Applied Statistical Programming - Spring 2022

Rex Deng

Problem Set 4

Due Wednesday, March 16, 10:00 AM (Before Class)

Instructions

1. The following questions should each be answered within an Rmarkdown file. Be sure to provide many comments in your code blocks to facilitate grading. Undocumented code will not be graded.
2. Work on git. Continue to work in the repository you forked from https://github.com/johnsontr/AppliedStatisticalProgramming2022 and add your code for Problem Set 4. Commit and push frequently. Use meaningful commit messages because these will affect your grade.
3. You may work in teams, but each student should develop their own Rmarkdown file. To be clear, there should be no copy and paste. Each keystroke in the assignment should be your own.
4. For students new to programming, this may take a while. Get started.

`tidyverse`

Your task in this problem set is to combine two datasets in order to observe how many endorsements each candidate received using only `dplyr` functions. Use the same Presidential primary polls that were used for the in class worksheets on February 28 and March 2.

```
# Change eval=FALSE in the code block. Install packages as appropriate.
# install.packages('fivethirtyeight')
rm(list = ls())
library(fivethirtyeight)
```

```
## 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')
```

```
library(tidyverse)
```

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

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     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
library(RColorBrewer)
library(webshot)
library(htmlwidgets)
library(tidytext)
# URL to the data that you've used.
polls <- read.csv("/Users/rexdeng/Dropbox/mine/academics_career/WashU/Classes/202122Spring-Stat programm
    na.strings = "")

`%notin%` <- Negate(`%in%`)
```

First, create two new objects `polls` and `Endorsements`. Then complete the following.

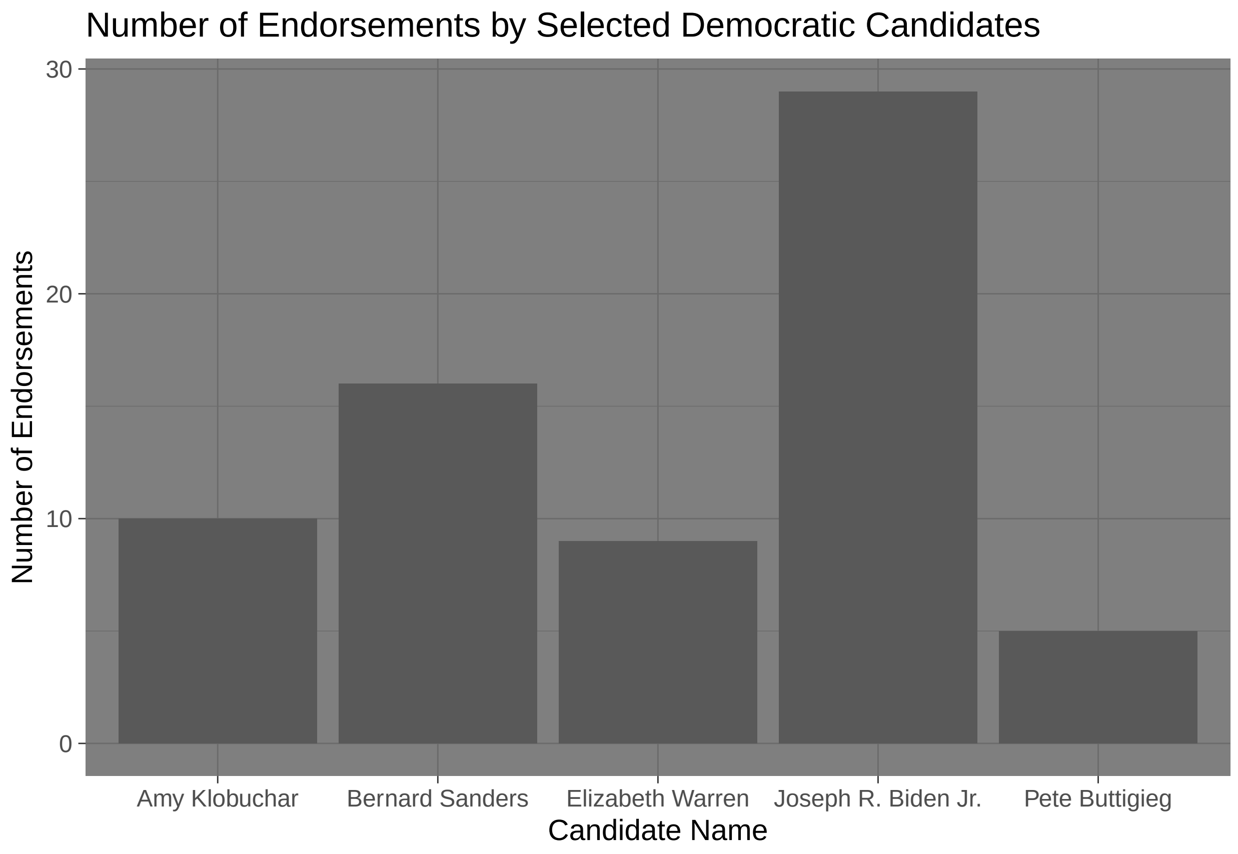
- Change the `Endorsements` variable name endorsee to `candidate_name`.
- Change the `Endorsements` dataframe into a `tibble` object.
- Filter the `poll` variable to only include the following 6 candidates: Amy Klobuchar, Bernard Sanders,Elizabeth Warren, Joseph R. Biden Jr., Michael Bloomberg, Pete Buttigieg **and** subset the dataset to the following five variables: `candidate_name, sample_size, start_date, party, pct`
- Compare the candidate names in the two datasets and find instances where the a candidates name is spelled differently i.e. Bernard vs. Bernie. Using only `dplyr` functions, make these the same across datasets.
- Now combine the two datasets by candidate name using `dplyr` (there will only be five candidates after joining).
- Create a variable which indicates the number of endorsements for each of the five candidates using `dplyr`.
- Plot the number of endorsement each of the 5 candidates have using `ggplot()`. Save your plot as an object `p`.
- Rerun the previous line as follows: `p + theme_dark()`. Notice how you can still customize your plot without rerunning the plot with new options.
- Now, using the knowledge from the last step change the label of the X and Y axes to be more informative, add a title. Save the plot in your forked repository.

