Problem Set 3

Data Wrangling

[YOUR NAME]

Due Date: 2024-07-12

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 `[YOUR NAME]_ps3.Rmd` to your `code` folder.

Copy and paste the contents of this `.Rmd` file into your `[YOUR NAME]_ps3.Rmd` file. Then change the `author: [Your Name]` on line 2 to your name.

We will be using two different files. First is the `Pres2020_PV.Rds` data from the course github page (https://github.com/jbisbee1/ISP_Data_Science_2024/blob/main/data/Pres2020_PV.Rds). Second is the `game_summary.Rds` data, which is also on the course github page (https://github.com/jbisbee1/ISP_Data_Science_2024/blob/main/data/game_summary.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 10 total points, plus **two** extra credit questions, each worth **two** 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, email the knitted output to Eun Ji Kim (kej990804@snu.ac.kr (mailto:kej990804@snu.ac.kr)) **as a PDF** by the start of class on Friday, July 12th. If you need help converting to a PDF, see this tutorial (https://github.com/jbisbee1/ISP_Data_Science_2024/blob/main/Psets/ISP_pset_0_HELPER.pdf).

Good luck!

*Copy the link to ChatGPT you used here: _________________

Question 0

Require `tidyverse` and load the `Pres2020_PV.Rds` data to an object called `pres`. ALSO load a new package called `labelled`, which will allow us to read the labels for our variables. Remember, if you don't have this package yet, you need to use `install.packages("labelled")` in the `Console` window.

```
require(tidyverse)
```

```
## Loading required package: tidyverse
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
## ── 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 t
o become errors
```

```
require(labelled)
```

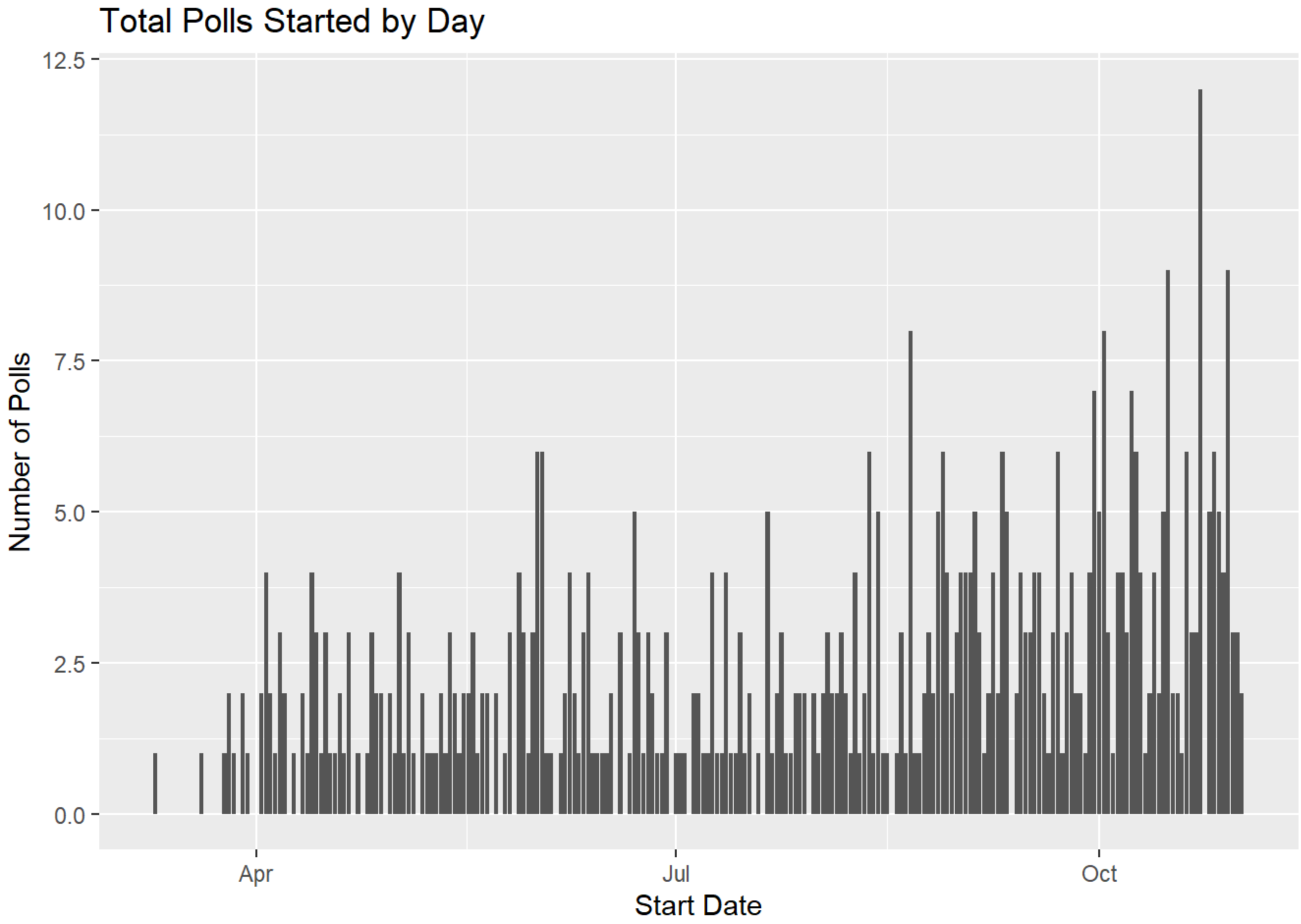
```
## Loading required package: labelled
```

```
pres <- read_rds("https://github.com/jbisbee1/ISP_Data_Science_2024/raw/main/data/Pres20
20_PV.Rds")
```

