Applied Statistical Programming - Spring 2022

Jordan Duffin Wong

Problem Set 4

Due Wednesday, March 23, 10:00 AM (Before Class)

DISCLAIMER: I'm doing a lot of copying of Rex Deng's submission for this assignment

tidyverse

```r
# Loading libraries
library(fivethirtyeight)
```

```
## Warning: package ’fivethirtyeight’ was built under R version 4.1.3
```

```
## Some larger datasets need to be installed separately, like senators and
## house_district_forecast. To install these, we recommend you install the
## fivethirtyeightdata package by running:
## install.packages(’fivethirtyeightdata’, repos =
## ’https://fivethirtyeightdata.github.io/drat/’, type = ’source’)
```

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
```

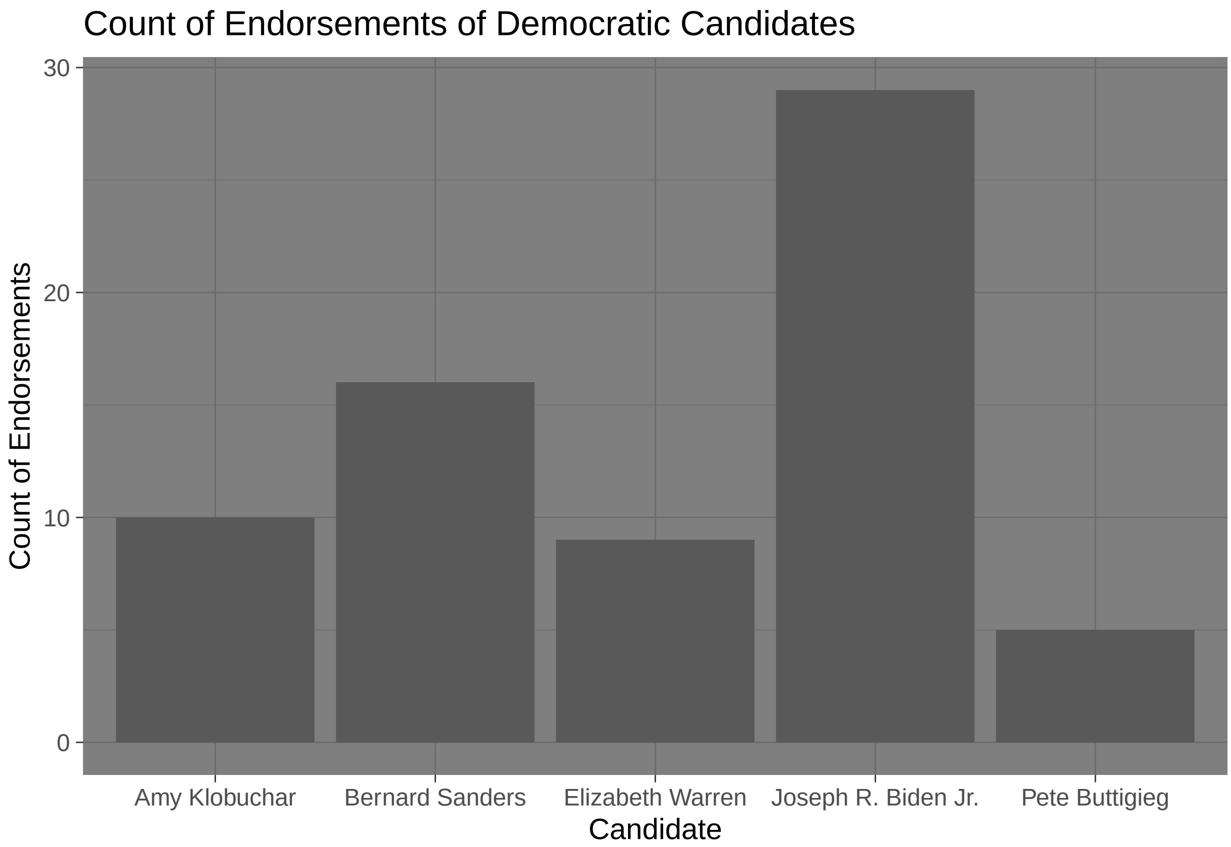
```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
# URL to the data that you've used.
# url <- 'https://jmontgomery.github.io/PDS/Datasets/president_primary_polls_feb2020.csv'
# I'm too lazy to bother with the `connection buffer size` issue, so we're doing this directly

