Problem Set 3

Due February 27, 10:00 AM (Before Class)

Instructions

1. The following questions should each be answered within an R script. Be sure to provide many comments in the script to facilitate grading. Undocumented code will not be graded.
2. Work on git. Fork the repository found at https://github.com/domlockett/PDS-PS3 and add your code, committing and pushing frequently. Use meaningful commit messages – these may affect your grade.
3. You may work in teams, but each student should develop their own R script. To be clear, there should be no copy and paste. Each keystroke in the assignment should be your own.
4. If you have any questions regarding the Problem Set, contact the TAs or use their office hours.
5. For students new to programming, this may take a while. Get started.
6. You will need to install `ggplot2` and `dplyr` to complete this dataset.

`ggplot2`

1. Finish the exercise we started in class on 2/11/2020:
   - Alabama, Arkansas, California, Colorado, Maine, Massachusetts, Minnesota, North Carolina, Oklahoma, Tennessee, Texas, Utah, Vermont, and Virginia will all hold their primraries on March 3
   - You have been assigned to create a visulaization of the state of the race for this date.
   - You will make a plot to show this.
   - In addition to the kinds of issues discussed above
     - Change to the minimial theme
     - Figure out how to change the axis labels and legends beyond the defaults
   - Visit https://ggplot2.tidyverse.org/reference/
2. Finish the exercise we started in class on 2/13/2020: -Re-organize the dataset so that there is only one row for each candidate-state dyad -Feel free to limit this down to only the relevant candidates -Compare the size of this dataset to our original dataset using the object_size command.

`dplyr`

1. Now you are going to combine two datasets in order to observe how many endorsements each candidate recieved using **only** `dplyr` functions.
   - Create two new objects `polls` and `Endorsements`:

```
library(fivethirtyeight)
library(tidyverse)
```

```
## -- Attaching packages --------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

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

```
## x dplyr::lag()    masks stats::lag()
```

```
polls <- read_csv('https://jmontgomery.github.io/PDS/Datasets/president_primary_polls_feb2020.c
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   question_id = col_double(),
##   poll_id = col_double(),
##   cycle = col_double(),
##   pollster_id = col_double(),
##   sponsor_ids = col_number(),
##   pollster_rating_id = col_double(),
##   sample_size = col_double(),
##   sponsor_candidate = col_logical(),
##   internal = col_logical(),
##   partisan = col_logical(),
##   tracking = col_logical(),
##   nationwide_batch = col_logical(),
##   candidate_id = col_double(),
##   pct = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

```
Endorsements <- endorsements_2020
```

- Change the `Endorsements` variable name `endorsee` to `candidate_name`

```
colnames(Endorsements)[colnames(Endorsements) == 'endorsee'] <- 'candidate_name'
```

- Change the Endorsement dataframe into a `tibble` object.

```
polls <- as_tibble(polls)
Endorsements <- as_tibble(Endorsements)
```

- 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`

```
polls <- filter(polls, candidate_name %in% c("Amy Klobuchar","Bernard Sanders",
"Elizabeth Warren","Joseph R. Biden Jr.","Michael Bloomberg","Pete Buttigieg")) %>%
  select(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.

```
polls$candidate_name <-  recode(polls$candidate_name, "Joseph R. Biden Jr."
= "Joe Biden", "Bernard Sanders" = "Bernie Sanders")
```

- Now combine the two datasets by candidate name using `dplyr` (there will only be five candidates after joining).

