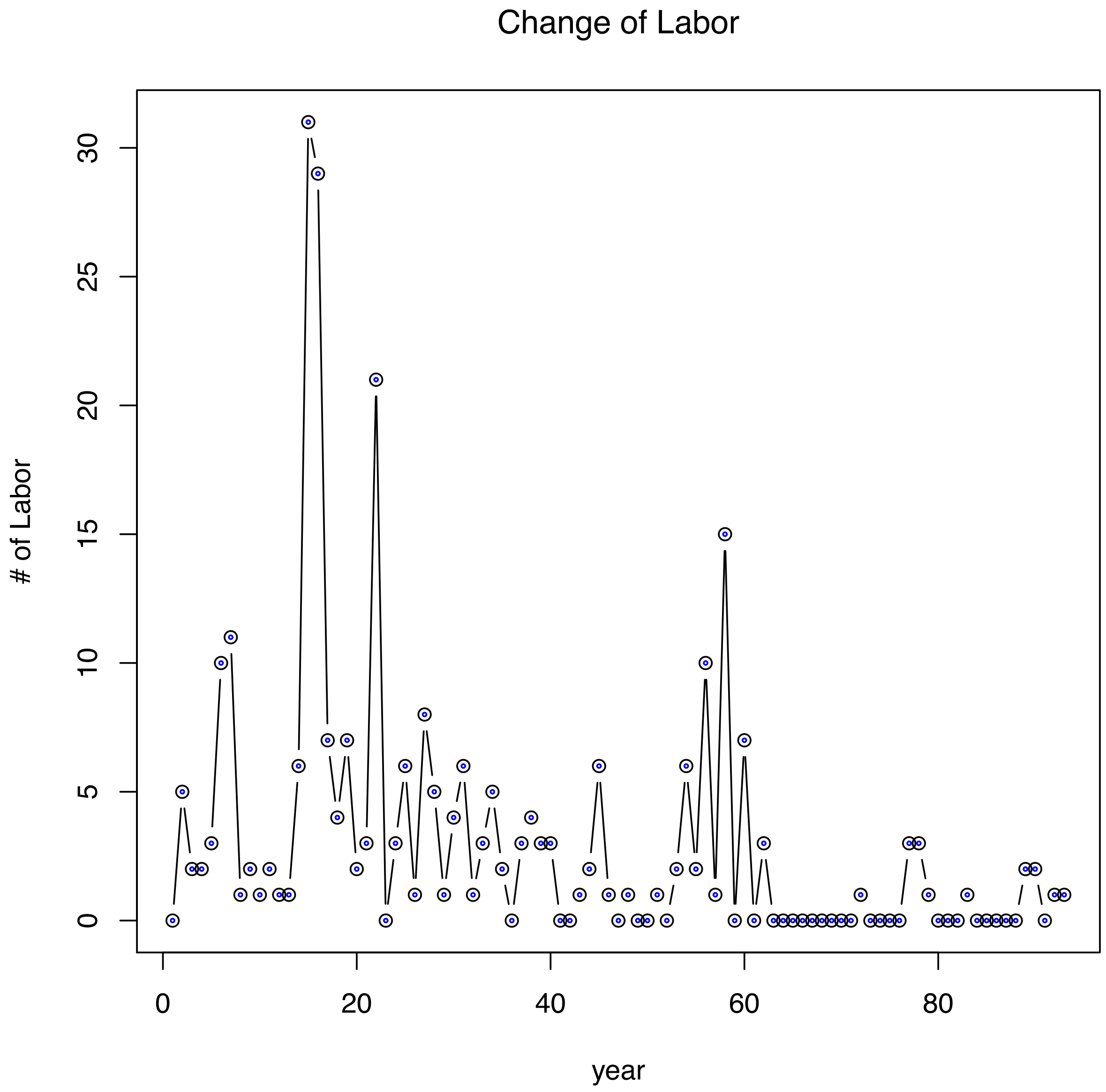


Part(i)

```
## I observed how the number of "income" and "labor" has changed over time.
## The counting function is included in the speechinfo function.

## Number of "income": the word "income" was used especially a lot in 1970s.

incomechange<-sapply(1:len,function(x){speechsummary[[x]]$IC})
plot(incomechange,type="b",main="Change of Income",font.main=1,xlab="year",ylab="# of Income")
points(incomechange, cex = .3, col = "red")
```



```
## Number of "labor": the word "labor" was used especially a lot in 1950s which is the post-war period

laborchange<-sapply(1:len,function(x){speechsummary[[x]]$LB})
plot(laborchange,type="b",main="Change of Labor",font.main=1,xlab="year",ylab="# of Labor")
points(laborchange, cex = .3, col = "blue")
```


Change of Labor
# of Labor
year
0
5
10
15
20
25
30
0
20
40
60
80