Homework 6

Aidan Pizzo

10/6/21

#data_munged.R

```
library("stringr")

munge_data <- function()
{
  con <- file("../data/Presidential_Debate_Transcript.txt", "r")
  lines <- readLines(con)
  lines <- setdiff(lines, "")
  close(con)
  ind <- grep("\\(", lines)
  time <- vector("character", length(ind))
  speaker <- vector("character", length(ind))
  their_line <- vector("character", length(ind))
  get_time_speaker <- lines[ind]
  speaker_header <- substr(get_time_speaker, 1, 3)
  for (i in 1:length(ind))
  {
    if (speaker_header[i] == "Chr") #Chris Wallace
    {
      speaker[i] <- "Wallace"
    }
    else if (speaker_header[i] == "Pre") #President Trump
    {
      speaker[i] <- "Trump"
    }
    else if (speaker_header[i] == "Vic") #VP Biden
    {
      speaker[i] <- "Biden"
    }
    time[i] <- get_time(get_time_speaker[i])
    for (i in 1:length(time))
    {
      time[i] <- str_replace(time[i], ":", "min")
      time[i] <- str_replace(time[i], "\\)", "sec")
    }
    their_line <- lines[ind+1]
    out_df <- data.frame(time, speaker, line = their_line)
  }
  return(out_df)
}

get_time <- function( line )
```

```
{
  split <- strsplit(line, split = "")
  start_ind <- grep("\\(", split[[1]])+1
  end_ind <- grep("\\)", split[[1]])
  time <- split[[1]][start_ind:end_ind]
  string_time <- paste(time, collapse= "")
  return(string_time)
}

d1 <- munge_data()
```

```
## Warning in readLines(con): incomplete final line found on '../data/
## Presidential_Debate_Transcript.txt'
```

```
write.csv(d1, "debate.csv", row.names = F)
```

