Problem 3

The American Presidency Project at UCSB has the text from all of the State of the Union speeches by US presidents, in which the president speaks to Congress to report on the situation in the country. We will use web scraping, text formatting and pattern matching to grab the data; and then do some statistical analysis on them.

(a)

From the website, I download the *index.html* file and use pattern matching to pull out the individual URLs for each speech in order to download individual HTML files. Files are converted to UNIX line-ending using *fromdos*.

```
#### Download all html files
system("wget -q -O 'index_pres.html'
        'http://www.presidency.ucsb.edu/sou.php#axzz265cEKp1a'")
system("fromdos index_pres.html")
indexPres <- readLines('index_pres.html',warn= FALSE)
# Get speech text source
patUrl1 <-
'\\s{16}<td width=\\"\\d{2}\\" align=\\"center\\" class=\\"doclist\\"><a href=\\"'
indexPres <- indexPres[grep(patUrl1,indexPres,perl= TRUE)]
indexPres <- sapply(indexPres,
                    function(x){gsub(patUrl1,"",x)},USE.NAMES= FALSE)
patUrl2 <- '\\">\\d{4}<\\/a>(\\*|)<\\/td>'
indexPres <- sapply(indexPres,
                    function(x){gsub(patUrl2,"",x)},USE.NAMES= FALSE)
# Get file id from the source url
patUrl3 <-
'http:\\/\\/www\\.presidency\\.ucsb\\.edu\\/ws\\/index\\.php\\?pid='
fileid <- sapply(indexPres,
                 function(x){gsub(patUrl3,"",x)},USE.NAMES= FALSE)
# Download all files and convert to unix
sapply(1:length(fileid),
       function(i){
       system(paste("wget -q -O '",fileid[i],".html' '",indexPres[i],"'",sep=""));
       system(paste("fromdos ",fileid[i],".html",sep=""))})
```

(b)

For each speech, I use pattern matching to extract the body of the speech while retaining the name of the president and the year of the speech. The function is applied with vector operations using *sapply()*.

For the *speechVec*, text pre-processing is done by

1. Replacing HTML line-end with UNIX ones
2. Removing all the HTML format operators
3. Modifying all the HTML special characters to similar UTF-8 ones

```
# Import all *.html lines
ff <- sapply(fileid, function(x) {
    readLines(paste(x, ".html", sep = ""), warn = FALSE)
})
# Get the president name
patName <- "^<title>(.*?)<\\/title>"
namePres <- ff[grep(patName, ff, perl = TRUE)]
namePres <- sapply(namePres, function(x) {
    gsub(patName, "\\1", x)
}, USE.NAMES = FALSE)
namePres <- sapply(namePres, function(x) {
    return(unlist(strsplit(x, ":"))[1])
}, USE.NAMES = FALSE)
# Get the talk date
patDate <- "^.*<span class=\\\"docdate\\\">(.*?)<\\/span>.*$"
dateTalk <- ff[grep(patDate, ff, perl = TRUE)]
dateTalk <- sapply(dateTalk, function(x) {
    gsub(patDate, "\\1", x)
}, USE.NAMES = FALSE)
dateTalk <- sapply(dateTalk, function(x) {
    gsub("^.*\\s", "", x)
}, USE.NAMES = FALSE)
# Get the speech content text and prune for nice-format print
patText <- "^.*<span class=\\\"displaytext\\\">(.*?)<\\/span>.*$"
speechVec <- ff[grep(patText, ff, perl = TRUE)]
speechVec <- sapply(speechVec, function(x) {
    gsub(patText, "\\1<p>", x)
}, USE.NAMES = FALSE)  # grab speech text
speechVec <- sapply(speechVec, function(x) {
    gsub("(<p.*?>|<\\/p>|<br>)", "\n", x)
}, USE.NAMES = FALSE)  # for line ending
speechVec <- sapply(speechVec, function(x) {
    gsub("<.*?>", "", x)
}, USE.NAMES = FALSE)  # remove all html format
speechVec <- sapply(speechVec, function(x) {
    x <- gsub("&mdash;", " -- ", x)
    x <- gsub("&nbsp;", "  ", x)
    x <- gsub("&lsquo;", " ' ", x)
    x <- gsub("&#8226;", " \\. ", x)
    x <- gsub("&lt;", " < ", x)
    x <- gsub("&deg;", " degree ", x)
    x <- gsub("&pound;", " pound ", x)
    x <- gsub("&fra.*?;", " 1/2 ", x)
    x <- gsub("&O.*?;", "O", x)
    x <- gsub("&e.*?;", "e", x)
}, USE.NAMES = FALSE)  # html special char
```

(c)