```
polls<-polls %>%
  inner_join(Endorsements, by= 'candidate_name')
```

- Create a variable which indicates the number of endorsements for each of the five candidates using `dplyr`.

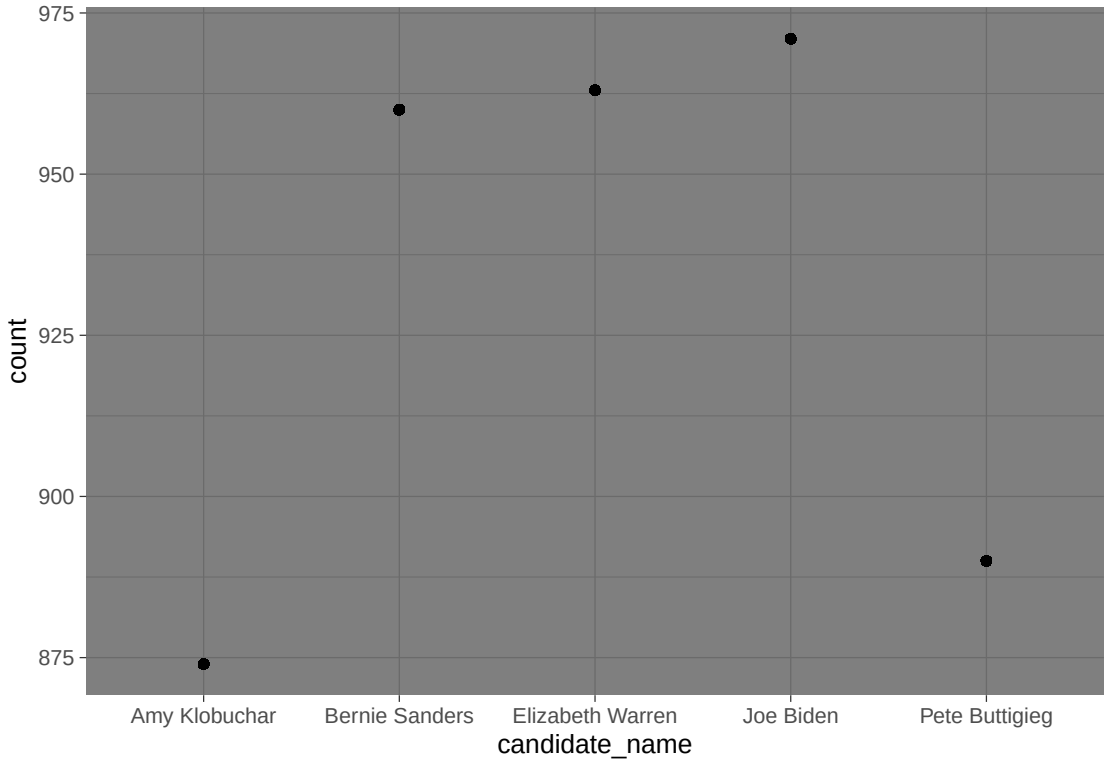
```
polls <- polls %>% group_by(endorser, candidate_name) %>% mutate(count = n())
```

- Plot the number of endorsement each of the 5 candidates have. Save your plot as an object `p`.