# Creating the `polls` and `Endorsements` objects. Not sure why we want a capital "E", but alright
polls <- read.csv("president_primary_polls_feb2020.csv")
Endorsements <- endorsements_2020 # from the fiverthirtyeight package
```

```r
# Changing the `endorsee` variable to `candidate_name` in `Endorsements`
Endorsements <- Endorsements %>%
  rename(candidate_name = endorsee)

# Making `Endorsements` a tibble
Endorsements <- as_tibble(Endorsements)

# Creating our pool of candidates, and then filtering `polls` to only include them.
# We're also subsetting the data to just five variables:
# `candidate_name`, `sample_size`, `start_date`, `party`, and `pct`

candidates <- c("Amy Klobuchar", "Bernard Sanders", "Elizabeth Warren", "Joseph R. Biden Jr.",
                "Michael Bloomberg", "Pete Buttigieg")

polls <- polls %>%
  filter(candidate_name %in% candidates) %>%
  dplyr::select(candidate_name, sample_size, start_date, party, pct)

# Making sure the names match up across data sets -- this means changing
# "Joe Biden" to "Joeseph R. Biden Jr." and "Bernie Sanders" to "Bernard Sanders"

Endorsements <- Endorsements %>%
  mutate(candidate_name = ifelse(candidate_name == "Joe Biden", "Joseph R. Biden Jr.",
                                 ifelse(candidate_name == "Bernie Sanders", "Bernard Sanders",
                                        candidate_name)))

# And making sure we've captured every candidate
intersect(unique(polls$candidate_name),  unique(Endorsements$candidate_name))
```

```
## [1] "Bernard Sanders"     "Pete Buttigieg"      "Joseph R. Biden Jr."
## [4] "Amy Klobuchar"       "Elizabeth Warren"
```

```r
# That seems to work, although it looks like Bloomberg is not in `Endorsements`

# Joining the datasets by `candidate_name`
polls_endorse <- left_join(polls, Endorsements,
                           by = "candidate_name")

# Counting the number of endorsements
# We're pulling from the initial `Endorsements` because the joined data has duplicates
endorse_count <- Endorsements %>%
  filter(candidate_name %in% candidates) %>%
  group_by(candidate_name) %>%
  summarise(count_endorsements = sum(!is.na(endorser)))

# Plotting: we're condensing all of these into a single step
p <- ggplot(data = endorse_count,
            aes(x = candidate_name,
                y = count_endorsements))+
  geom_bar(stat = "identity")+
  labs(title = "Count of Endorsements of Democratic Candidates",
       x = "Candidate",
       y = "Count of Endorsements")+
  theme_dark() # As a theme_minimal() purist, this pains me
p
```



```
# And saving it
ggsave("PS4_endorsement_counts.png",
       plot = p)
```

```
## Saving 6.5 x 4.5 in image
```

Text-as-Data with tidyverse

```
# Clearing the environment, since we aren't reusing anything
# from part 1
rm(list = ls())

# Libraries
library(tidyverse)
library(tm)
```

```
## Warning: package ’tm’ was built under R version 4.1.3

## Loading required package: NLP

##
## Attaching package: ’NLP’

## The following object is masked from ’package:ggplot2’:
##
##     annotate
```

```
library(lubridate)
```

```
##
## Attaching package: ’lubridate’

## The following objects are masked from ’package:base’:
##
##     date, intersect, setdiff, union
```

```
library(wordcloud)
```

```
## Warning: package ’wordcloud’ was built under R version 4.1.3

## Loading required package: RColorBrewer
```

```
# Getting our data
trump_tweets_url <- 'https://politicaldatascience.com/PDS/Datasets/trump_tweets.csv'
tweets <- read_csv(trump_tweets_url)
```