Each speech is stored as a single character vector with all non-text stripped out. The encoding is converted from WINDOWS-1251 to UTF-8. Meanwhile, the information about the tags of "Laughter" and "Applause" and the number of times it was used are kept as a record for each speech. The *speechVec[i]* will be printed out in a nicely-formatted manner.

```r
# Remove audience response tags (laughter & applause)
patLau <- "\\[.*?(Laughter|laughter).*?\\]"
patApp <- "\\[.*?(Applause|applause).*?\\]"
getlauNum <- function(x) {
    if (length(gregexpr(patLau, x, perl = TRUE)[[1]]) == 1 && gregexpr(patLau,
        x, perl = TRUE)[[1]] == -1) {
        return(0)
    } else {
        return(length(gregexpr(patLau, x, perl = TRUE)[[1]]))
    }
}  # get the number of [laughter]
getappNum <- function(x) {
    if (length(gregexpr(patApp, x, perl = TRUE)[[1]]) == 1 && gregexpr(patApp,
        x, perl = TRUE)[[1]] == -1) {
        return(0)
    } else {
        return(length(gregexpr(patApp, x, perl = TRUE)[[1]]))
    }
}  # get the number of [applause]
lauNum <- sapply(speechVec, getlauNum, USE.NAMES = FALSE)
appNum <- sapply(speechVec, getappNum, USE.NAMES = FALSE)
speechVec <- sapply(speechVec, function(x) {
    iconv(x, from = "WINDOWS-1251", to = "UTF-8", sub = " ")
})
speechVec <- sapply(speechVec, function(x) {
    x <- gsub(patLau, "", x, perl = TRUE)
    x <- gsub(patApp, "", x, perl = TRUE)
})
# remove all the non-verbal tags from the speech text
names(speechVec) <- NULL  # clean the speechVec name (otherwise, long and messy)
```

(d)

The collection of speeches is stored in a clean fashion of list elements. The ith entries of all the elements in *listSpeech* give information about the ith speech. This is easy later for plotting variables changes over time.

```r
listSpeech <- list()
listSpeech$id <- fileid
listSpeech$name <- namePres
listSpeech$date <- as.integer(dateTalk)  # conversion for plotting, sorting
listSpeech$numLaughter <- lauNum
listSpeech$numApplause <- appNum
listSpeech$speech <- speechVec
```

(e) (f)

Words and sentences are extracted from each speech, and are stored as individual elements of a (rather long) character vector. Counts are also done on both words and sentences.

```r
# Speech analysis Grab words from the speech by replacing all non-word
# char with space and split
getWords <- function(x) {
    x <- gsub("'", "", x, perl = TRUE)
    x <- gsub("\\W+", " ", x, perl = TRUE)
    xs <- unlist(strsplit(x, "[ ]+", perl = TRUE))
    return(xs[xs != ""])
}
# Grab sentences from the speech by spliting with ending chars like [.!?]
getSents <- function(x) {
    x <- gsub(" (Mr|Ms|Mrs|Dr|St|Sr|Jr)\\.", "\\1", x, perl = TRUE)
    x <- gsub("[\\.!\\?][ \t]+", "\n", x, perl = TRUE)
    xs <- unlist(strsplit(x, "\n", perl = TRUE))
    return(xs[xs != ""])
}
listSpeech$wc <- sapply(speechVec, function(x) {
    return(length(getWords(x)))
}, USE.NAMES = FALSE)  # word count
listSpeech$sc <- sapply(speechVec, function(x) {
    return(length(getSents(x)))
}, USE.NAMES = FALSE)  # sentence count
listSpeech$wMean <- sapply(speechVec, function(x) {
    return(mean(nchar(getWords(x))))
}, USE.NAMES = FALSE)  #avg word length
listSpeech$wSD <- sapply(speechVec, function(x) {
    return(sd(nchar(getWords(x))))
}, USE.NAMES = FALSE)  # word length sd
listSpeech$sMean <- sapply(speechVec, function(x) {
    return(mean(nchar(getSents(x))))
}, USE.NAMES = FALSE)  # avg sentence length
listSpeech$sSD <- sapply(speechVec, function(x) {
    return(sd(nchar(getSents(x))))
}, USE.NAMES = FALSE)  # sentence length sd
```

(g) (h)

We now start to extract some features of interest from the speeches to analyze how the speeches have changed over time. The result of all this is a list with each element containing the information about a speech: the speech as a single string, the vector of sentences, the vector of words and the additional quantification of variables about the speech from (g) as well as the non-verbal variables from (c).

