Analysis of Presidential Inagural Adresses

Tomas Meade

All the data relating to the speeches was webscrapped from the www.presidency.ucsb.edu website.

```r
#function to webscrape website and return the links to all speeches
getLinks <- function(){
  #URL for all speeches
  URL <- "https://www.presidency.ucsb.edu/documents/presidential-documents-archive-guidebook/annual-mess

  html <- read_html(URL)

  links <- html_nodes(html, css = "td~ td+ td a") %>% html_attr("href")

  #remove special case with Nixon
  links <- links[-(53)]

  return(links)
}

#Generate HTML of all speeches
#I created this variable as a global variable that can always be accessed
#because it is computationally exhaustive to read all the htmls
speechesHtml <- lapply(getLinks(), read_html)


#function that gets the names of each President associated with each speech
#using the global speechesHtml variable
getSpeechesPresName <- function(){
  speechesPresName <- lapply(speechesHtml,
                             function(x)
                               html_text(html_nodes(x,
                                                    css = ".diet-title a")))
  return(speechesPresName)
}


#function that gets the year of each speech
getSpeechesYear <- function(){
  speechesDate <- lapply(speechesHtml,
                         function(x)
                           html_text(html_nodes(x,
                                                css = ".date-display-single")))

  speechesYear <- gsub(".*, *", "", speechesDate)
  return(speechesYear)
}
```

```r
#function that gets the body of each speech
getSpeechesBody <- function(){
  speechesBody <- lapply(speechesHtml,
                         function(x)
                           html_text(html_nodes(x,
                                                css = ".field-docs-content p")))
  return(speechesBody)
}


#function that gets the extra things in the body of the speech
#not said by the president
#for this I just focused on things inside square brackets like [Laughter],
#[Applause], and [As delivered in person before a joint session]
getExtras <- function(){

  #get the body of all the speeches and initialize a vector to store the extras
  speeches <- getSpeechesBody()
  extra <- c()
  extra[[1]] <- unlist(str_extract_all(speeches[[1]], "\\[.*?\\]"))

  #loop through the speeches and store anything inside brackets
  #could have used lapply but I think a loop improves readability here
  #I use lapply for basically everything else
  for (i in 2:length(speeches)) {
    extra[[i]] <- unlist(str_extract_all(speeches[[i]], "\\[.*?\\]"))
  }
  return(extra)
}

#show stored extra information
head(getExtras()[[1]])
```

```
## [1] "[Laughter]" "[Applause]" "[laughter]" "[under]"    "[child]"
## [6] "[Laughter]"
```

```r
#function to actually remove all the extras
getSpeechesClean <- function(){
  speechesClean <- lapply(getSpeechesBody(), str_replace_all, "\\[.*?\\]", "")
  return(speechesClean)
}

#count the amount of laughter in each speech
laughter <- lapply(getExtras(), str_count, "Laughter")
laughter_count <- lapply(laughter, sum)

#show times laughter occured in speech
laughter_count[1:3]
```

```
## [[1]]
## [1] 3
##
## [[2]]
## [1] 4
##
## [[3]]
## [1] 5
```

```r
#count the amount of applause in each speech
applause <- lapply(getExtras(), str_count, "Applause")
applause_count <- lapply(applause, sum)

#show times applasue occured in speeches
applause_count[1:3]
```

```
## [[1]]
## [1] 1
##
## [[2]]
## [1] 2
##
## [[3]]
## [1] 4
```

```r
#get the sentences of each speech and store in vector
getSentences <- function(){
  sentences <- lapply(getSpeechesClean(), str_split,
                      "(?<=\\.|\\?|!)\\s+(?=[A-Z])")
  sentences <- lapply(sentences, unlist)

  return(sentences)
}

#show splitting up by sentence and storing in vector
head(unlist(getSentences()))
```