```
p <-  ggplot(data = polls)+
  geom_point(mapping =aes(
    x = candidate_name,
    y = count)) +
      theme_bw()
```

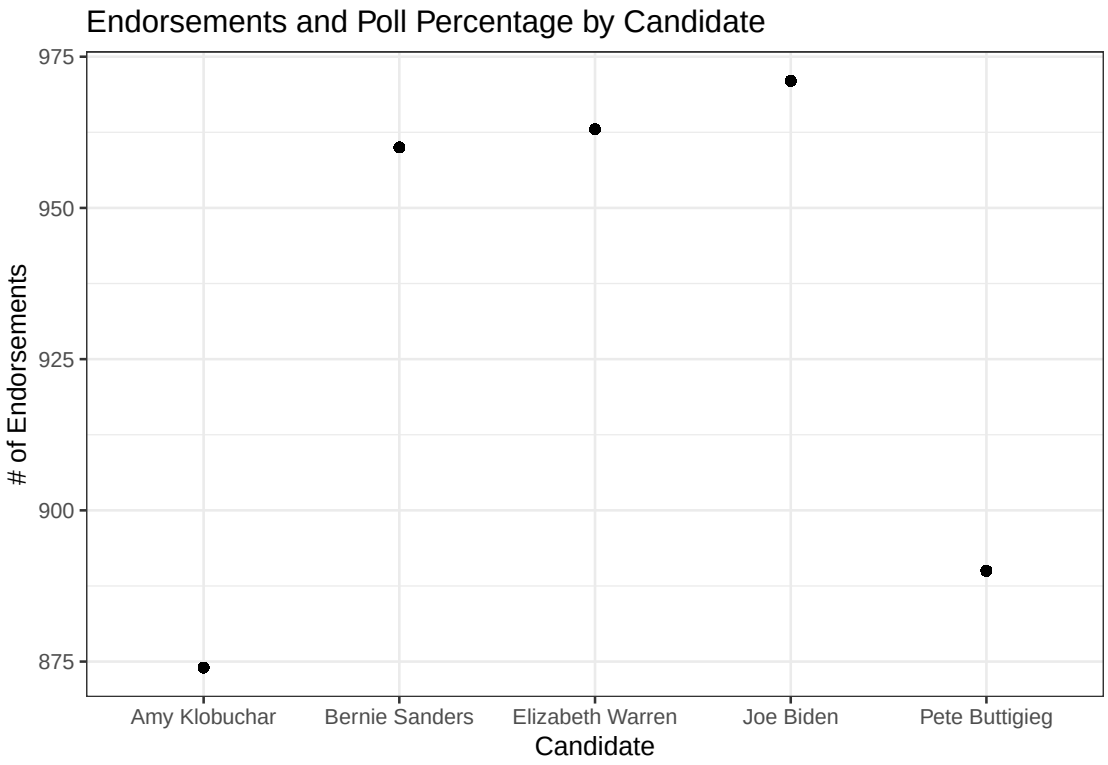
- Rerun the previous line as follows: \texttt{p + theme_dark()}. Notice how you can still customize your plot without rerunning the plot with new options.

```
p +theme_dark()
```



- 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.

```
p + ggtitle('Endorsements and Poll Percentage by Candidate') +
labs(y="# of Endorsements",
x= "Candidate")
```



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

2.
```
library(tidyverse)
#install.packages('tm')
library(tm)
```

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

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

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

```
#install.packages('lubridate')
library(lubridate)
```

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

```
## The following object is masked from 'package:base':
##
##     date
```

```
#install.packages('wordcloud')
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
tweets <- read_csv('https://politicaldatascience.com/PDS/Datasets/trump_tweets.csv')
```

```
## Parsed with column specification:
## cols(
##   source = col_character(),
##   text = col_character(),
##   created_at = col_character(),
##   retweet_count = col_double(),
##   favorite_count = col_double(),
##   is_retweet = col_logical()
## )
```

- First separate the `created_at` variable into two new variables where the date and the time are in separate columns. **Then** report the range of dates that is in this dataset.

```
tweets$created_at <- mdy_hm(tweets$created_at)
tweets <- separate(data = tweets, col = created_at, into  = c('Date', 'Time'), sep = ' ')
```

- 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.)

```
tweets <- tweets %>%
  filter(is_retweet==F)
tweets$text[match(sort(tweets$favorite_count, decreasing=TRUE)[1:5],tweets$favorite_count)]
```

[1] "A*APRockyreleasedfromprisonandonhiswayhometotheUnitedStatesfromSweden.ItwasaRockyWeekgethor
[2] "https://t.co/VXeKiVzpTf"
[3] "All is well! Missiles launched from Iran at two military bases located in Iraq. Assessment of casualties & damages taking place now. So far so good! We have the most powerful and well equipped military anywhere in the world by far! I will be making a statement tomorrow morning."
[4] "MERRY CHRISTMAS!"
[5] "Kobe Bryant despite being one of the truly great basketball players of all time was just getting started in life. He loved his family so much and had such strong passion for the future. The loss of his beautiful daughter Gianna makes this moment even more devastating. . . ."