1. Length in words and sentences *wc,sc*
2. Average and SD of word and sentence lengths *wMean,wSD,sMean,sSD*
3. Number of quotations in each speech, mean length (in words), and SD of length (in words) of the quotations in each speech *quoNum,quoMean,quoSD*
4. The most common meaningful words, where non-meaningful words are pre-defined *cmw*
5. Counts of the following words or word stems:

   I, we

   America,n

   democracy,tic

republic

Democrat,ic

Republican

free,dom

war

God – not including God bless

God Bless

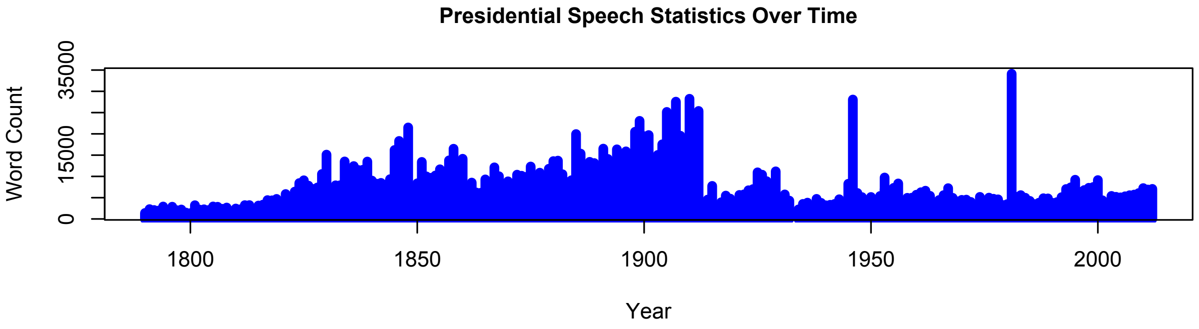
Jesus, Christ, Christian

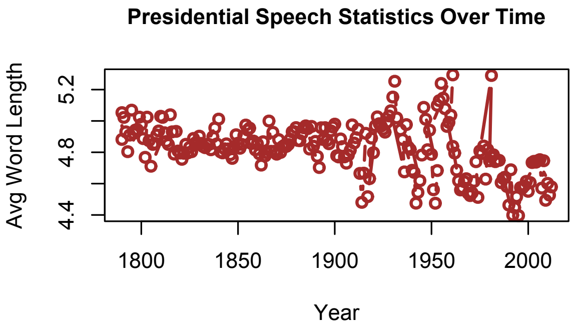
Woman – I think would be interesting

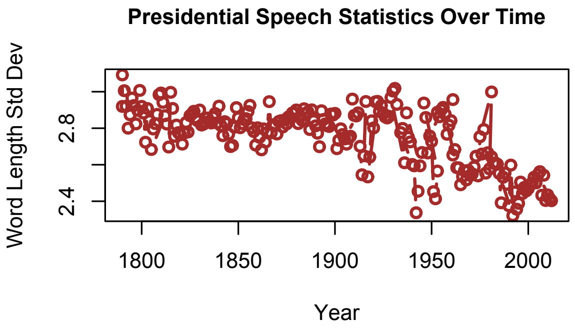
```r
tmpList <- matrix(rep(0, 226 * 15), nrow = 226, ncol = 15)
## Speech list with element-wise analysis
speechList <- list()  #empty list
# Get the non-meaningful words file
system("wget -O 'cmw.txt' 'http://www.textfixer.com/resources/common-english-words.txt'")
commonWords <- readLines("cmw.txt", warn = FALSE)
commonWords <- unlist(strsplit(commonWords, ",", perl = TRUE))
# Element-wise analysis on each speech
for (i in 1:length(fileid)) {
    ss <- list()  #empty list element
    # Global attributes: from results before (reproduce for better storage)
    ss$id <- fileid[i]
    ss$name <- namePres[i]
    ss$date <- as.integer(dateTalk[i])
    ss$numLaughter <- lauNum[i]
    ss$numApplause <- appNum[i]
    ss$speech <- speechVec[i]
    # Indiv attributes: get from within each speech
    talkWords <- getWords(speechVec[i])
    talkSents <- getSents(speechVec[i])
    ss$words <- talkWords  # words vector
    ss$sents <- talkSents  # sentence vector
    ss$wc <- length(talkWords)  # word count
    ss$sc <- length(talkSents)  # sentence count
    ss$wMean <- mean(nchar(talkWords))  # avg word length
    ss$wSD <- sd(nchar(talkWords))  # word length sd
    ss$sMean <- mean(nchar(talkSents))  # avg sentence length
    ss$sSD <- sd(nchar(talkSents))  # sentence length sd; ss[14]
    patQuo <- "\"(.*?)\""  # quotation pattern
    quo <- talkSents[grep(patQuo, talkSents, perl = TRUE)]
    if (length(quo) != 0) {
        # get quotation attributes
        quo <- sapply(quo, function(x) {
