Problem Set 5

Multivariate Visualization

[YOUR NAME]

Due Date: 2024-02-16

Getting Set Up

Open `RStudio` and create a new RMarkDown file (`.Rmd`) by going to `File -> New File -> R Markdown...` . Accept defaults and save this file as `[LAST NAME]_ps5.Rmd` to your `code` folder.

Copy and paste the contents of this `.Rmd` file into your `[LAST NAME]_ps5.Rmd` file. Then change the `author: [Your Name]` to your name.

We will be using the `Pres2020_PV.Rds` file from the course github page (https://github.com/jbisbee1/DS1000_S2024/blob/main/data/Pres2020_PV.Rds).

All of the following questions should be answered in this `.Rmd` file. There are code chunks with incomplete code that need to be filled in.

This problem set is worth 8 total points, plus two extra credit points. The point values for each question are indicated in brackets below. To receive full credit, you must have the correct code. In addition, some questions ask you to provide a written response in addition to the code.

You are free to rely on whatever resources you need to complete this problem set, including lecture notes, lecture presentations, Google, your classmates…you name it. However, the final submission must be complete by you. There are no group assignments. To submit, compiled the completed problem set and upload the PDF file to Brightspace on Friday by midnight. Also note that the TAs and professors will not respond to Campuswire posts after 5PM on Friday, so don't wait until the last minute to get started!

Good luck!

*Copy the link to ChatGPT you used here: _____

```
require(tidyverse)
```

```
## Loading required package: tidyverse
```

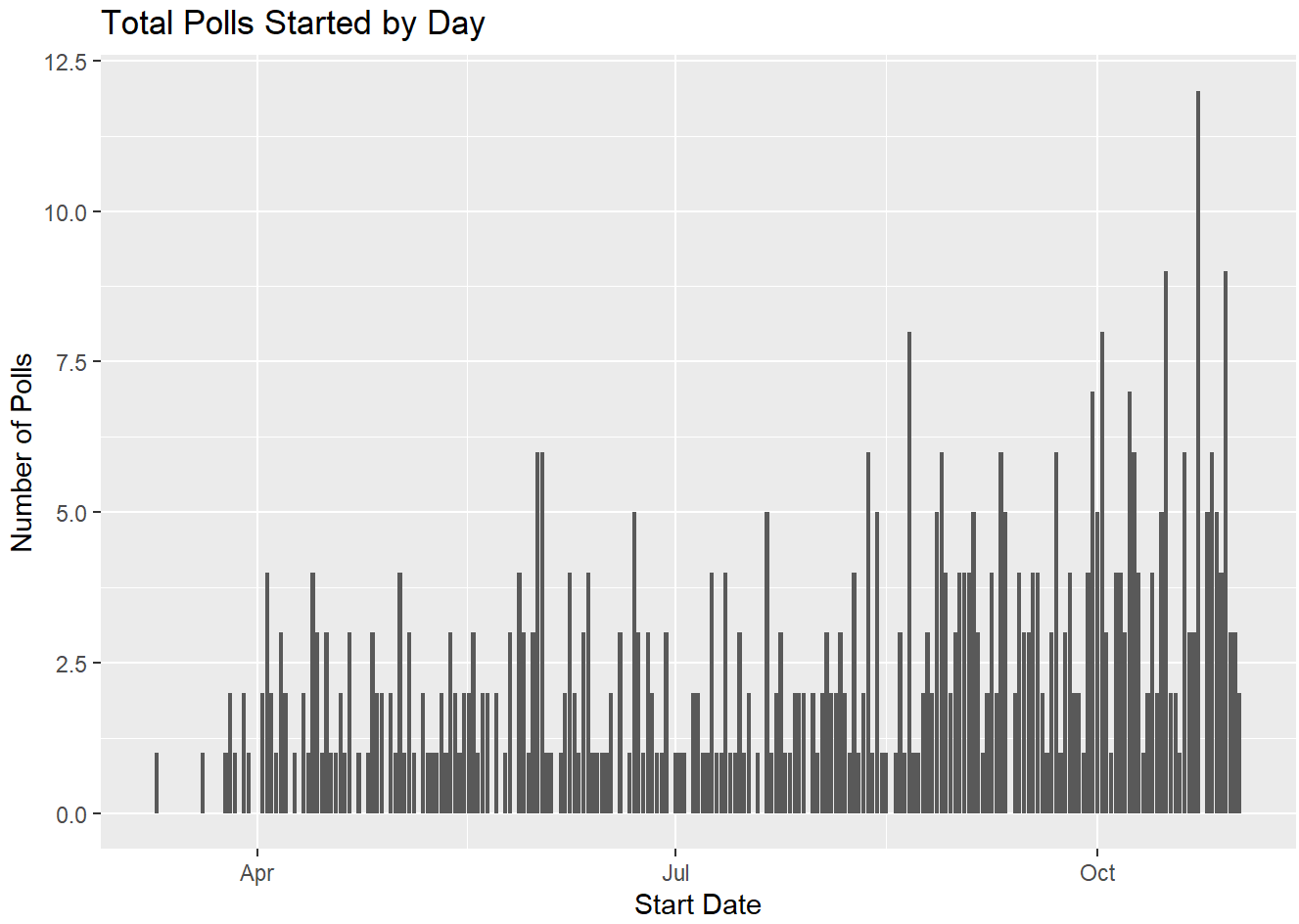
```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.2     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.2     ✓ tibble    3.2.1
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ✓ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
pres <- read_rds('https://github.com/jbisbee1/DS1000_S2024/raw/main/data/Pres2020_PV.Rds')
```