Question 1 [1 point]

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 pattern in the number of polls over time? Why do you think this is?

```
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. This is probably because there is more demand for information about the election the closer the election is, and media outlets want to make more money by selling polling data.

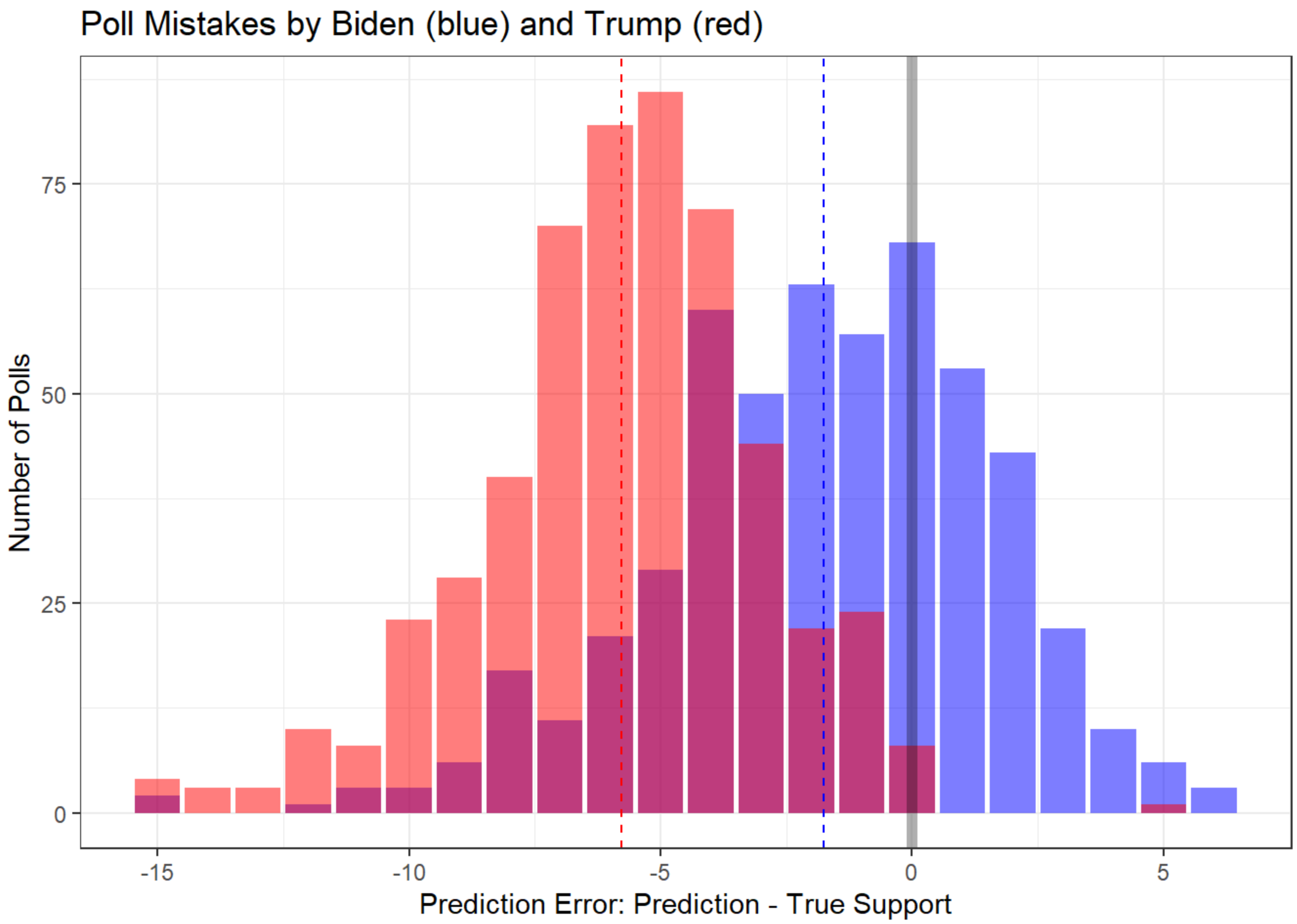
Question 2 [1 point]

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$demErr,na.rm=T),color = 'blue',linetype = 'dashe
d') +
  geom_vline(xintercept = mean(toplot$repErr,na.rm=T),color = 'red',linetype = 'dashed')
```