            gsub(patQuo, "\\1", x)
        }, USE.NAMES = FALSE)
        ss$quoNum <- length(quo)
        ss$quoMean <- mean(nchar(quo))
        ss$quoSD <- sd(nchar(quo))
    } else {
        ss$quoNum <- 0
        ss$quoMean <- 0
        ss$quoSD <- 0
    }  #ss[17]
    cmw <- sort(table(talkWords), decreasing = TRUE)
    cmw <- cmw[which(!(names(cmw) %in% commonWords))]  # get meaningful words
    ss$cmw <- cmw[cmw >= 10]  #arbitrary cut-off for display
    ss$strIwe <- cmw[grep("^(I|[Ww]e)$", names(cmw))]  #string 'I|We'; ss[19]
    ss$strAme <- cmw[grep("[Aa]merica(|n)", names(cmw))]  #string 'America'
    ss$strDem <- cmw[grep("[Dd]emocra(cy|tic)", names(cmw))]  #string 'democratic'
    ss$strRep <- cmw[grep("[Rr]epublic(|n)", names(cmw))]  #string 'republican'
    ss$strFree <- cmw[grep("^[Ff]ree(|dom)$", names(cmw))]  #string 'free'
    ss$strWar <- cmw[grep("^[Ww]ar(|s)$", names(cmw))]  #string 'war'
    ss$strGod <- cmw[grep("^[Gg]od(|s)$", names(cmw))]  #string 'God'
    ss$strChr <- cmw[grep("(Jesus|Christ|Christian)", names(cmw))]  #string 'Chirst|Jesus'
    ss$strWoman <- cmw[grep("^[Ww]om[ae]n$", names(cmw))]  #Mystring 'Woman'; ss[27]
    ssGodBless <- talkSents[grep("[Gg]od [Bb]less", talkSents, perl = TRUE)]
    if (length(ssGodBless) != 0) {
        #string 'God Bless' from sentences
        ss$strGodBless <- sapply(ssGodBless, function(x) {
            return(length(gregexpr("[Gg]od [Bb]less", x, perl = TRUE)[[1]]))
        }, USE.NAMES = FALSE)
    } else {
        ss$strGodBless <- 0
    }
    # add to speechList and listSpeech
    speechList[[i]] <- ss
    tmpList[i, 1:3] <- unlist(ss[15:17])  #quo attr
    tmpList[i, 4:13] <- sapply(ss[19:28], sum)  #cmw related attr
}
# prepare for plotting, store as listSpeech elements
listSpeech$quoNum <- tmpList[, 1]
listSpeech$quoMean <- tmpList[, 2]
listSpeech$quoSD <- tmpList[, 3]
listSpeech$strIwe <- tmpList[, 4]
listSpeech$strAme <- tmpList[, 5]
listSpeech$strDem <- tmpList[, 6]
listSpeech$strRep <- tmpList[, 7]
listSpeech$strFree <- tmpList[, 8]
listSpeech$strWar <- tmpList[, 9]
listSpeech$strGod <- tmpList[, 10]
listSpeech$strChr <- tmpList[, 11]
listSpeech$strWoman <- tmpList[, 12]
listSpeech$strGodBless <- tmpList[, 13]
```