Question 1 [2 points]

Plot the total number of polls per start date in the data. NB: you will have convert `StartDate` to a `date` class with `as.Date()`. If you need help, see this post (https://www.r-bloggers.com/2013/08/date-formats-in-r/). Do you observe a noteworthy trend in the number of polls over time?

```
pres %>%
  mutate(StartDate = as.Date(StartDate,'%m/%d/%Y')) %>%
  ggplot(aes(x = StartDate)) +
  geom_bar(stat = 'count') +
  labs(title = 'Total Polls Started by Day',
       x = 'Start Date',
       y = 'Number of Polls')
```



- There are more polls fielded the closer we get to the election.

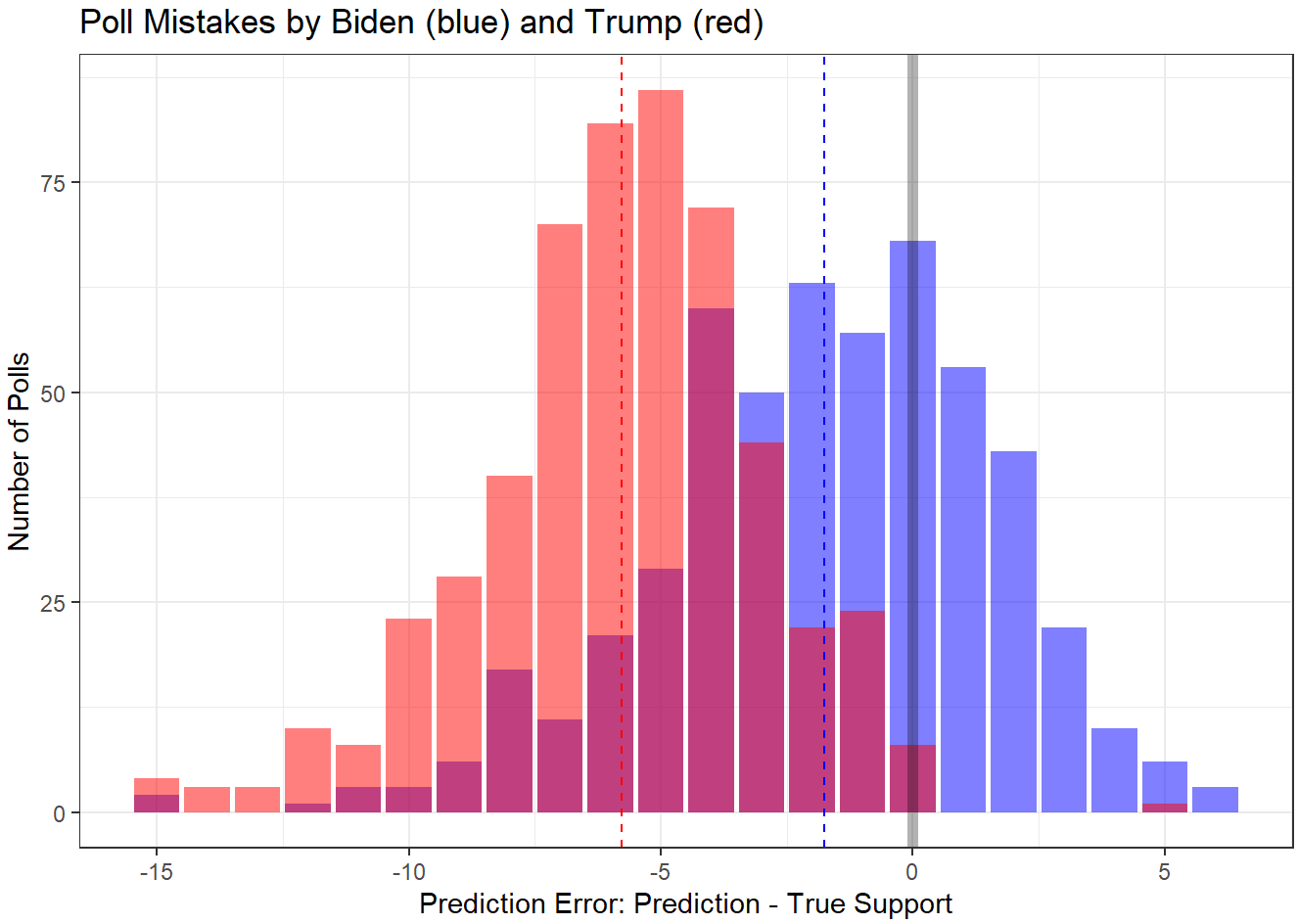
Question 2 [2 points]