```r
# 0 Reset the data frame when running this chunk
Endorsements <- endorsements_2020  # from the fiverthirtyeight package

# 1.1 Rename
Endorsements <- Endorsements %>%
    rename(candidate_name = endorsee)

# 1.2 as.tibble
Endorsements <- as_tibble(Endorsements)

# 1.3 filter and select
polls <- polls %>%
    filter(candidate_name %in% c("Amy Klobuchar", "Bernard Sanders", "Elizabeth Warren",
        "Joseph R. Biden Jr.", "Michael Bloomberg", "Pete Buttigieg")) %>%
    dplyr::select(candidate_name, sample_size, start_date, party, pct)

# 1.4 Make names the same
Endorsements <- Endorsements %>%
    mutate(candidate_name = ifelse(candidate_name == "Joe Biden", "Joseph R. Biden Jr.",
        ifelse(candidate_name == "Bernie Sanders", "Bernard Sanders", candidate_name)))
```

```r
## check which candidate names in polls are still not in Endorsements
unique(polls$candidate_name)[unique(polls$candidate_name) %notin% unique(Endorsements$candidate_name)]
```

```
## [1] "Michael Bloomberg"
```

```r
# 1.5 Join
polls_Endorse_joined <- left_join(polls, Endorsements, by = "candidate_name")

# 1.6 The number of endorsements for each of the five candidates The joined
# dataset is not as useful as the original endorsement dataset as the joined
# one contains duplication of endorsers due to multiple matches
Endorsements_sum <- Endorsements %>%
    filter(candidate_name %in% c("Amy Klobuchar", "Bernard Sanders", "Elizabeth Warren",
        "Joseph R. Biden Jr.", "Michael Bloomberg", "Pete Buttigieg")) %>%
    group_by(candidate_name) %>%
    summarise(n_endorsements = sum(!is.na(endorser)))

# 1.7 ggplot
p <- Endorsements_sum %>%
    ggplot() + geom_bar(aes(x = candidate_name, y = n_endorsements), stat = "identity")

# 1.8, 1.9 add a theme and labs and so on
p <- p + labs(title = "Number of Endorsements by Selected Democratic Candidates",
    x = "Candidate Name", y = "Number of Endorsements") + theme_dark()

p
```



```r
ggsave("Q1.png", width = 10)
```

```
## Saving 10 x 4.5 in image
```

Text-as-Data with tidyverse

For this question you will be analyzing Tweets from President Trump for various characteristics. Load in the following packages and data:

```r
# Change eval=FALSE in the code block. Install packages as appropriate.
library(tidyverse)
library(tm)
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

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

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```r
library(wordcloud)
library(knitr)
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

- First separate the `created_at` variable into two new variables where the date and the time are in separate columns. After you do that, then report the range of dates that is in this dataset.
- Using `dplyr` subset the data to only include original tweets (remove retweents) and show the text of the President's **top 5** most popular and most retweeted tweets. (Hint: The `match` function can help you find the index once you identify the largest values.)
- Create a *corpus* of the tweet content and put this into the object `Corpus` using the `tm` (text mining) package. (Hint: Do the assigned readings.)
- Remove extraneous whitespace, remove numbers and punctuation, convert everything to lower case and remove 'stop words' that have little substantive meaning (the, a, it).
- Now create a `wordcloud` to visualize the top 50 words the President uses in his tweets. Use only words that occur at least three times. Display the plot with words in random order and use 50 random colors. Save the plot into your forked repository.