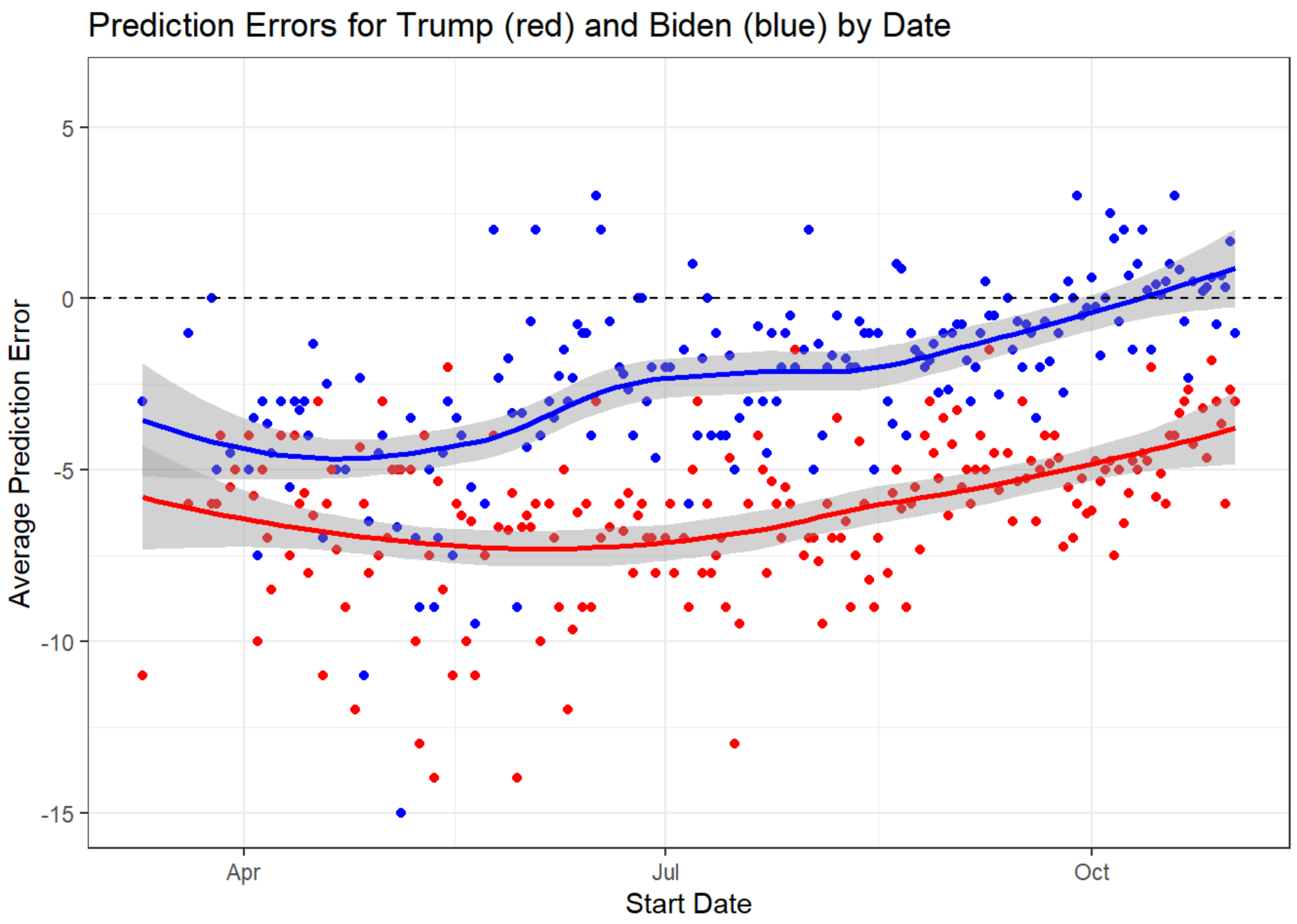
I observe a systematic bias against both candidates where the polls underestimate the amount of support for Biden and Trump. However, the magnitude of this bias against Trump is larger than the bias against Biden.