(i) (j)

Some basic plots that show how the variables have changed over time are given below.

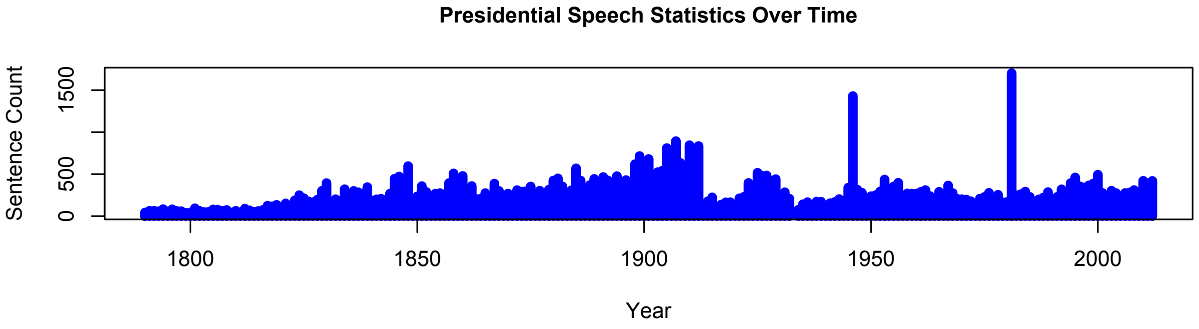


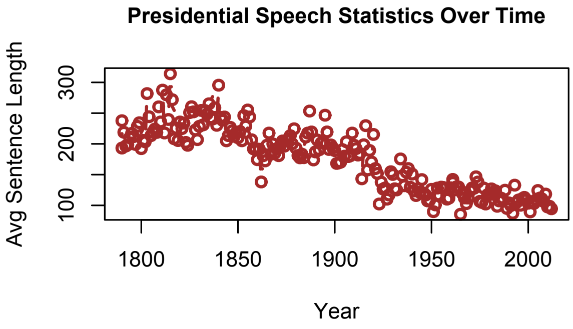


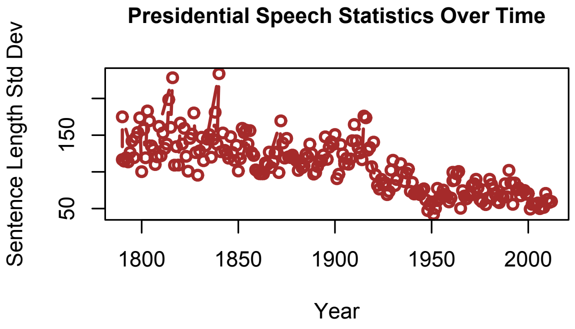


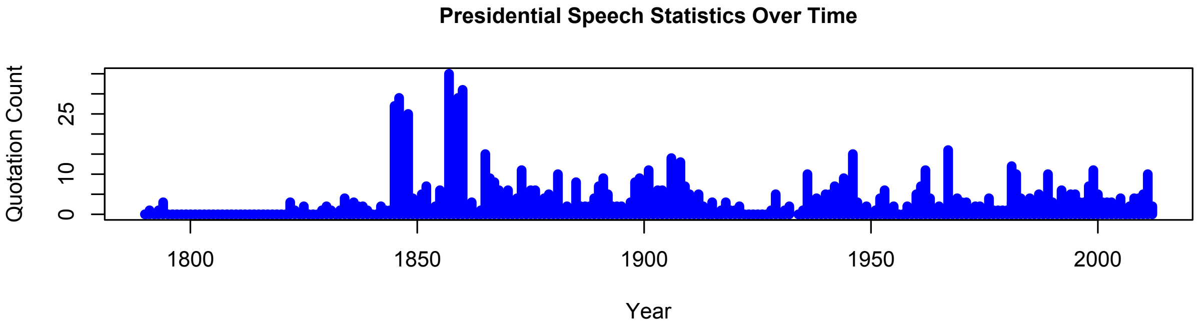
We can see that the word counts of the speeches are higher prior to 1920s than the modern ages. So does the avg word length and word length sd. The presidents nowadays seem to like brevity.

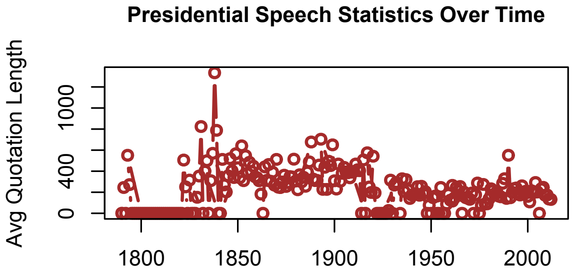
Similar plots for sentence counts and quotation counts are given below. The similar trends stand.