- Create a *document term matrix* called `DTM` that includes the argument `control = list(weighting = weightTfIdf)`
- Finally, report the 50 words with the the highest tf.idf scores using a lower frequency bound of .8.

```
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.
```

```
# 2.1 Separate date and time
tweets$date <- sapply(strsplit(tweets$created_at, " "), `[[`, 1)
tweets$time <- sapply(strsplit(tweets$created_at, " "), `[[`, 2)

tweets$date <- as.Date(tweets$date, format = "%m/%d/%Y")
range(tweets$date)
```

```
## [1] "2014-01-01" "2020-02-14"
```

```
# 2.2 Get the most retweeted tweets Not sure if most popular == most retweeted
# here though, but shouls use a similar syntax of codes to get those
top5_retweeted <- tweets %>%
    filter(is_retweet == F) %>%
    slice_max(retweet_count, n = 5)

top5_retweeted$text
```

```
## [1] "#FraudNewsCNN #FNN https://t.co/WYUnHjjUjg"
## [2] "TODAY WE MAKE AMERICA GREAT AGAIN!"
## [3] "Why would Kim Jong-un insult me by calling me \"old\" when I would NEVER call him \"short and f
## [4] "A$AP Rocky released from prison and on his way home to the United States from Sweden. It was a 
## [5] "Such a beautiful and important evening! The forgotten man and woman will never be forgotten aga
```

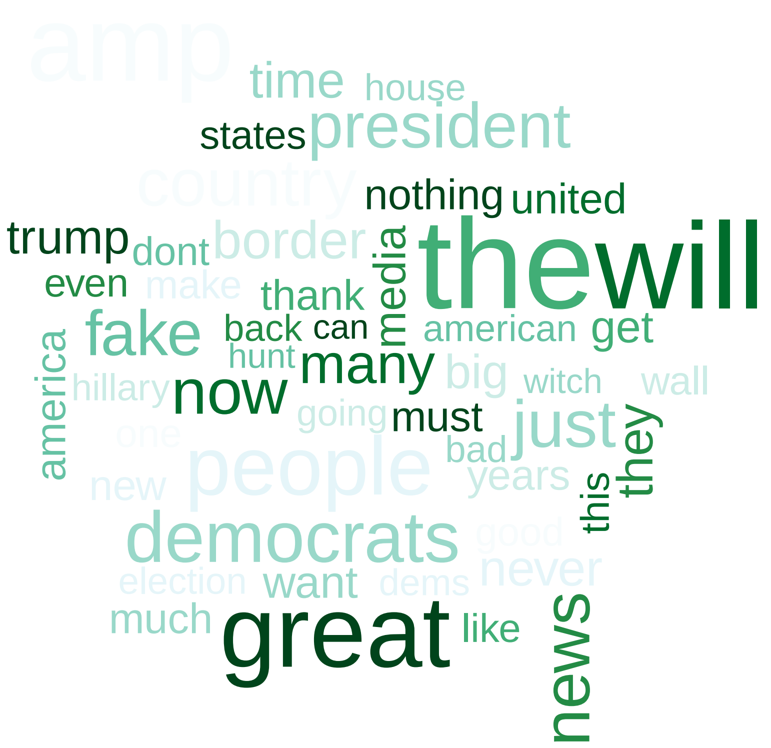
```
# 2.3 Corpus
most_retweeted <- tweets %>%
    filter(is_retweet == F) %>%
    slice_max(retweet_count, n = 5000)

Corpus <- VCorpus(VectorSource(most_retweeted$text))
writeLines(head(strwrap(Corpus[[1]]), 10))  ## Check the tweets
```

```
## #FraudNewsCNN #FNN https://t.co/WYUnHjjUjg
```

```
# 2.4 Remove Create a function called 'addspace' that finds a user specified
# pattern and substitutes the pattern with a space.
addspace <- content_transformer(function(x, pattern) {
    return(gsub(pattern, " ", x))
})
## For words connected by '-', replace it with a white space so that it won't
## be connected after space removal
Corpus <- tm_map(Corpus, addspace, "-")

## Remove patterns
removepattern <- content_transformer(function(x, pattern) {
    return(gsub(pattern, "", x))
})

## Remove urls
Corpus <- tm_map(Corpus, removepattern, "?(f|ht)(tp)(s?)(://)(.*)[.|/](.*)")

## Remove quotes
Corpus <- tm_map(Corpus, removepattern, "‘")
Corpus <- tm_map(Corpus, removepattern, "’")

## Remove others
Corpus <- tm_map(Corpus, stripWhitespace)
Corpus <- tm_map(Corpus, removePunctuation)
Corpus <- tm_map(Corpus, removeNumbers)
Corpus <- tm_map(Corpus, removeWords, stopwords("english"))

## Change all words to lower cases
Corpus <- tm_map(Corpus, content_transformer(tolower))

writeLines(head(strwrap(Corpus[[1]]), 10))  ## Check the tweets again
```

```
## fraudnewscnn fnn
```

```
# 2.5 wordcloud
pal = brewer.pal(9, "BuGn")
wordcloud(Corpus, min.freq = 3, random.order = T, random.color = T, max.words = 50,
    colors = pal)
```



```r
# 2.6 dtm term-frequency dfm
dtm_tf <- DocumentTermMatrix(Corpus)

## tfidf weighted dtm
dtm_tfidf <- DocumentTermMatrix(Corpus, control = list(weighting = weightTfIdf))

## Convert tf dfm to a data frame
dtm_tf_df <- tidy(dtm_tf)

## Get the tf.idf scores for each term
dtm_tfidf_df <- bind_tf_idf(dtm_tf_df, term = term, document = document, n = count)

df_DTM_top50 <- dtm_tfidf_df %>%
    group_by(term) %>%
    filter(tf_idf == max(tf_idf)) %>%
    ungroup() %>%
    slice_max(tf_idf, n = 50)

kable(df_DTM_top50, digits = 3, align = "c", booktabs = T) %>%
    kable_styling(position = "center")
```