```
## [1] "The President."
## [2] "Thank you."
## [3] "Thank you."
## [4] "Thank you."
## [5] "Good to be back."
## [6] "As Mitch and Chuck will understand, it's good to be almost home, down the hall."
```

```r
#get the words of each speech and store in vector
#I removed punctuation and then found words based on white space
#I also noted the special case of "-" which sperates words
getWords <- function(){

  words <- lapply(getSpeechesClean(), str_replace_all, "\\.|\\?|,|!|:|;", "")
  words <- lapply(words, str_replace_all, "-", " ")

  words <- lapply(words, str_split, "\\s+")
  words <- lapply(words, unlist)

  return(words)

}

#shows splitting up words and storing in vector
```

```r
head(unlist(getWords()))
```

```
## [1] "The"       "President" "Thank"     "you"       "Thank"     "you"
```

```r
#count the number of words of each speech
numWords <- lapply(getWords(), length)
head(unlist(numWords))
```

```
## [1] 8017 5094 5914 5600 6264 6801
```

```r
#count the number of characters
numChar <- lapply(getSpeechesClean(), nchar)
numChar <- lapply(numChar, sum)
head(unlist(numChar))
```

```
## [1] 45577 29029 33913 32782 37430 39307
```

```r
#calculate average word length
aveWordLength <- unlist(numChar)/unlist(numWords)
head(aveWordLength)
```

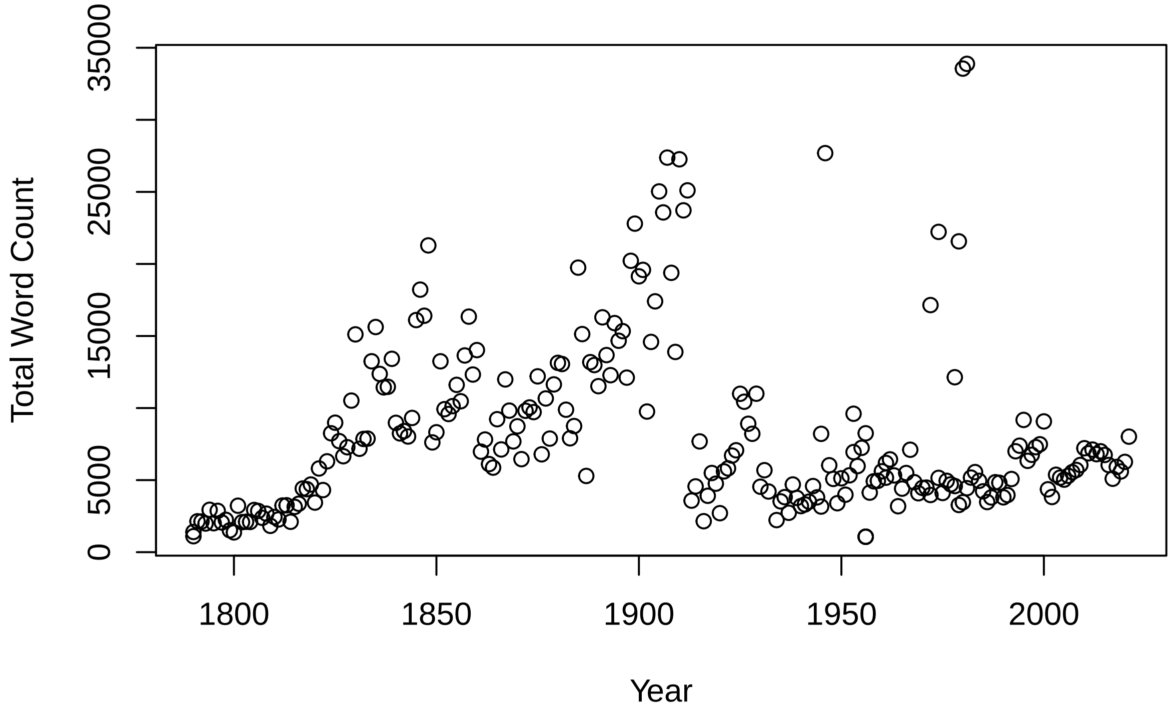
```
## [1] 5.685044 5.698665 5.734359 5.853929 5.975415 5.779591
```

```r
#functions to automate getting word count based on a given regular expression
getWordCount <- function(regex){
  numRegex <- lapply(getWords(), str_count, regex)
  numRegex <- lapply(numRegex, sum)
  return(numRegex)
}

#function to automate adding a word count to a dataframe as another variable
addWordColumns <- function(regexVector, dataframe){

  numWord <- lapply(regexVector, getWordCount)

  for (i in 1:length(regexVector)){
  dataframe[regexVector[i]] <- unlist(numWord[i])
  }
  return(dataframe)
}



#create dataframe with all information on speeches
pres_speech_data <- data.frame(unlist(getSpeechesPresName()),
                               unlist(getSpeechesYear()))