Question 3 [1 point]

Plot the average prediction error for Trump (red) and Biden (blue) by start date using `geom_point()` *and add a curvey line of best fit using* `geom_smooth()` *(allow it to be curved!). 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'
```

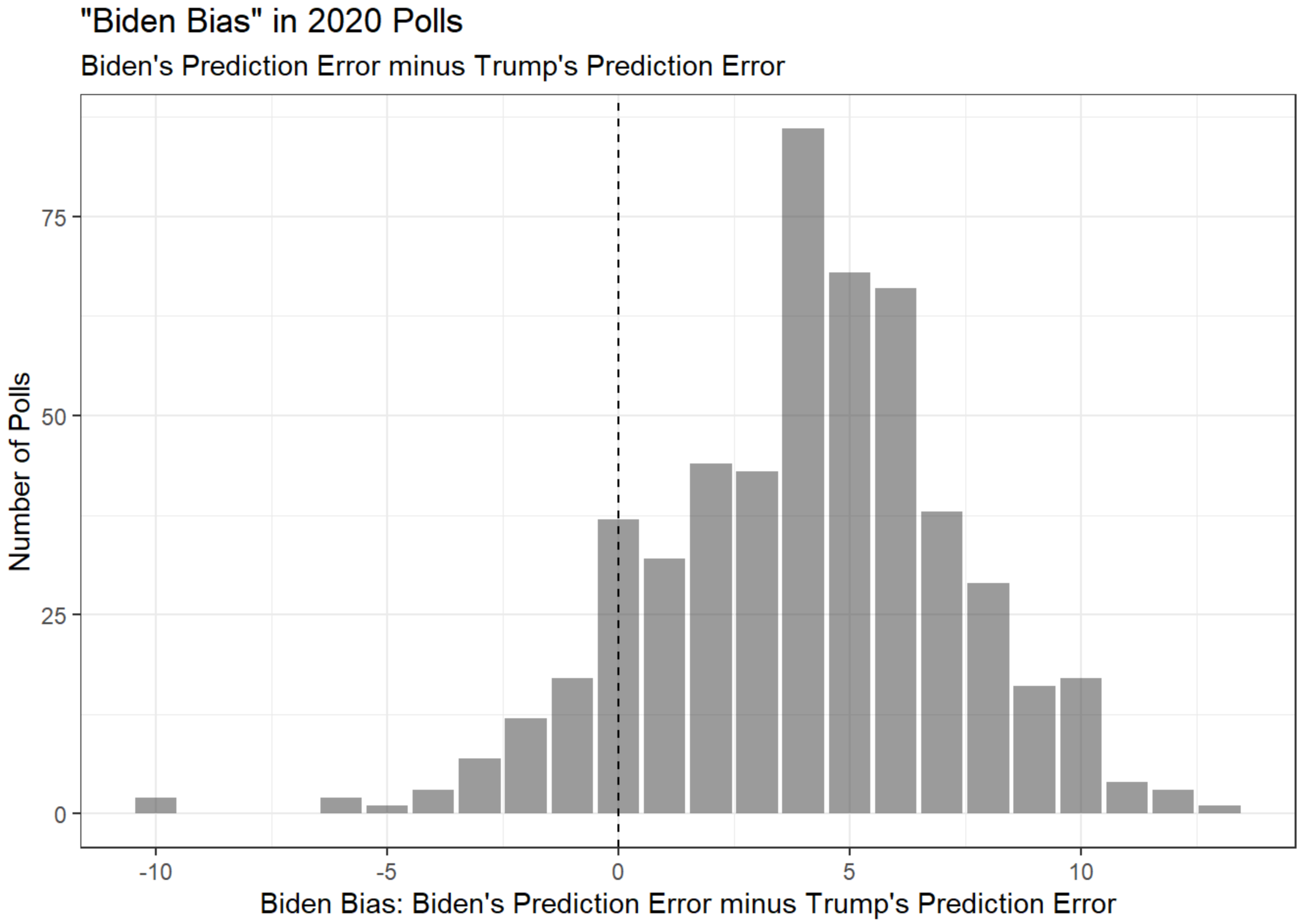


> I observe a gradual decline in the prediction error over time, where polls underestimate both Trump and Biden less and less. However, polls still underestimated Trump by the time of the election, whereas they perfectly predicted Biden's support.

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
```