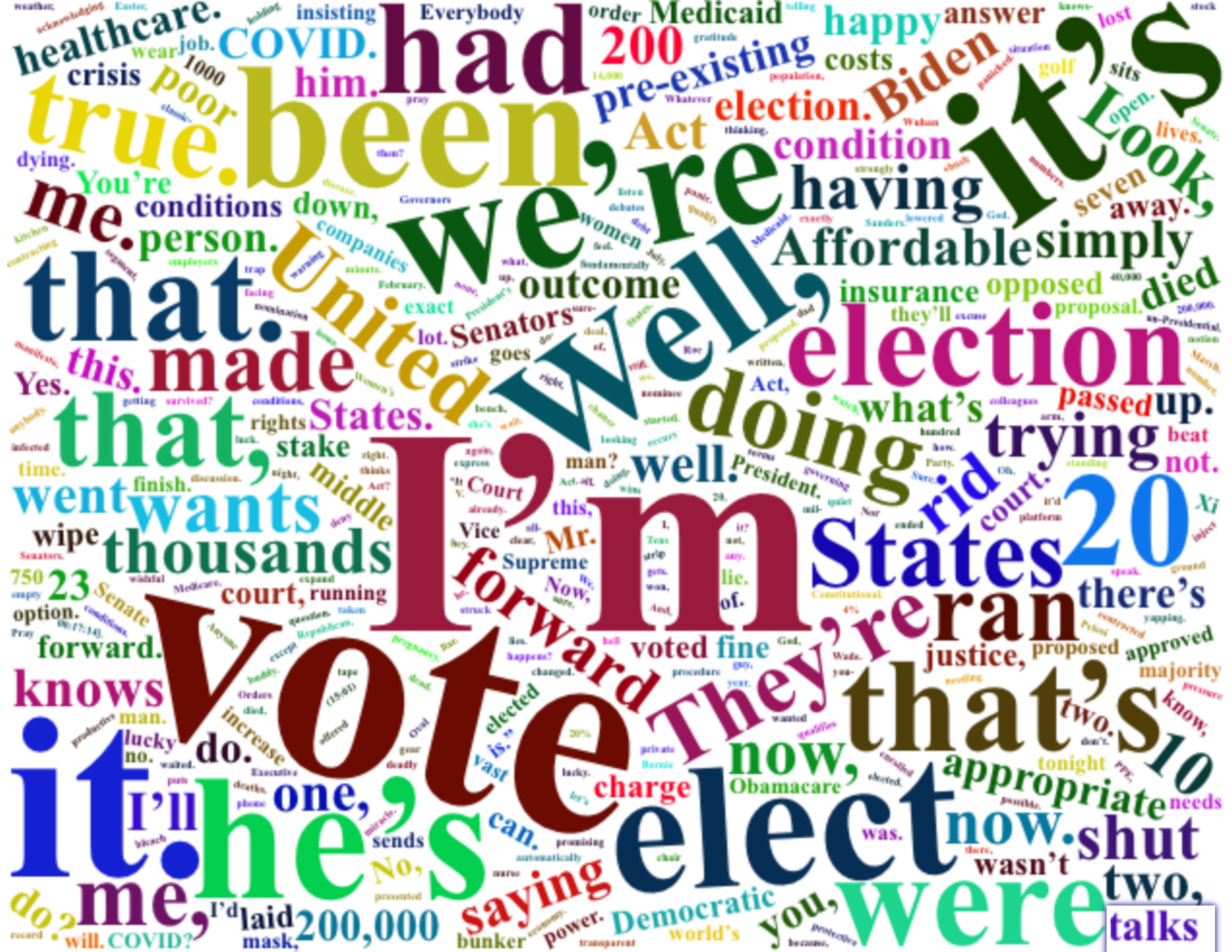
#analysis.R

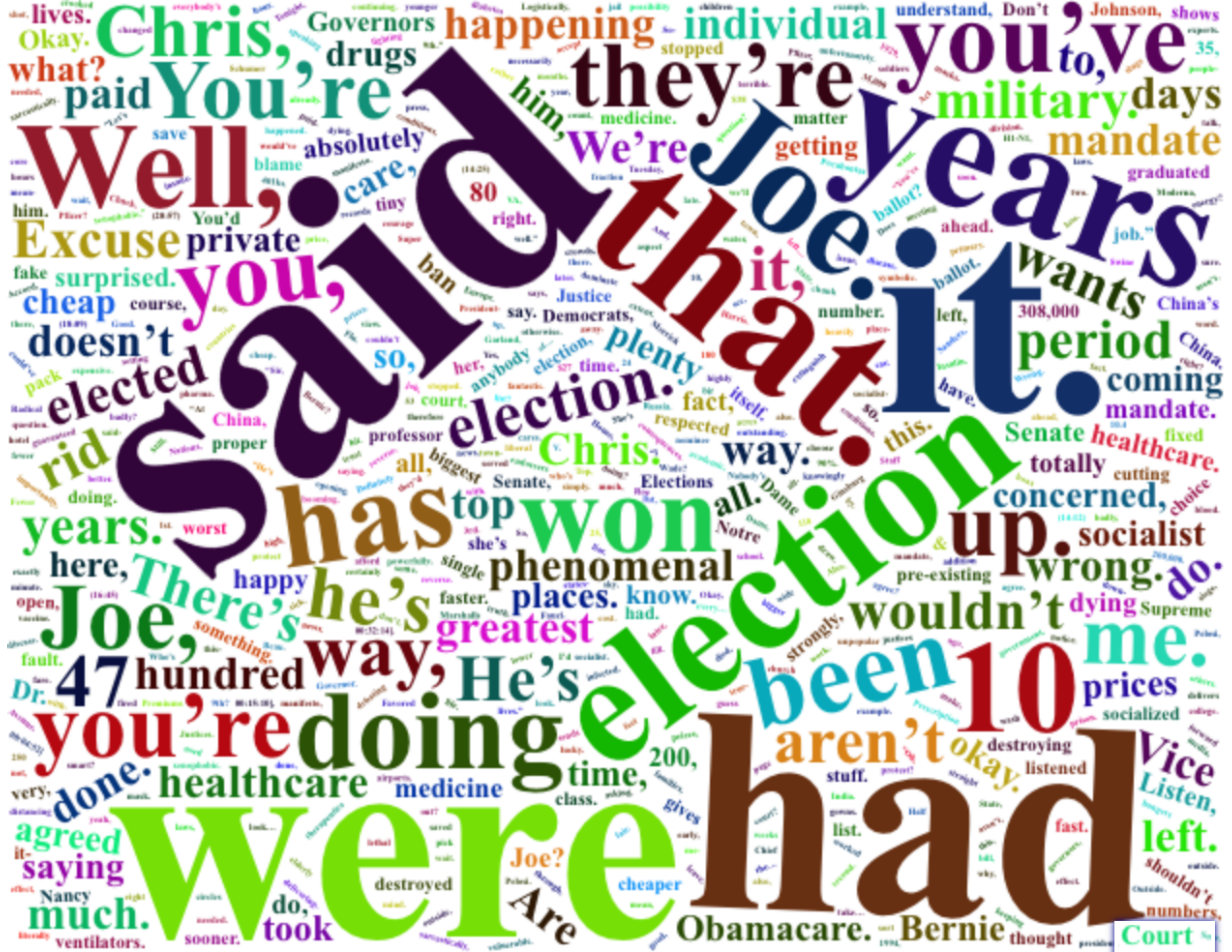
```
library(wordcloud2)

get_word_counts <- function(d, speaker)
{
  speaker_lines_vec <- d[d$speaker == speaker,3]
  split <- strsplit(speaker_lines_vec, " ")
  split <- unlist(split)
  unique_words <- unique(split)
  num_times_said <- vector("numeric", length(unique_words))
  for (i in 1:length(unique_words))
  {
    num_times_said[i] <- length(grep(unique_words[i], split, fixed = T))
  }
  out_df <- data.frame(word = unique_words, count = num_times_said)
  return(out_df)
}

total_word_counts <- function(d, speakers)
{
  if (length(speakers) == 2)
  {
    d1 <- d[d$speaker == speakers[1] | d$speaker == speakers[2], ]
  }
  else if (length(speakers) == 3)
  {
    d1 <- d[d$speaker == speakers[1] | d$speaker == speakers[2]
                    | d$speaker == speakers[3], ]
  }
  split <- strsplit(d1$line, " ")
  split <- unlist(split)
  unique_words <- unique(split)
  num_times_said <- vector("numeric", length(unique_words))
  for (i in 1:length(unique_words))
  {
    num_times_said[i] <- length(grep(unique_words[i], split, fixed = T))
  }
  out_df <- data.frame(word = unique_words, count <- num_times_said)
```

```
  return(out_df)
}

prepare_word_cloud <- function(d, speaker)
{
  if (length(speaker)>1)
  {
    df <- total_word_counts(d, speaker)
  }
  else
  {
    df <- get_word_counts(d, speaker)
  }
  common_words_df <- read.csv("../data/word_frequency.csv", header=T, stringsAsFactors = F, strip.white
  common_words <- toupper(common_words_df[1:500,2]) #vector is returned with spaces in front
  common_words <- clean_words(common_words)
  uncommon_words <- vector("character")
  freq <- vector("numeric")
  for (i in 1:length(df$word))
  {
    if (is.element(toupper(df$word[i]), common_words) == FALSE) #add the word to uncommon if it isn't i
    {
      uncommon_words <- c(uncommon_words, df$word[i])
      freq <- c(freq, df$count[df$word == df$word[i]])
    }
  }
  return(data.frame(uncommon_words, weight = freq))
}

clean_words <- function(words)
{
  for (i in 1:length(words))
  {
    split <- strsplit(words[i], "")
    split <- split[[1]][4:length(split[[1]])]
    the_word<-paste(split, collapse= "")
    words[i] <- the_word
  }
  return(words)
}
```