pres_speech_data$laughter_counts = unlist(laughter_count)

pres_speech_data$applause_counts = unlist(applause_count)

pres_speech_data$number_of_words = unlist(numWords)

pres_speech_data$number_of_characters = unlist(numChar)

pres_speech_data$average_word_length = unlist(aveWordLength)
```

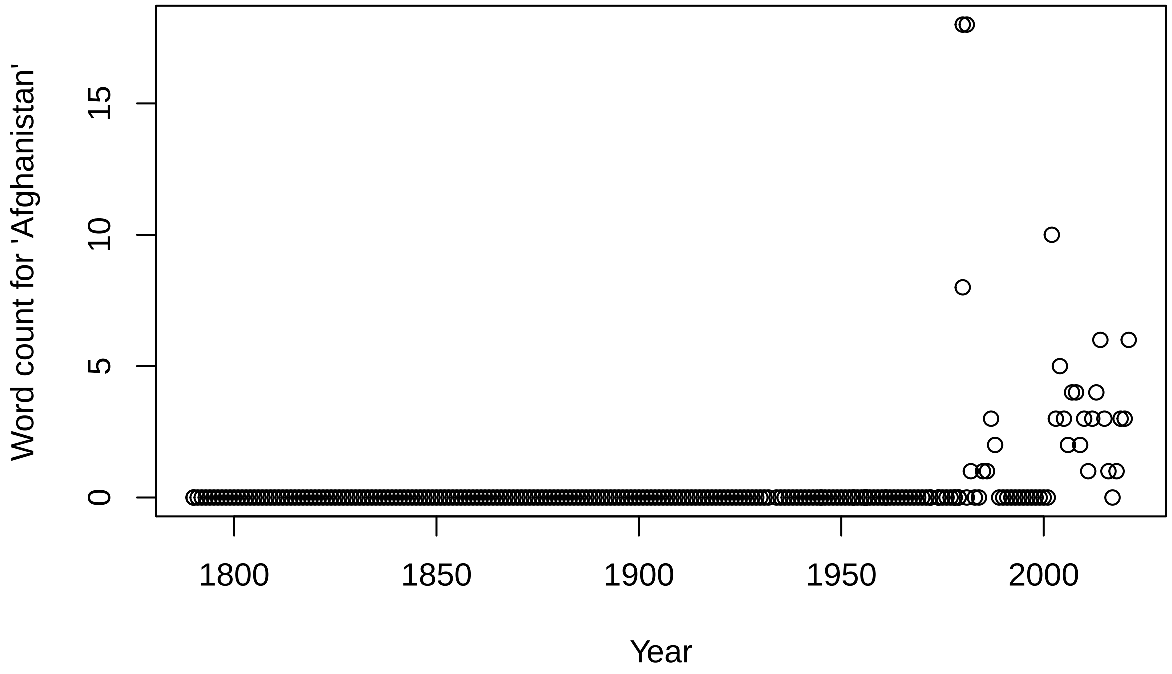
```r
#vector of regular expressions for match all the words specified in
#the problem set
regexVec <- c("^I$", "^we$", "^America$|^American$","^democracy$|^democractic$",
              "^republic$", "^Democrat$","^Republican$",
              "^Free$|^Freedom$", "^war$", "^God$",
              "^Jesus$|^Christ$|^Christian$", "^Afghanistan$")


#add to dataframe using addWordColumns function
all_speech_data <- addWordColumns(regexVec, pres_speech_data)

#add God bless since it is two words and so slightly different handling
numGodBlessCount <- lapply(getSentences(), str_count, "God Bless|God bless")
numGodBless <- lapply(numGodBlessCount, sum)

all_speech_data$"God Bless|God bless" <- unlist(numGodBless)

#note that I simplified the names of the columns to make the df cleaner
colnames(all_speech_data) <- c("name", "year", "laughter_count",
                               "applause_count", "number_of_words",
                               "number_of_characters", "average_word_length",
                               "I_word_count", "we_word_count",
                               "american_word_count", "democracy_word_count",
                               "republic_word_count", "democrat_word_count",
                               "republican_word_count", "freedom_word_count",
                               "war_word_count", "god_word_count",
                               "jesus_word_count", "afghanistan_word_count", "god_bless_word_count")
```

For the plots I created some visualizations looking at overall word count, word counts of different words and I compared the total the amount of times recent republican and democratic presidents said different words.