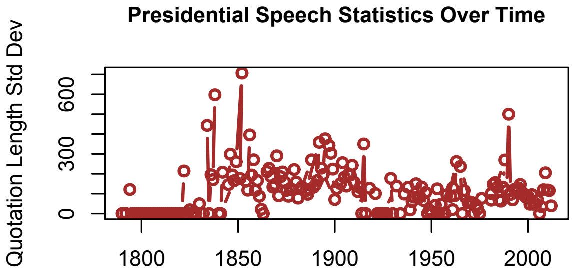




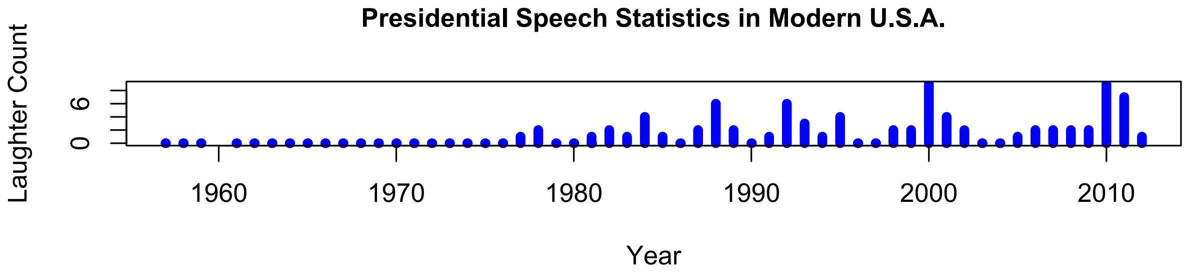


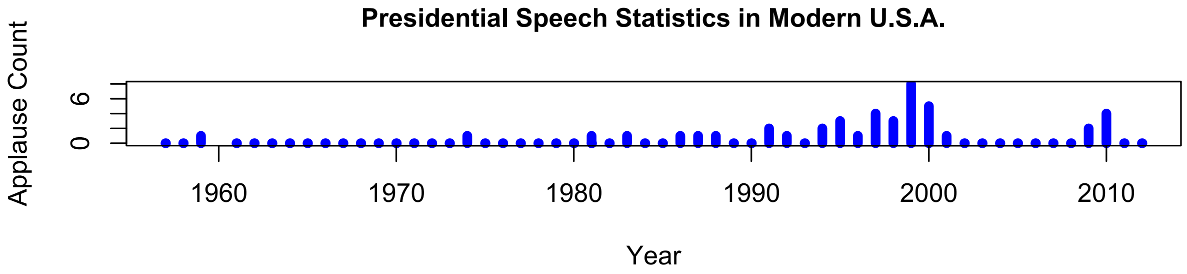

Presidential Speech Statistics Over Time


Presidential Speech Statistics Over Time

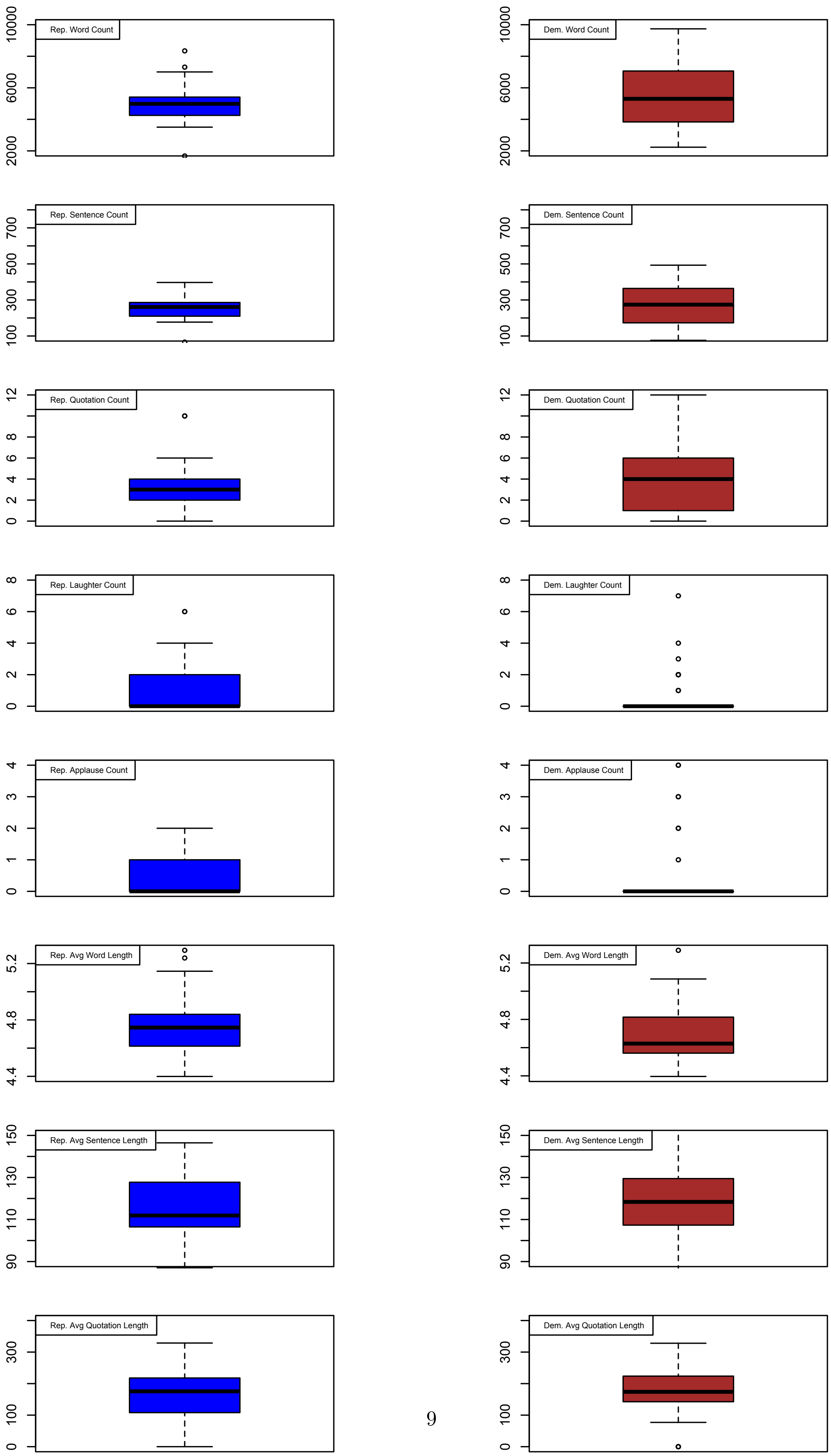

Presidential Speech Statistics Over Time

In recent years, with full records of non-verbal attributes, we can see that [Laughter] and [Applause] counts have some positive correlation. The speeches receive more responses from the audience around 2000 than other years.