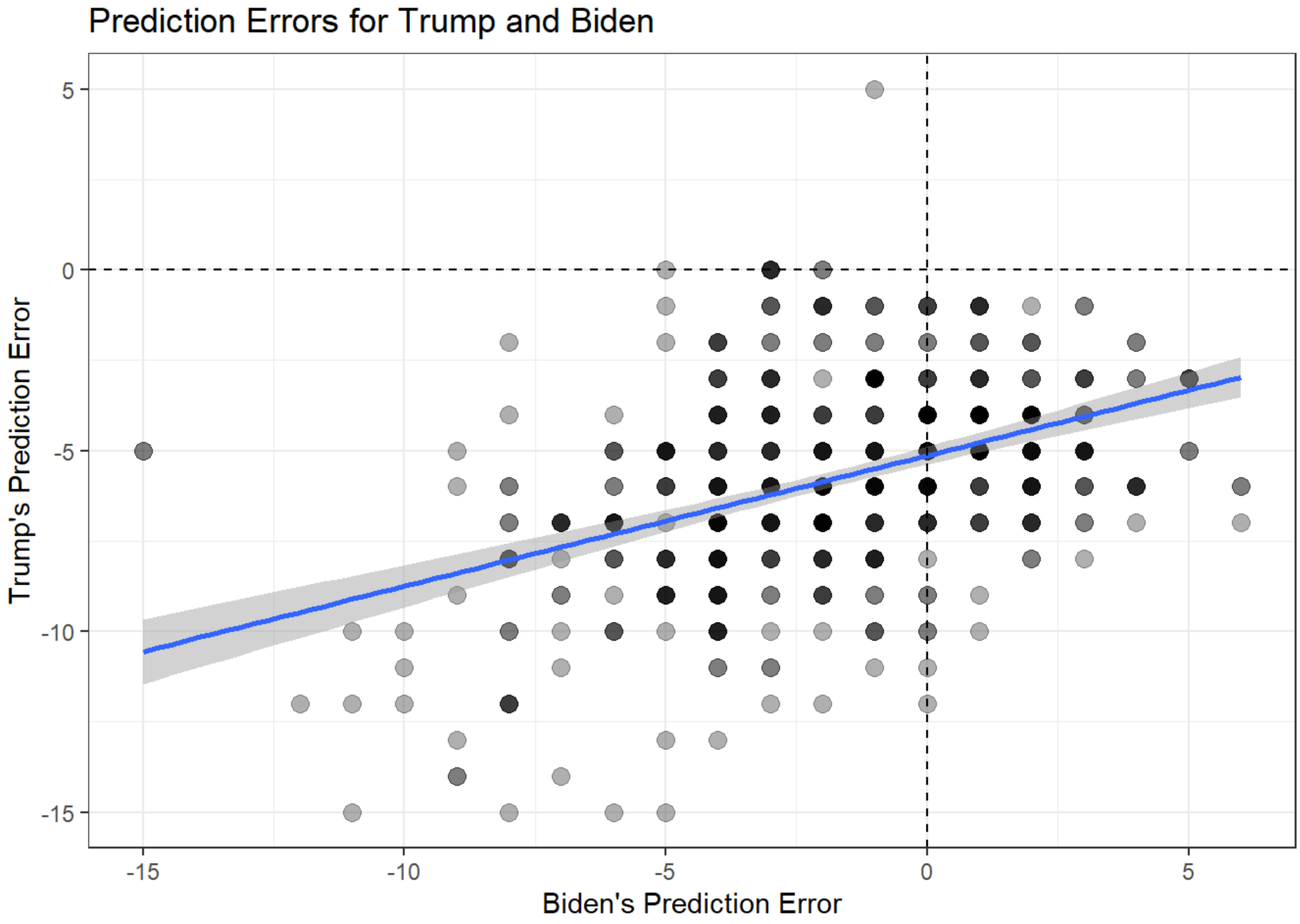
> 84.7% of polls had prediction errors that favored Biden over Trump. This means that polling in the United States is biased against Donald Trump, although it is not clear whether this is intentional or unintentional.