Calculate the **prediction error** for Biden and Trump such that positive values mean that the poll *overestimated* the candidate's popular vote share (`DemCertVote` for Biden and `RepCertVote` for Trump). Plot the Biden and Trump prediction errors on a single plot using `geom_bar()` , with red indicating Trump and blue indicating Biden (make sure to set alpha to some value less than 1 to increase the transparency!). Add vertical lines for the average prediction error for both candidates (colored appropriately) as well as a vertical line indicating no prediction error. **HINT**: create a new object called `toplot` which adds the prediction error columns to `pres` via `mutate()` .

Do you observe a systematic bias toward one candidate or the other?

```
toplot <- pres %>%
  mutate(demErr = Biden - DemCertVote,
         repErr = Trump - RepCertVote)

toplot %>%
  ggplot() +
  geom_bar(aes(x = demErr),fill = 'blue',alpha = .5) +
  geom_bar(aes(x = repErr),fill = 'red',alpha = .5) +
  labs(title = 'Poll Mistakes by Biden (blue) and Trump (red)',
       x = 'Prediction Error: Prediction - True Support',
       y = 'Number of Polls') +
  theme_bw() +
  geom_vline(xintercept = 0,lwd = 2,alpha = .3) +
  geom_vline(xintercept = mean(toplot$demErr,na.rm=T),color = 'blue',linetype = 'dashed') +
  geom_vline(xintercept = mean(toplot$repErr,na.rm=T),color = 'red',linetype = 'dashed')
```