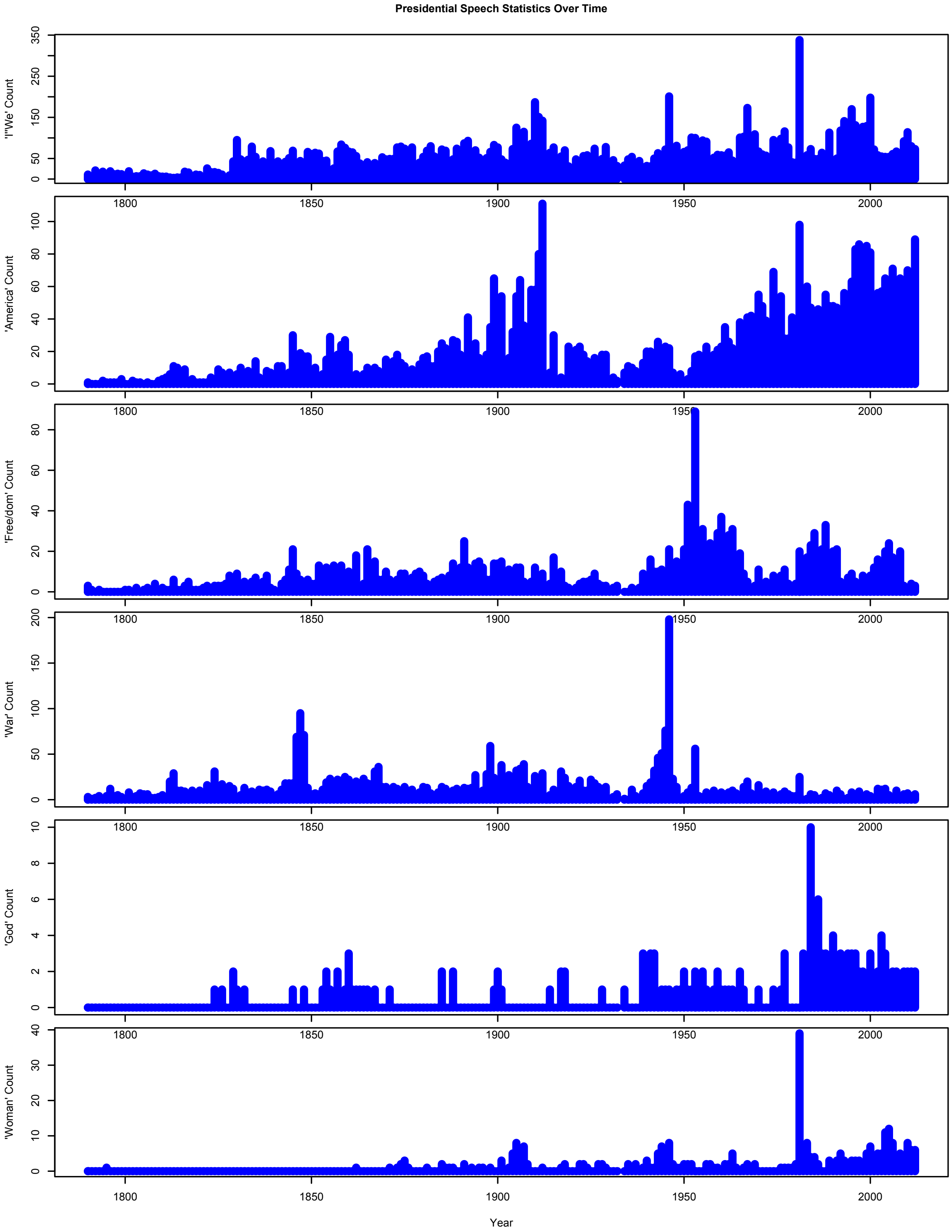

Presidential Speech Statistics in Modern U.S.A.


Presidential Speech Statistics in Modern U.S.A.

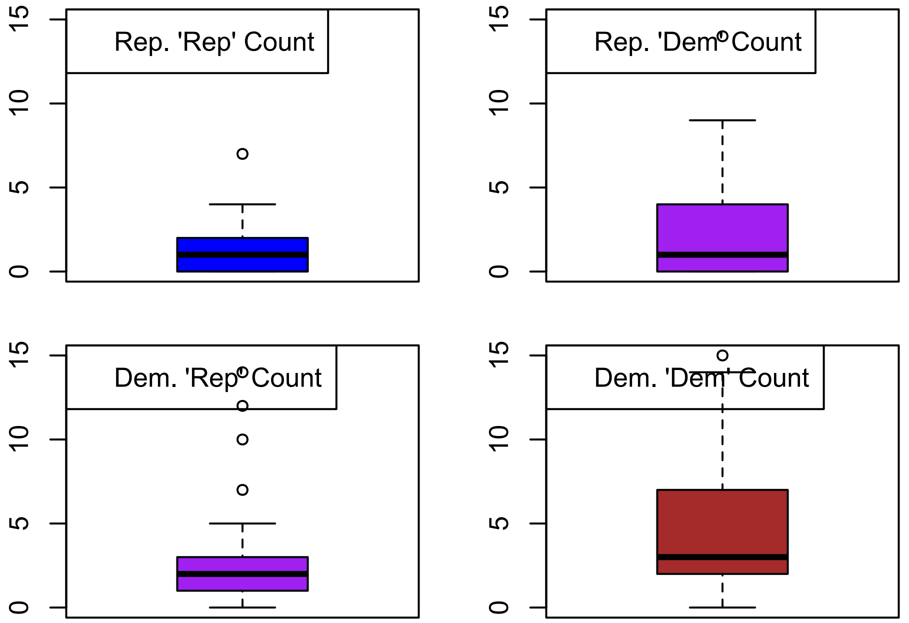
And for presidents since Franklin Roosevelt in 1932, comparison between Republican presidents (Eisenhower, Nixon, Ford, Reagan, G. Bush, G.W. Bush) and Democratic presidents (Roosevelt, Truman, Kennedy, Johnson, Carter, Clinton, Obama) are also given below in the box-plot comparison.