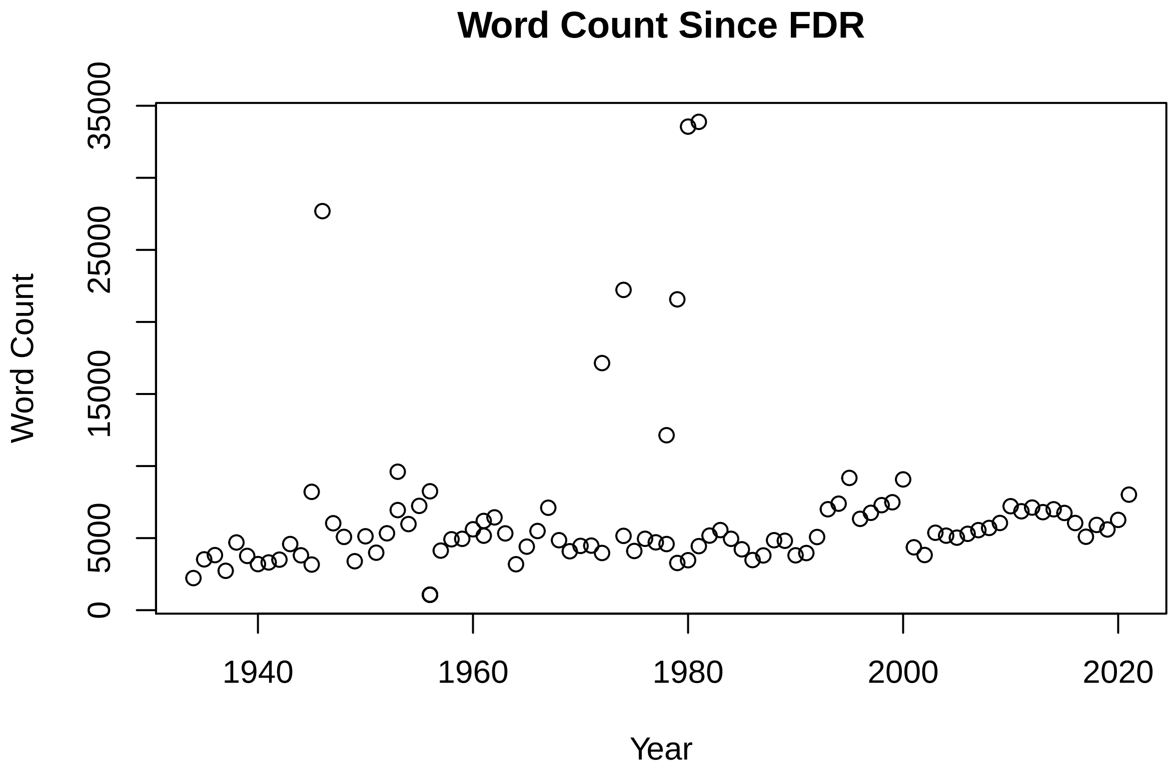
```r
#basic plot for total words for each speech
plot(all_speech_data$year, all_speech_data$number_of_words,
     xlab = "Year", ylab = "Total Word Count")
```



```
plot(all_speech_data$year, all_speech_data$afghanistan_word_count,
     xlab = "Year", ylab = "Word count for 'Afghanistan'")
```



```
#plot since FDR
sinceFDR <- subset(all_speech_data, year > 1932)

plot(sinceFDR$year, sinceFDR$number_of_words, ylab = "Word Count",
     xlab = "Year", main = "Word Count Since FDR")
```



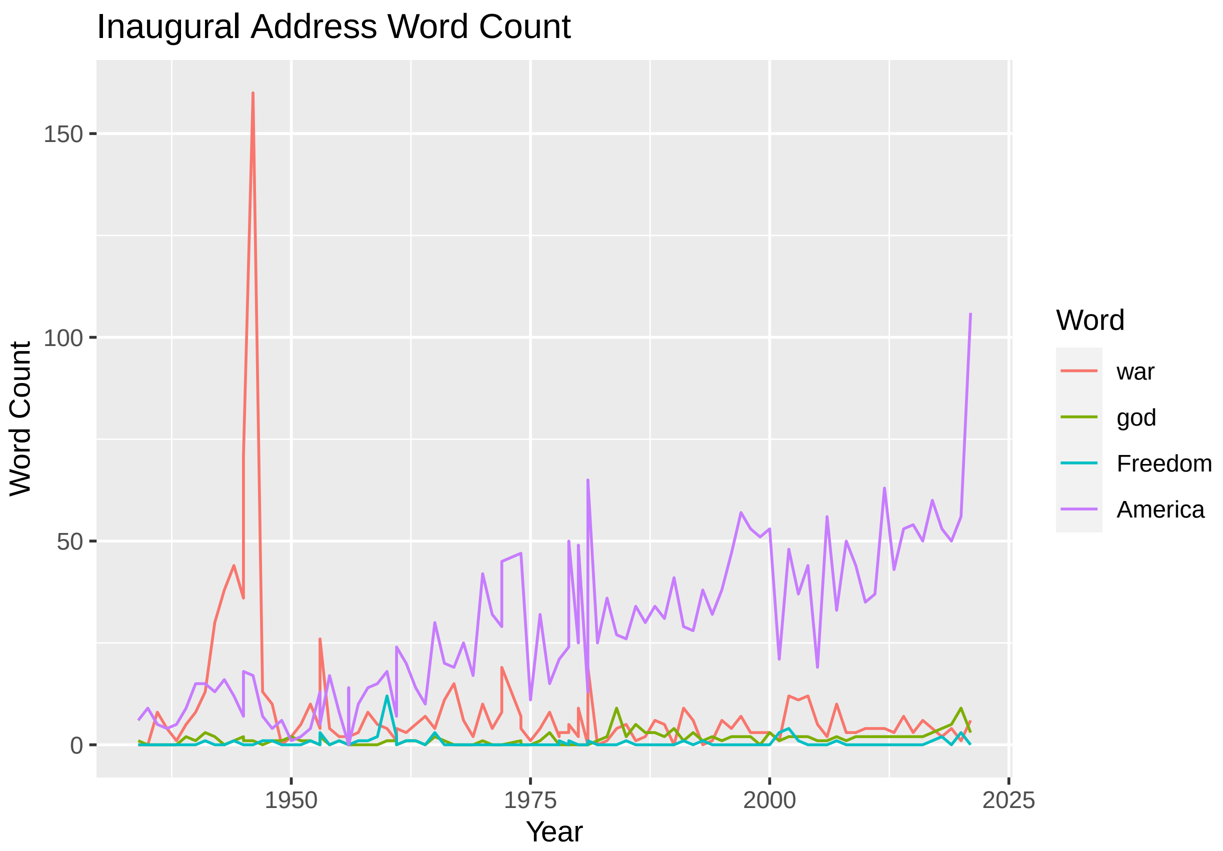
```
#plot showing changes in different words in comparsion to eacother
sinceFDR1 <- select(sinceFDR, "year", "war_word_count",
                    "god_word_count", "freedom_word_count",
                    "american_word_count")

colnames(sinceFDR1) <- c("year", "war", "god", "Freedom", "America")
head(sinceFDR1)
```

```
##   year war god Freedom America
## 1 2021   6   3       0     106
## 2 2017   4   3       1      60
## 3 2018   2   4       2      53
## 4 2019   4   5       0      50
## 5 2020   1   9       3      56
## 6 2013   3   2       0      43
```

