Problem Set 2: Data Wrangling

Grace Fragakis

Background

Political advertising has traditionally been focused on the medium of television, but in recent cycles, online advertising has become much more popular. In this problem set, you will explore a dataset that has information on Facebook ad spending and impressions by candidates in the 2018 election cycle in the United States. The variables in this data are described below.

Name	Description
`cand_id`	unique identifier code for candidate
`cand_name`	full name of the candidate
`cand_name_last`	last name of the candidate
`party`	party affiliation of the candidate (R = Republican, D = Democrat)
`office`	office being sought by candidate
`state`	state in which the candidate is running
`incumbency`	incumbency status of candidate (incumbent, challenger, or open seat)
`spend`	estimated total spending on Facebook ads by candidate
`impressions`	estimated total impressions of Facebook ads
`ad_tone_attack`	proportion of FB ads that mention candidate's opponent only
`ad_tone_promote`	proportion of FB ads that mention candidate only
`ad_tone_contrast`	proportion of FB ads that mention candidate and candidate's opponent

Question 1 (8 points)

Load the data using the `read_csv` function and save it as `fb_ads` (using this will automatically make `fb_ads` a tibble). In the text, describe how many candidates there are in the dataset.

Use `dplyr` functions to create a table with the number of candidates in each type of incumbency status in the data set. Save this table output as `incumbency_table` (for the autograder). Use the function `knitr::kable()` on this table to have a nicely formatted table produced in the knitted output.

Rubric: 2pt for loading the data (autograder); 1pt for describing the number of candidates (PDF); 3pts for creating the table (autograder); 2pt for using `kable()` to nicely format the output (PDF)

Answer 1

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

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

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.5
## v ggplot2   3.5.1     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1

## -- Conflicts ------------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
fb_ads <- read_csv('data/fb_ads.csv')
```

```
## Rows: 7014 Columns: 12
## -- Column specification --------------------------------------------------------------------------
## Delimiter: ","
## chr (7): cand_id, cand_name, cand_name_last, party, office, state, incumbency
## dbl (5): spend, impressions, ad_tone_attack, ad_tone_promote, ad_tone_contrast
##
## 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
incumbency_table <-
  fb_ads |>
  count(incumbency)

knitr::kable(incumbency_table, col.names = c("Incumbency Status", "Number of Candidates"))
```

Incumbency Status	Number of Candidates
Challenger	2510
Incumbent	2022
Open Seat	2482

Question 2 (7 points)

Filter the data to just US House and US Senate races and use this to create a tibble called `party_incumbent_promote` that has 6 rows that summarizes the average of `ad_tone_promote` for each combination of `party` and `incumbency`. Call the variable summarizing the promote variable as `promote_prop` and be sure to remove any missing values when computing the averages.

Use `knitr::kable()` to produce a nicely formatted table. In this call, set the `digits` arguments to 3 and use the `col.names` argument to pass a nicer set of names. You can use the following as a template:

```
filter_q2 <-
  fb_ads |>
  filter(office %in% c('US House', 'US Senate'))

party_incumbent_promote <-
  filter_q2 |>
  group_by(party, incumbency) |>
  summarise(promote_prop = mean(ad_tone_promote, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'party'. You can override using the `.groups` argument.
```