Extra Credit 1 [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. What do you conclude?

```
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'
```

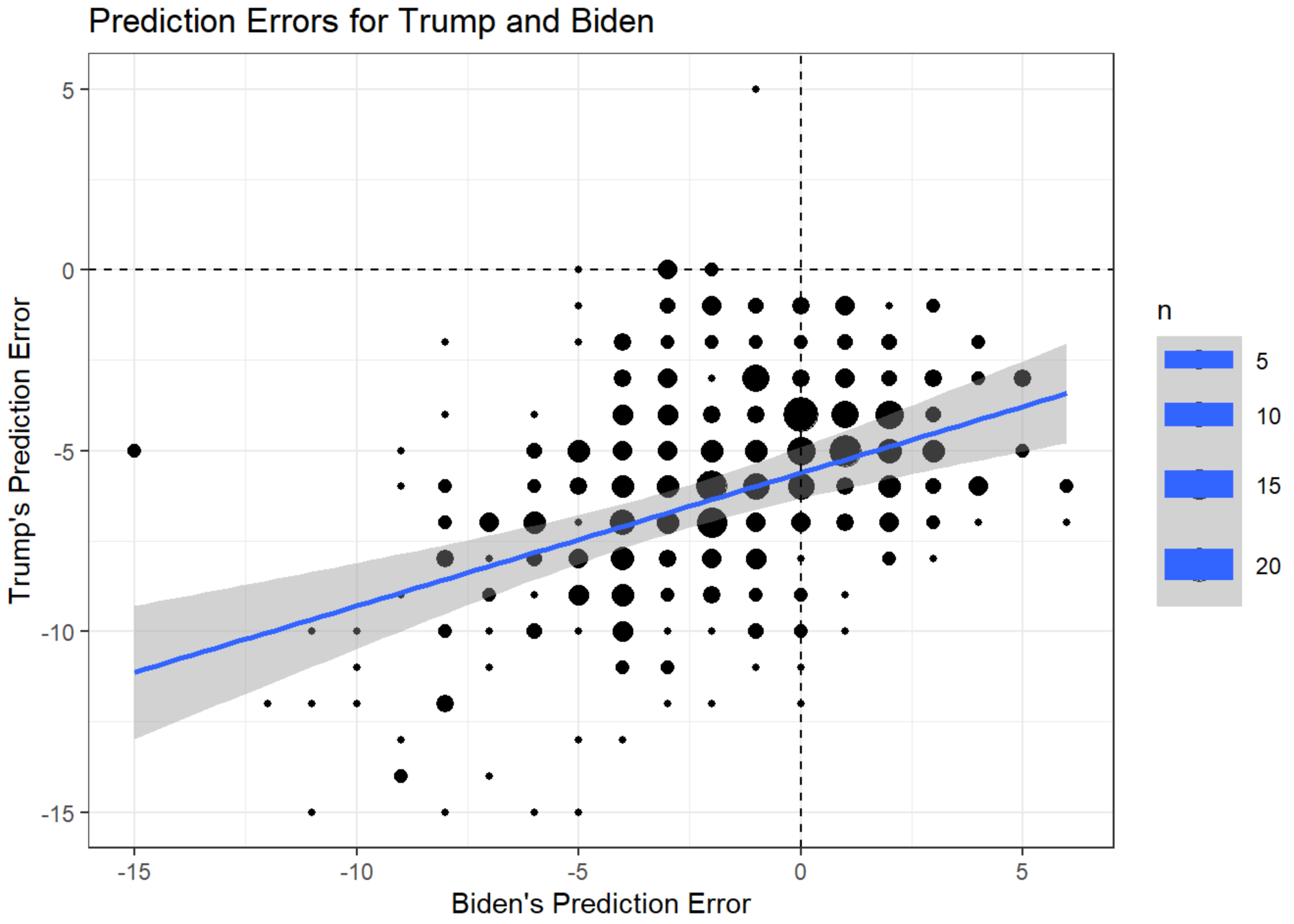


```
# Alternative #1
toplot %>%
  count(demErr,repErr) %>%
  ggplot(aes(x = demErr,y = repErr,size = n)) +
  geom_point() +
  geom_smooth(method = 'lm') +
  labs(title = "Prediction Errors for Trump and Biden",
       x = "Biden's Prediction Error",
       y = "Trump's Prediction Error") +
  geom_vline(xintercept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

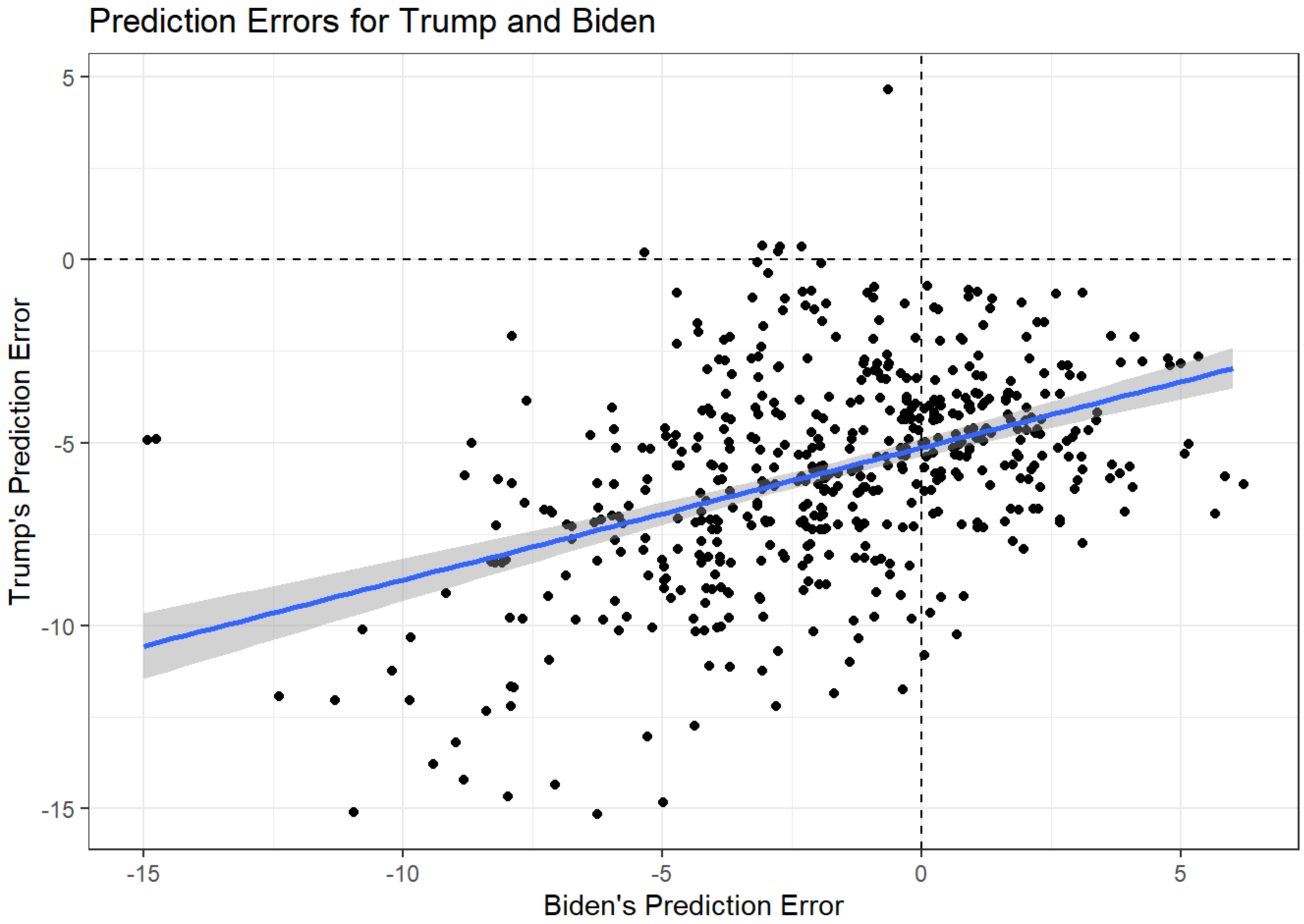
```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: The following aesthetics were dropped during statistical transformation: siz
e.
## 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
```