```
## Rows: 32974 Columns: 6

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): source, text, created_at
## dbl (2): retweet_count, favorite_count
## lgl (1): is_retweet

##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```r
# Separating `created_at` where date and times are separate columns
tweets$date <- sapply(strsplit(tweets$created_at, " "), `[[`, 1)
tweets$date <- as.Date(tweets$date,
                       format = "%m/%d/%y")

tweets$time <- sapply(strsplit(tweets$created_at, " "), `[[`, 2)

# Reporting the range
range(tweets$date)
```

```
## [1] "2020-01-01" "2020-12-31"
```

```r
# Removing retweets and displaying Trump's `top 5` most popular arnd retweeted tweets.
topfive <- tweets %>%
  filter(is_retweet == FALSE) %>%
  slice_max(retweet_count, n = 5)

# Creating the `corpus`
# Of all the things I copied from Rex, this is this is easily the most-copied
corpus <- VCorpus(VectorSource(tweets$text))
writeLines(head(strwrap(corpus[[1]]), 10)) # Checking that we pulled the content correctly
```

```
## RT @DailyCaller: ‘Why Would I Not:’ Chiefs’ Bashaud Breeland Looking
## Forward To WH Visit After Super Bowl Win https://t.co/0t9bdLQKDn
```

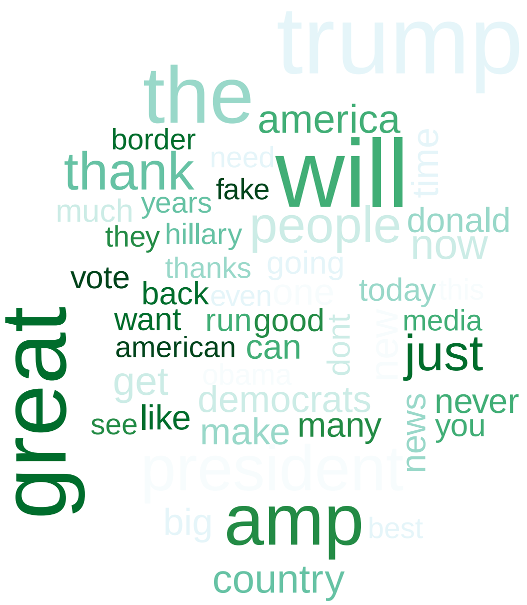
```r
# Removing whitespace, numbers, and other text cleaning
# `addspace` finds whatever pattern we want and replaces it with a space
addspace <- content_transformer(function(x, pattern){
  return(gsub(pattern, " ", x))
})

# For instance, changing `-` to whitespace
corpus <- tm_map(corpus, addspace, "-")

# Removing patterns -- basically the opposite of `addspace()`
removepattern <- content_transformer(function(x, pattern){
  return(gsub(pattern, "", x))
})

# using it to remove URLs
corpus <- tm_map(corpus, removepattern, "?(f|ht)(tp)(s?)(://)(.*)(.|/])(.*)")

# and to remove the other stuff
corpus <- tm_map(corpus, removepattern, "‘")
corpus <- tm_map(corpus, removepattern, "’")
corpus <- tm_map(corpus, stripWhitespace)
corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, removeNumbers)
corpus <- tm_map(corpus, removeWords, stopwords("english"))

# changing the case
corpus <- tm_map(corpus, content_transformer(tolower))
```

```
# And checking, again, that it worked
writeLines(head(strwrap(corpus[[1]]), 10)) # lol this is such a mess of a tweet how did this guy become
```

```
## rt dailycaller why would i not chiefs bashaud breeland looking forward
## to wh visit after super bowl win
```

```
# Making the `wordcloud`, where words have a minimum of 3 appearances
pal = brewer.pal(9, "BuGn")
wc <- wordcloud(corpus, min.freq = 3,
                random.order = TRUE,
                random.color = TRUE,
                max.words = 50,
                colors = pal)
```

```
## Warning in wordcloud(corpus, min.freq = 3, random.order = TRUE, random.color =
## TRUE, : realdonaldtrump could not be fit on page. It will not be plotted.
```