Rep. Word Count
10000
6000
2000
Dem. Word Count
10000
6000
2000
Rep. Sentence Count
700
500
300
100
Dem. Sentence Count
700
500
300
100
Rep. Quotation Count
12
8
6
4
2
0
Dem. Quotation Count
12
8
6
4
2
0
Rep. Laughter Count
8
6
4
2
0
Dem. Laughter Count
8
6
4
2
0
Rep. Applause Count
4
3
2
1
0
Dem. Applause Count
4
3
2
1
0
Rep. Avg Word Length
5.2
4.8
4.4
Dem. Avg Word Length
5.2
4.8
4.4
Rep. Avg Sentence Length
150
130
110
90
Dem. Avg Sentence Length
150
130
110
90
Rep. Avg Quotation Length
300
100
0
Dem. Avg Quotation Length
300
100
0

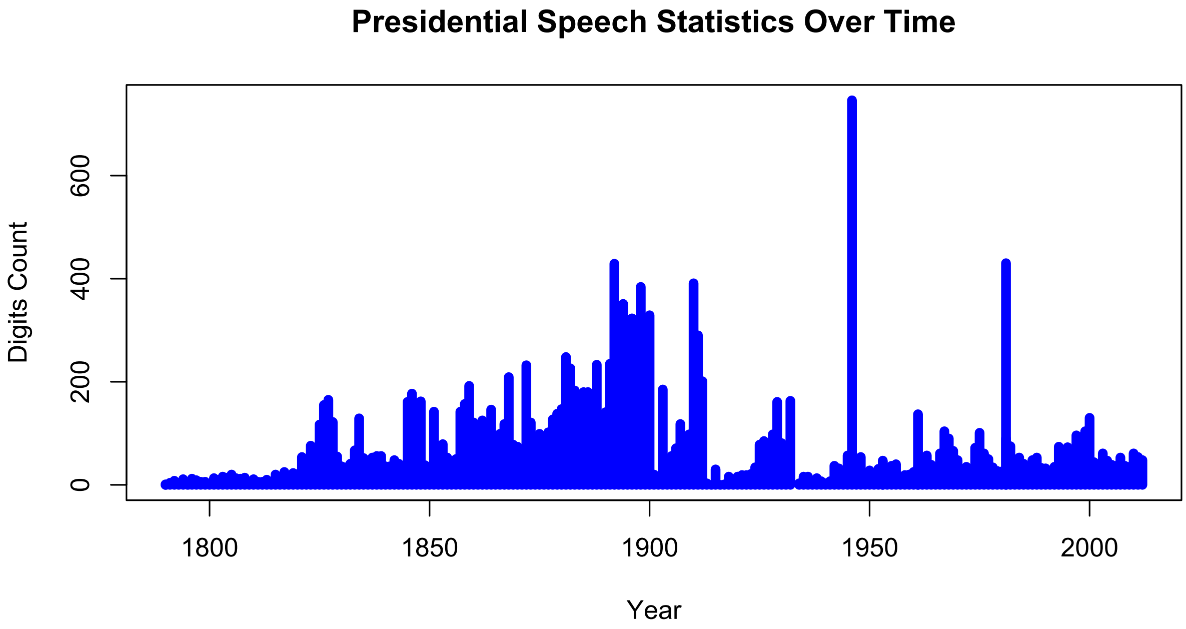
Some additional research with plotting that illustrates how the speeches have changed over time are shown below. The different keywords have evolved over time, with peak in WAR during wartime and GOD-rich in recent years.



The comparison between Reps and Dems could be more interesting with the strings REPUBLICAN and DEMOCRATIC. It seems that the Dems are more focused on their persuit while Reps are impartial to both words.



Some additional variables that quantify speech in interesting ways are produced. The number of digits shown within the speech is a good indicator whether the president speaks more quantatively, or vice versa, more qualitatively.


Presidential Speech Statistics Over Time
Digits Count
Year