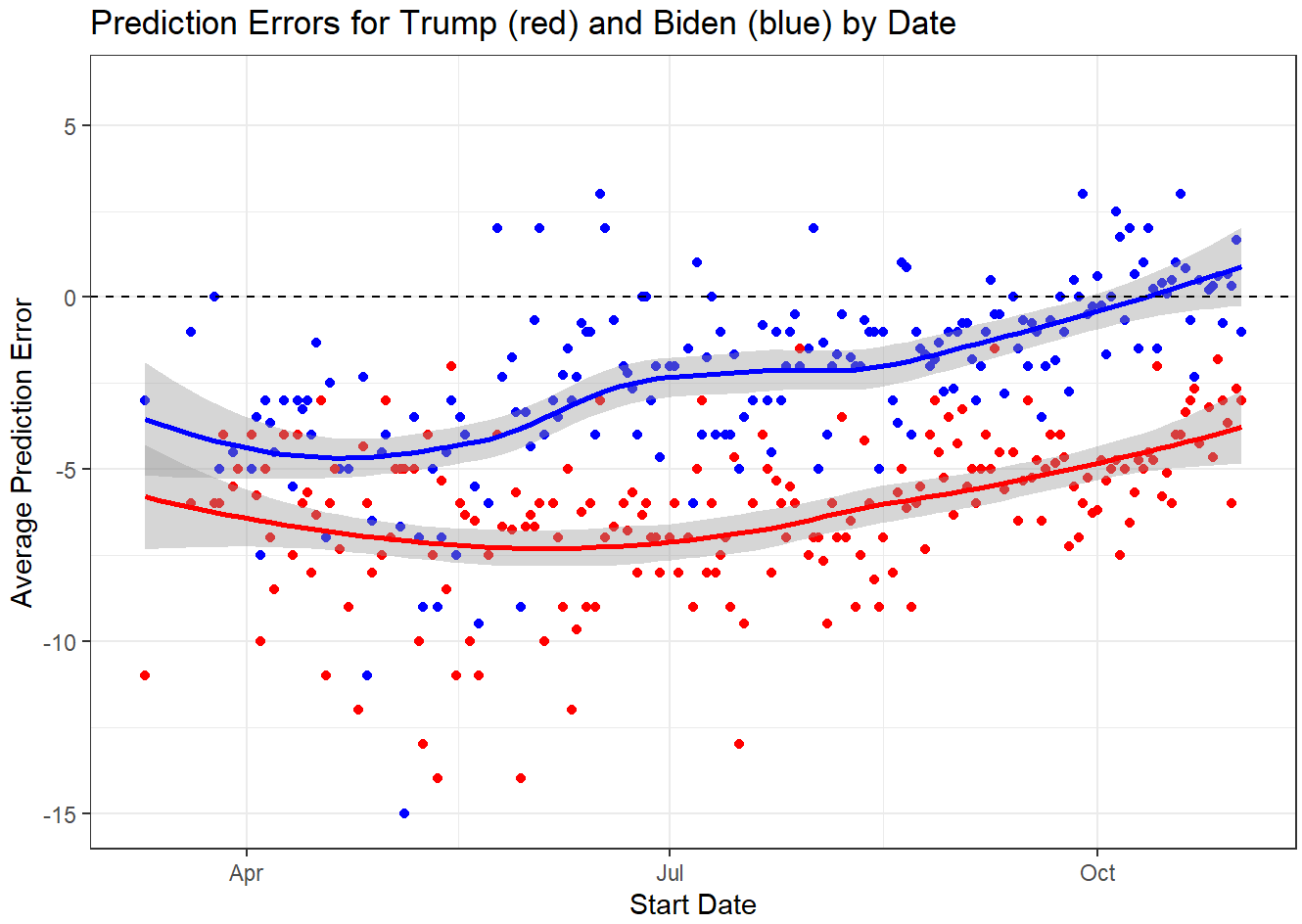
- I observe a systematic bias against both candidates where the polls underestimate the amount of support for Biden and Trump. However, the magnitude of this bias against Trump is larger than the bias against Biden.

Question 3 [2 points]