```
knitr::kable(party_incumbent_promote, col.names = c("Party", "Incumbency Status", 'Promote Average'))
```

Party	Incumbency Status	Promote Average
D	Challenger	0.8332344
D	Incumbent	0.8544311
D	Open Seat	0.8454562
R	Challenger	0.8132945
R	Incumbent	0.7923189
R	Open Seat	0.8281463

In the writeup, describe which type of candidate sponsored the most promoting ads on average.

Rubric: 3pts for creating `party_incumbent_promote` correctly (autograder); 2pt for a nicely formatted table (PDF); 1pt for changing the column names of the output table (PDF); 1pt for correctly identifying the type of candidate with highest average (PDF)

Answer 2

The candidate that sponsered the the most promoting ads on average was the Democratic Incumbent.

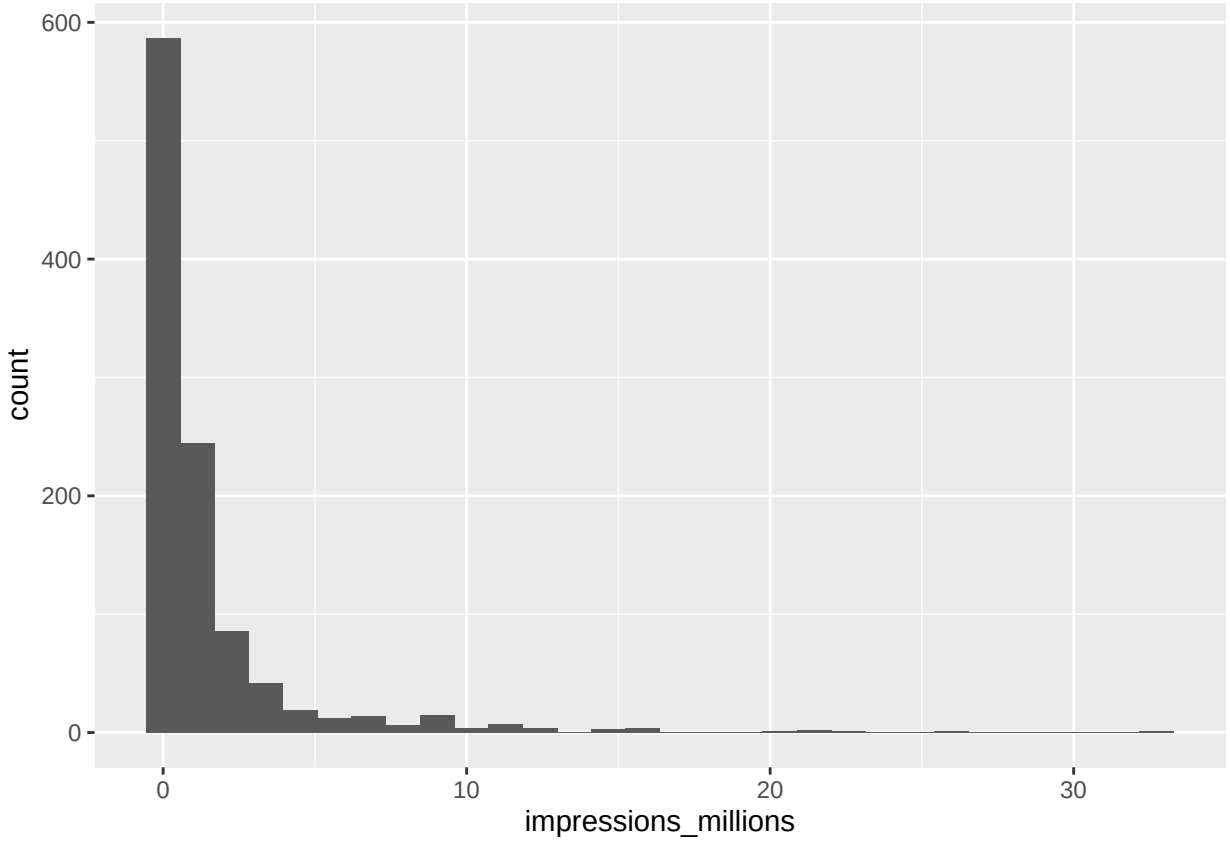
Question 3 (7 points)

Create a new variable called `impressions_millions` that is the total Facebook ad impressions in millions (as opposed to single impressions). Make sure to save the resulting dataset back as `fb_ads`.

Create a histogram of this variable for just the US House races. Save the ggplot output as `plot_q3` and also print it to produce a plot in the output. In the text, describe the shape of the histogram and tell the reader if most of the House candidates had more than 10 million ads impressions on Facebook.

```
library(ggplot2)

fb_ads <- fb_ads %>%
  mutate(impressions_millions = impressions / 1000000)

house_filter <-
  fb_ads |>
  filter(office == 'US House')

plot_q3 <-
  ggplot(house_filter, aes(x=impressions_millions)) +
  geom_histogram()

plot_q3
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Rubric: 2pt for creating the new variable (autograder); 3pts for creating the histogram object (au

Answer 3

The histogram is shaped like a downward sloping curve. It appears that most of the House candidates did

\newpage

Question 4 (13 points)

Let's now recreate the following plot that shows the top 15 House candidates in terms of Facebook ad imp

You should save the `ggplot` output as `fb_top_plot`. You should also write `fb_top_plot` on its own lin