Wordclouds on next pages:
1. Trump
2. Biden
3. Trump and Biden


They're
Well,
been
has
wants
United
Vice
Joe.
States.
you,
fact,
stake
this,
her,
places.
court.
President.
wouldn't
election.
it,
He's
Court
up.
Chris.
top
time.
House,
Senate,
outcome
had.
biggest
nominee
doing?
Elections
respected
Dame
all,
he's
phenomenal
well.all.
she's
professor
Mr.
plenty
won
know.
way,
anybody
single
There's
greatest
already.
States
Senators
way.
forward
voted
Supreme
Now,
election
rid
middle
Affordable
that.doing


I'm
vote
it's
we're
Well,
been
elect
that's
he's
it.
had
true.
that.
that,
election
doing
States
They're
were
20
United
made
me.
me,
forward
thousands
wants
went
having
Affordable
simply
Look,
trying
ran
now,
now.
shut
appropriate
knows
healthcare.
COVID.
200
pre-existing
Biden
election.
condition
Act
happy
answer
Medicaid
insisting
Everybody
order
costs
crisis
poor
him.
conditions
person.
You're
companies
down,
outcome
insurance
opposed
died
proposal.
Senators
this.
Yes.
States.
rights
stake
middle
wipe
23
court,
running
Supreme
Court
Mr.
Vice
voted
fine
justice,
proposed
approved
majority
there's
10
two,
two.
tonight
Obamacare
charge
one,
I'll
do.
increase
lucky
man.
forward.
option.
750
saying
200,000
laid
Democratic
power.
bunker
world's
wasn't
you,
talks
needs
passed
up.
beat
not.
what's
well.
President.
rid
court.
Xi
seven
away.
golf
lives.
open.
lost
sits
talking
job.
wear
1000
dying.
time.
finish.
exact
lot.
goes
this,
Now,
lie.
of.
elected
vast
can.
sends
No,
do?
will.
COVID?
mask,
I'd


said
were
had
election
it.
that.
years
Joe.
won
has
they're
Well,
you,
You're
you've
doing
you're
Chris,
election.
been
10
Joe,
he's
He's
way,
up.
There's
47
hundred
healthcare
wouldn't
phenomenal
greatest
aren't
me.
period
wants
military.
mandate
individual
happening
Governors
Excuse
private
doesn't
elected
cheap
paid
drugs
Chris.
way.
top
wrong.
socialist
concerned,
prices
Vice
Listen,
left.
Obamacare.
Bernie
okay.
time,
medicine
done.
agreed
saying
much.
took
Are
Court