document	term	count	tf	idf	tf_idf
414	merrychristmas	1	1.0	8.495	8.495
981	newhoaxsameswamp	1	1.0	8.495	8.495
1572	stopthecoup	1	1.0	8.495	8.495
3637	godblesstheusa	1	1.0	8.495	8.495
3856	usstatevisit	1	1.0	8.495	8.495
3977	remembering	1	1.0	8.495	8.495
4636	donothingdems	1	1.0	8.495	8.495
4675	ddaythanniversary	1	1.0	8.495	8.495
69	standforouranthem	1	1.0	7.802	7.802
526	stopthebias	1	1.0	7.802	7.802
3422	salutetoamericajulyth	1	1.0	7.802	7.802
4782	memorialday	1	1.0	7.802	7.802
588	veto	1	1.0	7.396	7.396
2809	electionday	1	1.0	7.396	7.396
3901	marchforlife	1	1.0	7.396	7.396
38	abeshinzo	1	1.0	7.108	7.108
241	revealing	1	1.0	7.108	7.108
1702	jobsnotmobs	1	1.0	7.108	7.108
4351	jobsnotmobs	1	1.0	7.108	7.108
623	sotu	1	1.0	6.885	6.885
1458	crookedhillary	1	1.0	6.885	6.885
1836	sotu	1	1.0	6.885	6.885
4880	mariabartiromo	1	1.0	6.885	6.885
57	boring	1	1.0	6.415	6.415
1002	treason	1	1.0	6.010	6.010
911	amazing	1	1.0	5.127	5.127
4009	only	1	1.0	5.094	5.094
707	incredible	1	1.0	4.666	4.666
129	maga	1	1.0	4.583	4.583
1	fnn	1	0.5	8.495	4.247
315	diary	1	0.5	8.495	4.247
2165	throwbackthursday	1	0.5	8.495	4.247
2248	sleeping	1	0.5	8.495	4.247
2248	whereshillary	1	0.5	8.495	4.247
2262	gosakasummit	1	0.5	8.495	4.247
3802	wholeheartedly	1	0.5	8.495	4.247
4313	geauxtigers	1	0.5	8.495	4.247
1375	congratulations	1	1.0	4.151	4.151
3201	wow	1	1.0	4.006	4.006
1	fraudnewscnn	1	0.5	7.802	3.901
285	happyindependenceday	1	0.5	7.802	3.901
2364	jobs	3	1.0	3.841	3.841
2644	jobs	3	1.0	3.841	3.841
2975	jobs	3	1.0	3.841	3.841
3438	jobs	3	1.0	3.841	3.841
3585	jobs	3	1.0	3.841	3.841
4035	jobs	3	1.0	3.841	3.841
81	greenland	1	0.5	7.396	3.698
467	easter	1	0.5	7.396	3.698
1673	eric	1	0.5	7.396	3.698
2396	repcummings	1	0.5	7.396	3.698
3955	hanukkah	1	0.5	7.396	3.698