Plot the average prediction error for Trump (red) and Biden (blue) by start date using `geom_point()` and `geom_smooth()`. What pattern do you observe over time, if any?

```
toplot %>%
  mutate(StartDate = as.Date(StartDate,'%m/%d/%Y')) %>%
  group_by(StartDate) %>%
  summarise(demErr = mean(demErr),
            repErr = mean(repErr)) %>%
  ggplot() +
  geom_point(aes(x = StartDate,y = demErr),color = 'blue') +
  geom_point(aes(x = StartDate,y = repErr),color = 'red') +
  geom_smooth(aes(x = StartDate,y = demErr),color = 'blue') +
  geom_smooth(aes(x = StartDate,y = repErr),color = 'red') +
  labs(title = "Prediction Errors for Trump (red) and Biden (blue) by Date",
       x = "Start Date",
       y = "Average Prediction Error") +
  geom_vline(xintercept = 0,linetype = 'dashed') +
  geom_hline(yintercept = 0,linetype = 'dashed') +
  theme_bw() +
  scale_y_continuous(limits = c(min(c(toplot$demErr,toplot$repErr),na.rm=T),
                                max(c(toplot$demErr,toplot$repErr),na.rm=T)))
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

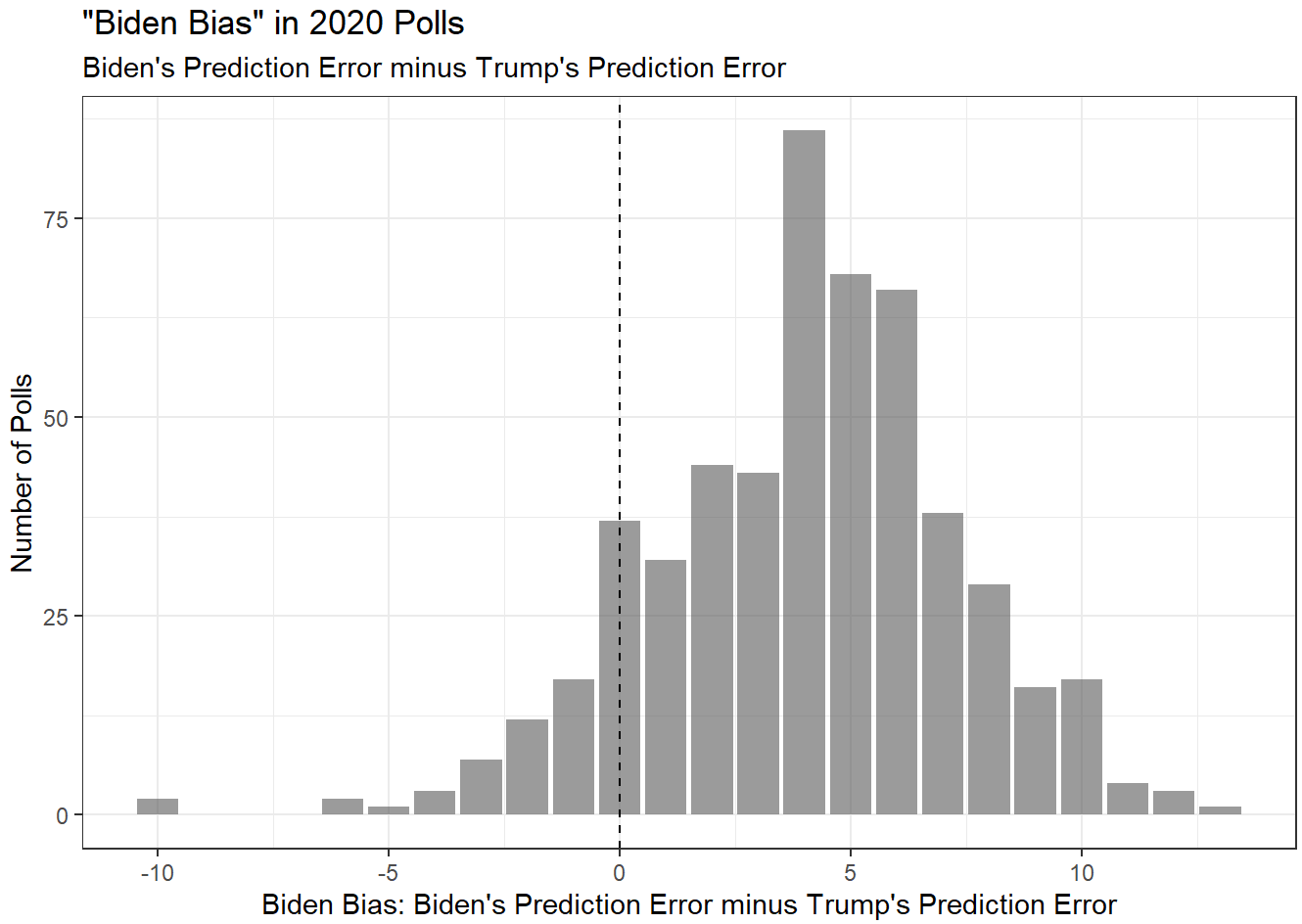


- I observe a gradual decline in the prediction error over time, where polls underestimate both Trump and Biden less and less.

Question 4 [2 points]

Calculate each poll's bias toward Biden (this should be the prediction error for Biden minus the prediction error for Trump) and plot the distribution. What proportion of polls' prediction error favored Biden over Trump? What does this mean about polling in the United States?

```
toplot %>%
  mutate(bidenBias = demErr - repErr) %>%
  ggplot(aes(x = bidenBias)) +
  geom_bar(alpha = .6) +
  labs(title = '"Biden Bias" in 2020 Polls',
       subtitle = "Biden's Prediction Error minus Trump's Prediction Error",
       x = "Biden Bias: Biden's Prediction Error minus Trump's Prediction Error",
       y = 'Number of Polls') +
  geom_vline(xintercept = 0,linetype = 'dashed') +
  theme_bw()
```




```
toplot %>%
  mutate(bidenBias = demErr - repErr) %>%
  summarise(mean(bidenBias > 0))
```