```
# Making our DTM with `control = list(weighting = weighTfIdf)`
library(tidytext) # we need this for tidy()
```

```
## Warning: package 'tidytext' was built under R version 4.1.3
```

```
DTM <- DocumentTermMatrix(corpus,
                          control = list(weighting = weightTfIdf))
```

```
## Warning in weighting(x): empty document(s): 22 216 237 251 283 287 293 295 297
## 327 381 528 529 530 531 537 538 543 583 587 589 590 633 634 653 705 780 839 840
## 842 1023 1050 1076 1133 1201 1204 1205 1206 1211 1232 1233 1235 1354 1355 1369
## 1401 1421 1603 1604 1650 1651 1653 1655 1664 1797 1853 1868 1869 1915 1958 1961
## 1962 2200 2364 2365 2437 2533 2584 2600 2603 2606 2627 2633 2691 2692 2693 2711
## 2734 2736 2769 2775 2790 2819 2844 2854 2862 2865 2866 2868 2872 2880 2881 2882
## 2884 2886 2888 2889 2912 2913 2914 2939 2941 2964 2965 2966 2967 2968 3034 3099
## 3100 3105 3106 3107 3124 3212 3232 3334 3354 3466 3566 3567 3579 3596 3605 3662
## 3679 3755 3757 3759 3774 3775 3782 3787 3795 3847 3849 3893 3908 3925 3956 3959
## 3960 3961 4007 4008 4081 4082 4101 4140 4141 4158 4202 4244 4245 4268 4269 4270
## 4283 4294 4296 4323 4333 4339 4366 4431 4448 4451 4473 4475 4489 4490 4492 4516
## 4564 4573 4581 4582 4583 4586 4587 4588 4643 4644 4659 4660 4662 4665 4681 4683
## 4700 4710 4714 4718 4728 4739 4740 4742 4744 4745 4754 4775 4776 4819 4891 4962
## 4966 5038 5047 5074 5078 5079 5127 5184 5310 5321 5330 5382 5388 5549 5657 5658
## 5672 5832 5919 5924 5926 5956 6018 6025 6042 6044 6085 6089 6097 6098 6139 6155
## 6157 6182 6223 6285 6455 6456 6544 6592 6600 6618 6626 6652 6667 6728 6729 6734
## 6739 6753 6813 6833 6946 6947 6954 6955 7018 7019 7053 7136 7172 7176 7219 7263
## 7452 7458 7725 7969 7979 7982 8010 8129 8215 8261 8262 8270 8271 8276 8279 8280
## 8299 8300 8315 8316 8318 8330 8332 8345 8351 8352 8355 8358 8387 8389 8412 8414
## 8416 8417 8419 8430 8447 8522 8614 8638 8639 8726 8758 8759 8774 8775 8777 8789
## 8790 8809 8811 8863 8893 8952 9026 9037 9051 9063 9108 9135 9185 9204 9228 9230
## 9632 9875 9998 10008 10070 10071 10072 10073 10130 10140 10154 10155 10156 10183
## 10202 10203 10250 10270 10281 10286 10307 10314 10328 10329 10339 10363 10365
## 10376 10386 10400 10504 10523 10658 10705 10750 10763 10764 10765 10772 10773
## 10775 10776 10798 10816 10820 10829 10894 10943 11068 11090 11260 11344 11528
## 11632 11703 11748 12181 12280 14686 18463 18507 19256 20160 20307 22897 26594
## 29107
```

```
dat <- tidy(DTM)

# Finally, getting our top 50 words with the highest tf.idf scores, and a lfb of 0,8
dat_top50 <- dat %>%
  slice_max(count, n = 50)
head(dat_top50)
```

```
## # A tibble: 6 x 3
##   document term                 count
##   <chr>    <chr>                <dbl>
## 1 1315     winred                15.0
## 2 1521     debport               15.0
## 3 3310     iranintlar            15.0
## 4 4248     donothingdemocrats    15.0
## 5 4408     donothingdems         15.0
## 6 4461     fakewhistleblower     15.0
```