```
plot <- ggplot(melt(sinceFDR1),
    aes(x=as.numeric(year),
    y=value, color=variable)) + geom_line() + xlab("Year") + ylab("Word Count")
```

```
## Using year as id variables
```

```
plot + ggtitle("Inaugural Address Word Count") + labs(color = "Word")
```



```
#code to get totals for democratic and republican presidents
demPrez <- c("Franklin D. Roosevelt", "Harry S. Truman",
             "John F. Kennedy", "Lyndon B. Johnson", "Jimmy Carter",
             "William J. Clinton", "Barack Obama", "Joseph R. Biden")

repPrez <- c("Dwight D. Eisenhower", "Richard Nixon",
             "Gerald R. Ford", "Ronald Reagan", "George Bush",
             "George W. Bush", " Donald J. Trump")

sinceFDR2 <- select(sinceFDR, "name", "year", "war_word_count",
                    "god_word_count", "freedom_word_count",
                    "american_word_count")

colnames(sinceFDR2) <- c("name", "year", "war", "god", "Freedom", "America")
head(sinceFDR2)
```

```
##               name year war god Freedom America
## 1 Joseph R. Biden 2021   6   3       0     106
## 2 Donald J. Trump 2017   4   3       1      60
## 3 Donald J. Trump 2018   2   4       2      53
## 4 Donald J. Trump 2019   4   5       0      50
## 5 Donald J. Trump 2020   1   9       3      56
## 6    Barack Obama 2013   3   2       0      43
```

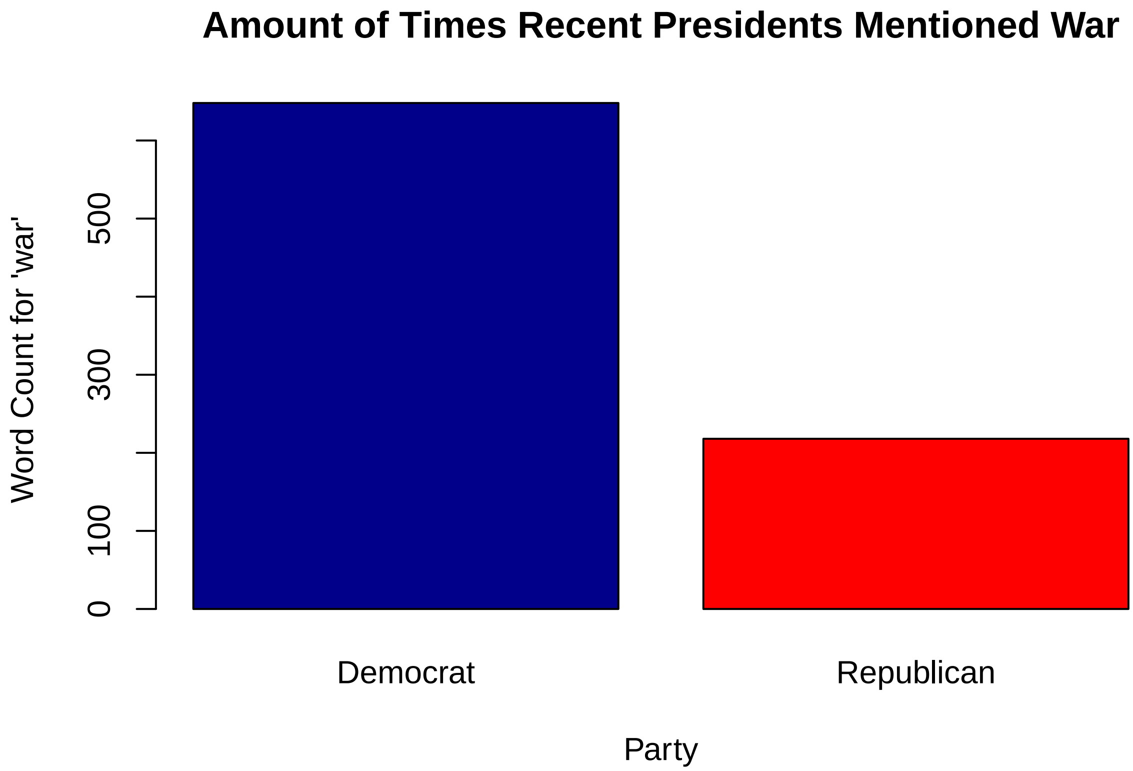
```
totals <- sinceFDR2 %>% group_by(name) %>% summarise(num = n(),
                                                     totalWar = sum(war),
                                                     totalGod = sum(god))
```

```r
#show totals for words
head(totals)
```

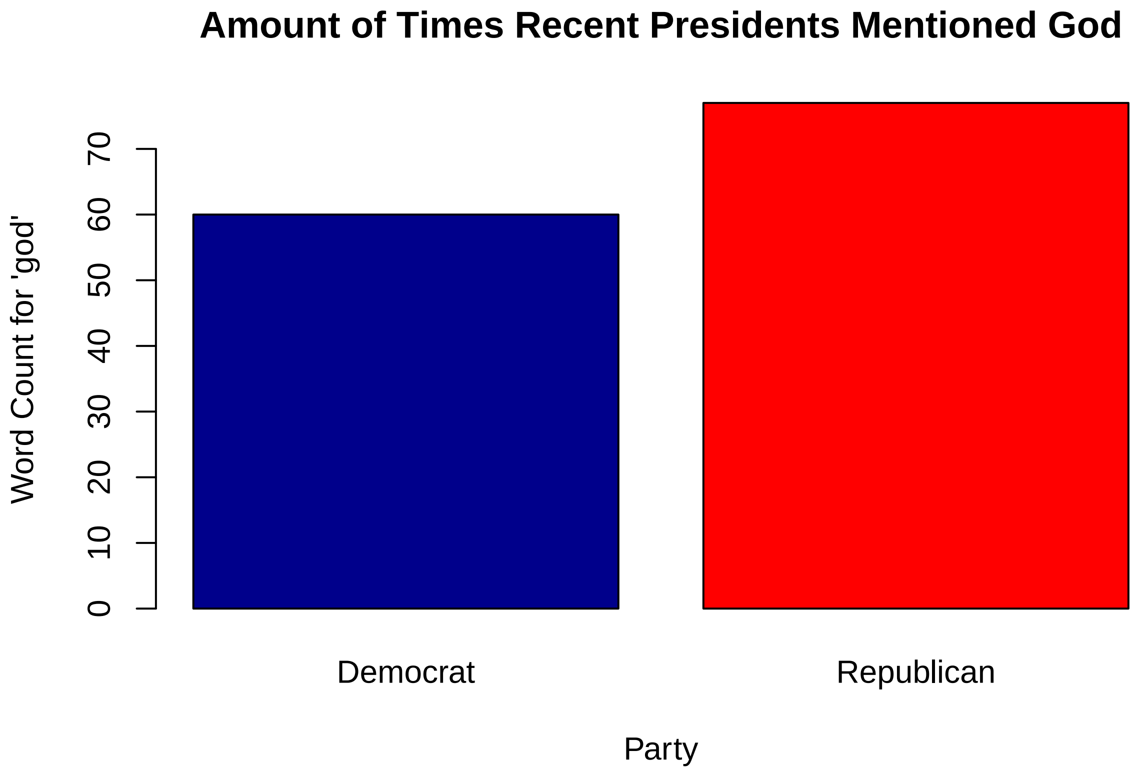
```
## Warning: `...` is not empty.
##
## We detected these problematic arguments:
## * `needs_dots`
##
## These dots only exist to allow future extensions and should be empty.
## Did you misspecify an argument?
```

```
## # A tibble: 6 x 4
##   name                    num totalWar totalGod
##   <chr>                 <int>    <int>    <int>
## 1 Barack Obama              8       34       16
## 2 Donald J. Trump           4       11       21
## 3 Dwight D. Eisenhower     13       47        3
## 4 Franklin D. Roosevelt    13      258       13
## 5 George Bush               4       20       10
## 6 George W. Bush            8       56       12
```

```r
#intialize sum of word counts for Democrats and Republicans
sumDemWar <- 0
sumRepWar <- 0
sumDemGod <- 0
sumRepGod <- 0


for (i in 1:nrow(totals)) {
  if (is.element(totals$name[i], demPrez)) {
    sumDemWar <- sumDemWar + totals$totalWar[i]
    sumDemGod <- sumDemGod + totals$totalGod[i]

  }
  else {
   sumRepWar <- sumRepWar + totals$totalWar[i]
   sumRepGod <- sumRepGod + totals$totalGod[i]
  }
}


war <- c(sumDemWar, sumRepWar)

god <- c(sumDemGod, sumRepGod)


barplot(war, names.arg = c("Democrat", "Republican"),
        main = "Amount of Times Recent Presidents Mentioned War",
        xlab = "Party", ylab = "Word Count for 'war'", col=c("darkblue","red"))
```



```
barplot(god, names.arg = c("Democrat", "Republican"),
        main = "Amount of Times Recent Presidents Mentioned God",
        xlab = "Party", ylab = "Word Count for 'god'", col=c("darkblue","red"))
```


Amount of Times Recent Presidents Mentioned God
Word Count for 'god'
0
10
20
30
40
50
60
70
Democrat
Republican
Party