```
## # A tibble: 1 × 1
##   `mean(bidenBias > 0)`
##                   <dbl>
## 1                 0.847
```

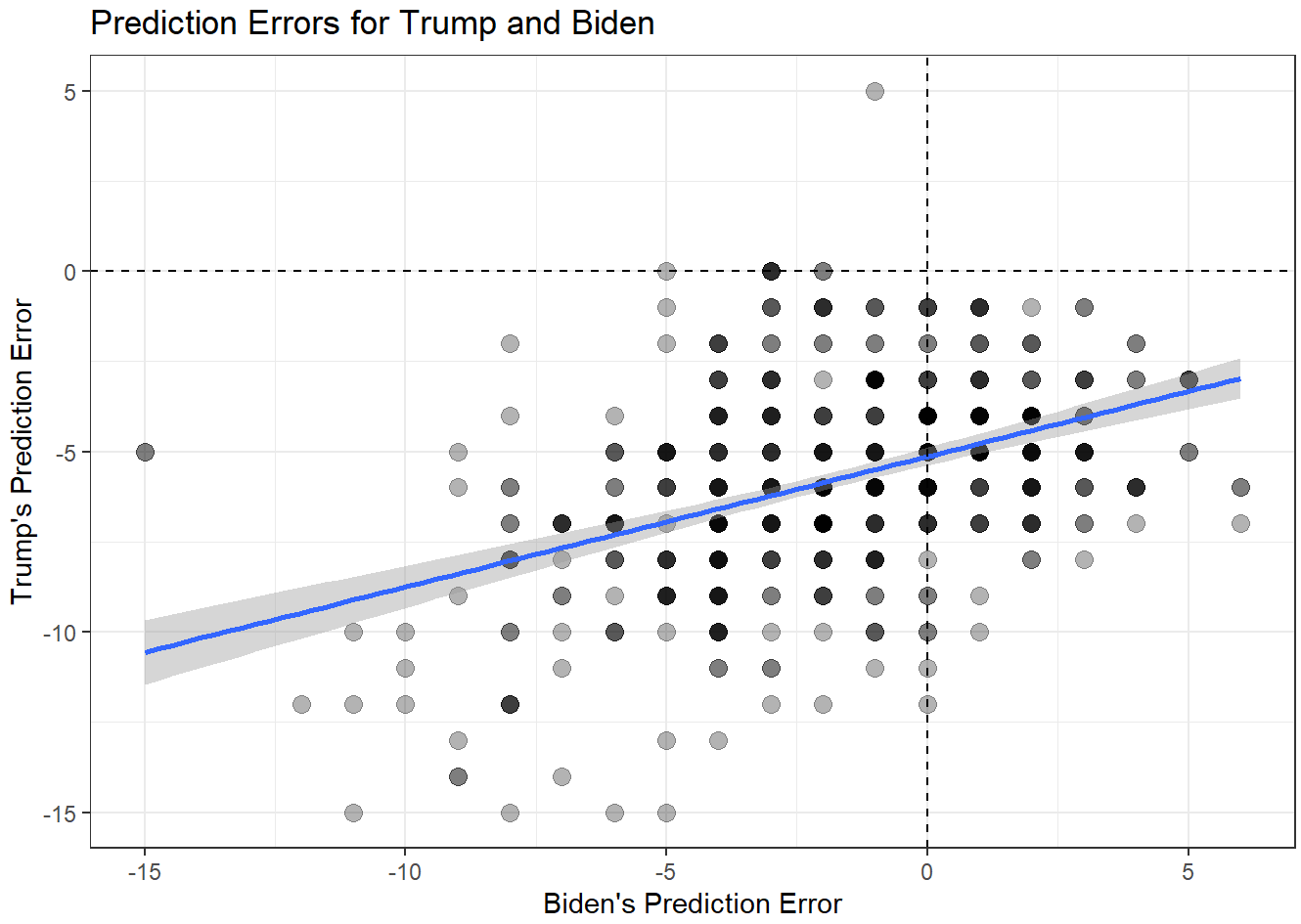
- 84.7% of polls had prediction errors that favored Biden over Trump.

Extra Credit [2 points]

Do polls that underestimate Trump's support overestimate Biden's support? Use a scatterplot to test, combined with a line of best fit. Then, calculate the proportion of polls that (1) underestimate both Trump and Biden, (2) underestimate Trump and overestimate Biden, (3) overestimate Trump and underestimate Biden, (4) overestimate both candidates. In these analyses, define "overestimate" as prediction errors greater than or equal to zero, whereas "underestimate" should be prediction errors less than zero.

```
toplot %>%
  ggplot(aes(x = demErr,y = repErr)) +
  geom_point(size = 3,alpha = .3) +
  geom_smooth(method = 'lm') +
  labs(title = "Prediction Errors for Trump and Biden",
       x = "Biden's Prediction Error",
       y = "Trump's Prediction Error") +
  geom_vline(xintercept = 0,linetype = 'dashed') +
  geom_hline(yintercept = 0,linetype = 'dashed') +
  theme_bw()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