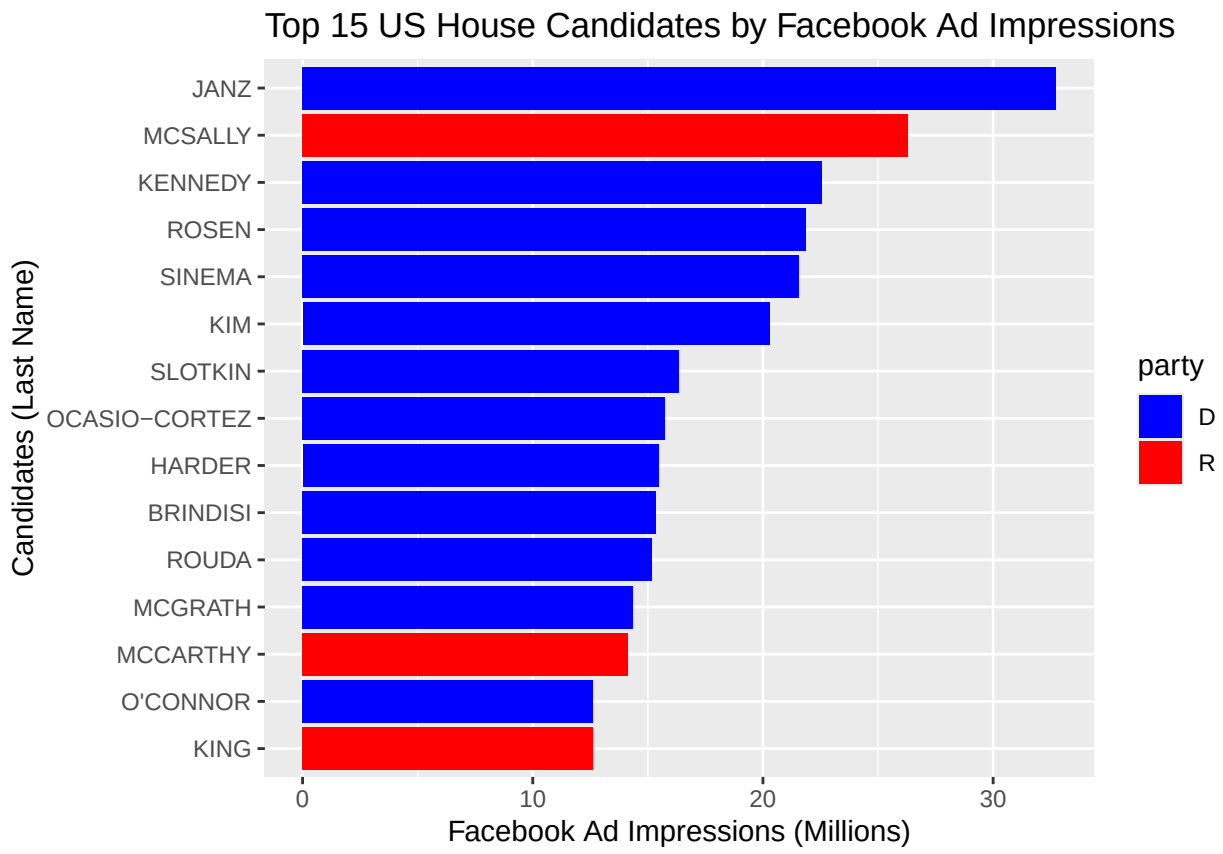
- The barplot should have candidate last names on the y-axis and the `impressions_millions` variable fr
- The data feeding into the `ggplot` call should only have US House candidates and only the candidates
- The y-axis should be ordered in ascending values of `impressions_millions` so that the lowest values a
- The fill color of the bar plot should be mapped to the `party` variable (but not globally!).

You do not need to exactly match the labels, but you should have informative labels. The color does not

Rubric: 3pts for correct axes (autograder); 3pts for correct data fed into `ggplot` (autograder); 3

Answer 4

```r
t_15_candidates <-
  house_filter |>
  arrange(desc(impressions_millions)) |>
  head(n = 15) |>
  mutate(cand_name_last = factor(cand_name_last, levels = rev(cand_name_last)))

fb_top_plot <-
  ggplot(t_15_candidates, aes(x=impressions_millions, y = cand_name_last)) +
  geom_bar(aes(fill = party), stat = 'identity') +
  scale_fill_manual(values = c(R = "red", D = "blue")) +
  labs(x = "Facebook Ad Impressions (Millions)",
       y = "Candidates (Last Name)",
       title= "Top 15 US House Candidates by Facebook Ad Impressions")

fb_top_plot
```


Top 15 US House Candidates by Facebook Ad Impressions
JANZ
MCSALLY
KENNEDY
ROSEN
SINEMA
KIM
SLOTKIN
OCASIO−CORTEZ
HARDER
BRINDISI
ROUDA
MCGRATH
MCCARTHY
O'CONNOR
KING
Candidates (Last Name)
0
10
20
30
Facebook Ad Impressions (Millions)
party
D
R

Code

```
options(width = 100)
library(dplyr)
library(tidyverse)

fb_ads <- read_csv('data/fb_ads.csv')

incumbency_table <-
  fb_ads |>
  count(incumbency)

knitr::kable(incumbency_table, col.names = c("Incumbency Status", "Number of Candidates"))


filter_q2 <-
  fb_ads |>
  filter(office %in% c('US House', 'US Senate'))

party_incumbent_promote <-
  filter_q2 |>
  group_by(party, incumbency) |>
  summarise(promote_prop = mean(ad_tone_promote, na.rm = TRUE))

knitr::kable(party_incumbent_promote, col.names = c("Party", "Incumbency Status", 'Promote Average'))
library(ggplot2)

fb_ads <- fb_ads %>%
  mutate(impressions_millions = impressions / 1000000)

house_filter <-
  fb_ads |>
  filter(office == 'US House')

plot_q3 <-
  ggplot(house_filter, aes(x=impressions_millions)) +
  geom_histogram()

plot_q3


t_15_candidates <-
  house_filter |>
  arrange(desc(impressions_millions)) |>
  head(n = 15) |>
  mutate(cand_name_last = factor(cand_name_last, levels = rev(cand_name_last)))

fb_top_plot <-
  ggplot(t_15_candidates, aes(x=impressions_millions, y = cand_name_last)) +
  geom_bar(aes(fill = party), stat = 'identity') +
  scale_fill_manual(values = c(R = "red", D = "blue")) +
  labs(x = "Facebook Ad Impressions (Millions)",
       y = "Candidates (Last Name)",
```

```
       title= "Top 15 US House Candidates by Facebook Ad Impressions")

fb_top_plot
```