```
tweets$text[match(sort(tweets$retweet_count, decreasing=TRUE)[1:5],tweets$retweet_count)]
```

[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 fat?" Oh well I try so hard to be his friend - and maybe someday that will happen!" [4] "A*APRockyreleasedfromprisonandonhiswayhometotheUnitedStatesfromSweden.ItwasaRockyWeekgethomeA
[5] "Such a beautiful and important evening! The forgotten man and woman will never be forgotten again. We will all come together as never before"

- 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.)

```
library(tm)
Corpus <- with(tweets, VCorpus(VectorSource(text)))
```

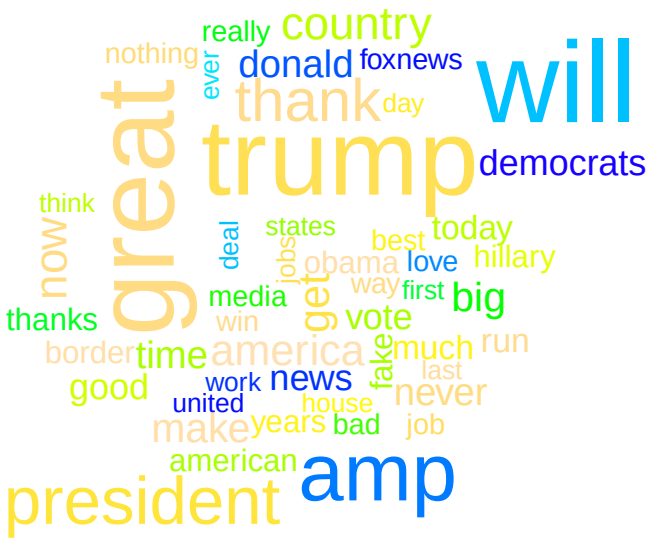
- Remove extraneous whitespace, remove numbers and punctuation, convert everything to lower case and a 'stop words' that have little substantive meaning (the, a, it).

```
Corpus <- Corpus %>%
  tm_map(stripWhitespace) %>%
  tm_map(removeNumbers) %>%
```

```
  tm_map(removePunctuation) %>%
  tm_map(content_transformer(tolower)) %>%
  tm_map(removeWords, c(stopwords("english"),"see", "people",'new','want','one','even','must',
```

- 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.

```
wordcloud(Corpus, max.words = 50, colors = topo.colors(n = 50),
random.color = TRUE, random.order =T, min.freq =3)
```



- Create a *document term matrix* called `DTM` that includes the argument `control = list(weighting = weightTfIdf)`

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

- Finally, report the 50 words with the the highest tf.idf scores using a lower frequency bound of .8.

```
findFreqTerms(DTM, lowfreq = 0.8)[1:50]
```

\texttt{ [1] “–clewandowski” “–cuomo”

[3] “–donald” “–tw”

[5] “—amazing” “—bad”

[7] “—big” “—conduct”

[9] “—crazy” “—donald”

[11] “—everybody” “—gt”

[13] “—hit” “—impeachmentevery”

[15] “—including” “—loser”

[17] “—mexico” “—pablo”

[19] “—political” “—president”

[21] “—remember” “—total”

[23] “—vote. . . ” “—wonder”

[25] “—worst” “¡latinos”

[27] “‘’d” “‘’m”

[29] “‘’re” “‘’s”

[31] “‘’t” “‘’ve”

[33] “‘abuse” “ ‘amnesty’ ”

[35] “‘angel” “ ‘antibush’ ”

[37] “‘bad” “‘big”

[39] “‘bill” “‘blew”

[41] “ ‘boring’ ” “‘bring”

[43] “ ‘caravan’ ” “ ‘cataclysmic’ ”

[45] “‘climate” “ ‘climate’ ”

[47] “‘clinton” “‘close”

[49] “‘completely” “‘crime” }