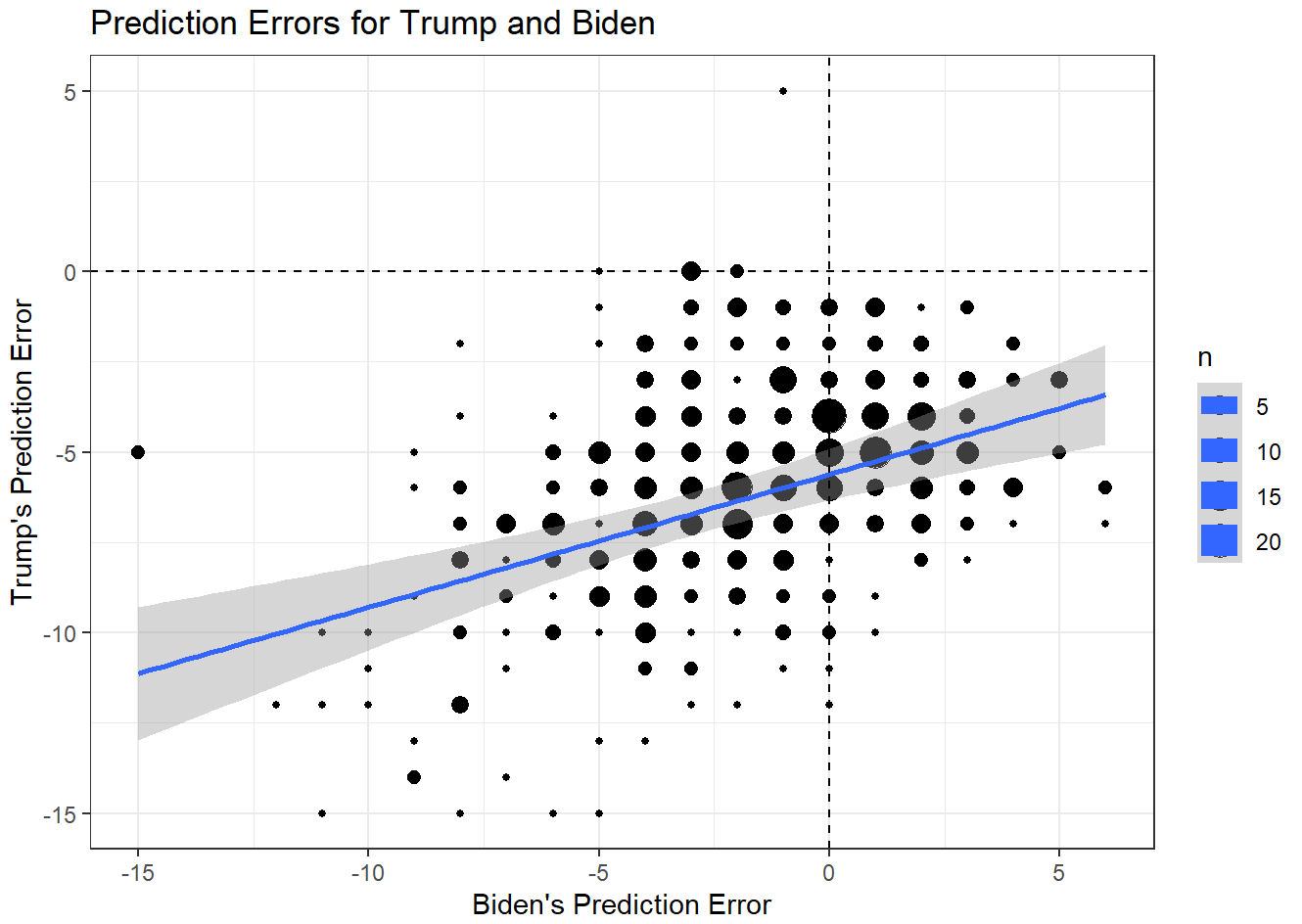


```
# Alternative #1
toplot %>%
  count(demErr,repErr) %>%
  ggplot(aes(x = demErr,y = repErr,size = n)) +
  geom_point() +
  geom_smooth(method = 'lm') +
  labs(title = "Prediction Errors for Trump and Biden",
       x = "Biden's Prediction Error",
       y = "Trump's Prediction Error") +
  geom_vline(xintercept = 0,linetype = 'dashed') +
  geom_hline(yintercept = 0,linetype = 'dashed') +
  theme_bw()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

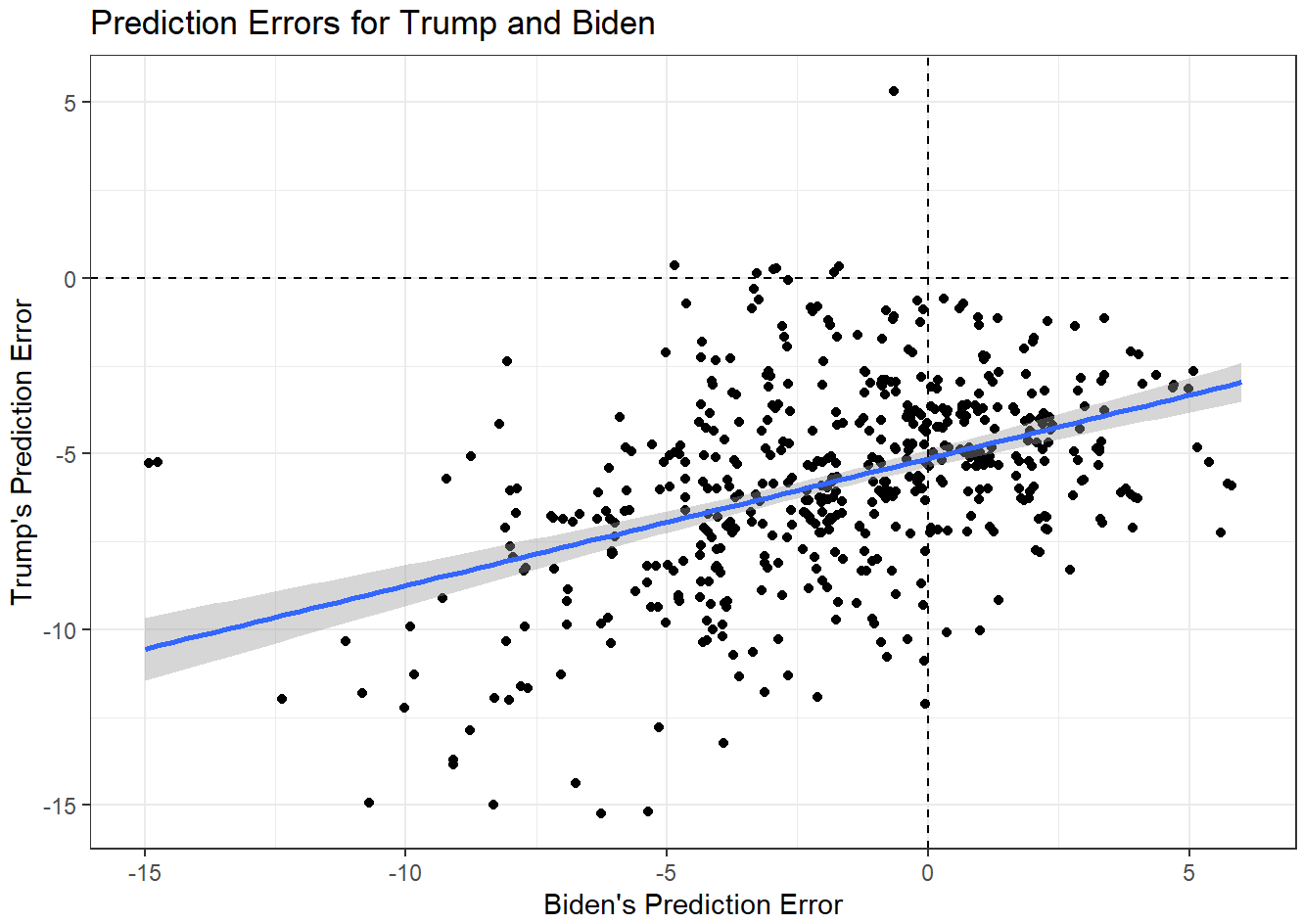
```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: The following aesthetics were dropped during statistical transformation: size
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```




```
# Alternative #2
toplot %>%
  ggplot(aes(x = demErr,y = repErr)) +
  geom_jitter() +
  geom_smooth(method = 'lm') +
  labs(title = "Prediction Errors for Trump and Biden",
       x = "Biden's Prediction Error",
       y = "Trump's Prediction Error") +
  geom_vline(xintercept = 0,linetype = 'dashed') +
  geom_hline(yintercept = 0,linetype = 'dashed') +
  theme_bw()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
toplot %>%
  summarise(UNboth = mean(demErr < 0 & repErr < 0),
            UNTrOVBi = mean(demErr >= 0 & repErr < 0),
            OVTrUNBi = mean(demErr < 0 & repErr >= 0),
            OVboth = mean(demErr >= 0 & repErr >= 0))
```

```
## # A tibble: 1 × 4
##   UNboth UNTrOVBi OVTrUNBi OVboth
##    <dbl>    <dbl>    <dbl>  <dbl>
## 1  0.595    0.388   0.0170      0
```