> The results show that the polls which underestimate Trump's support also underestimated Biden's support, as indicated by the positive slope. Almost 60% of the polls underpredicted both Biden and Trump, whereas 0% of the polls overpredicted both. Nevertheless, there is still some evidence of an anti-Trump bias, since 39% of polls underpredicted Trump and overpredicted Biden, while only 1.7% of polls overpredicted Trump and underpredicted Biden.

Question 5 [1 point]

Now let’s load a different dataset to practice multivariate visualization and confidence. Open `game_summary.Rds` *from the github page and save it to a new object called* `games` *. This dataset contains information on basketball games in the NBA from the 2017-2019 seasons. The codebook for it can be found in homework 5, which is also on github (https://github.com/jbisbee1/ISP_Data_Science_2024/blob/main/Homeworks/ISP_hw_5.pdf).*

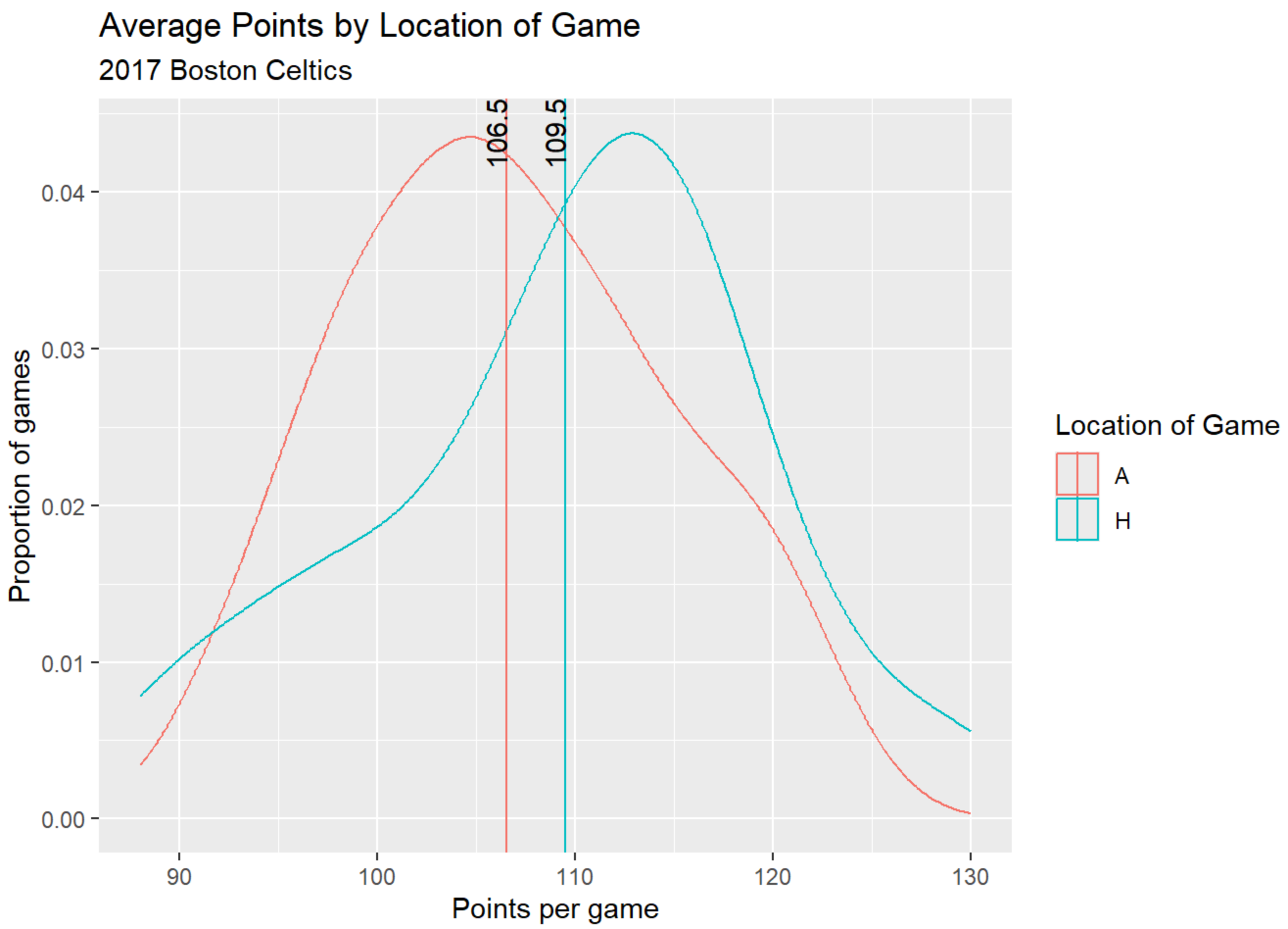
```
games <- read_rds("https://github.com/jbisbee1/ISP_Data_Science_2024/raw/main/data/game_
summary.Rds") # Insert link here
```

We are interested in the concept of “home court advantage”, which predicts that teams play better at home than on the road. We are focusing on one team in the dataset (meaning you will need to `filter()` *), my home team the Boston Celtics. How many points, on average, did the Boston Celtics score at home and away games in the 2017 season? Calculate this answer and also plot the multivariate relationship. Explain why your chosen visualization is justified. Draw two vertical lines for the average points at home and away.*

```
(vertLines <- games %>%
filter(yearSeason == 2017,
       nameTeam == 'Boston Celtics') %>% # Filter to the 2017 season (yearSeason) AND to
the Boston Celtics (nameTeam)
  group_by(locationGame) %>% # Group by the location of the game (locatoinGame)
  summarise(avg_pts = mean(pts,na.rm=T))) # Calculate the average points (pts)
```

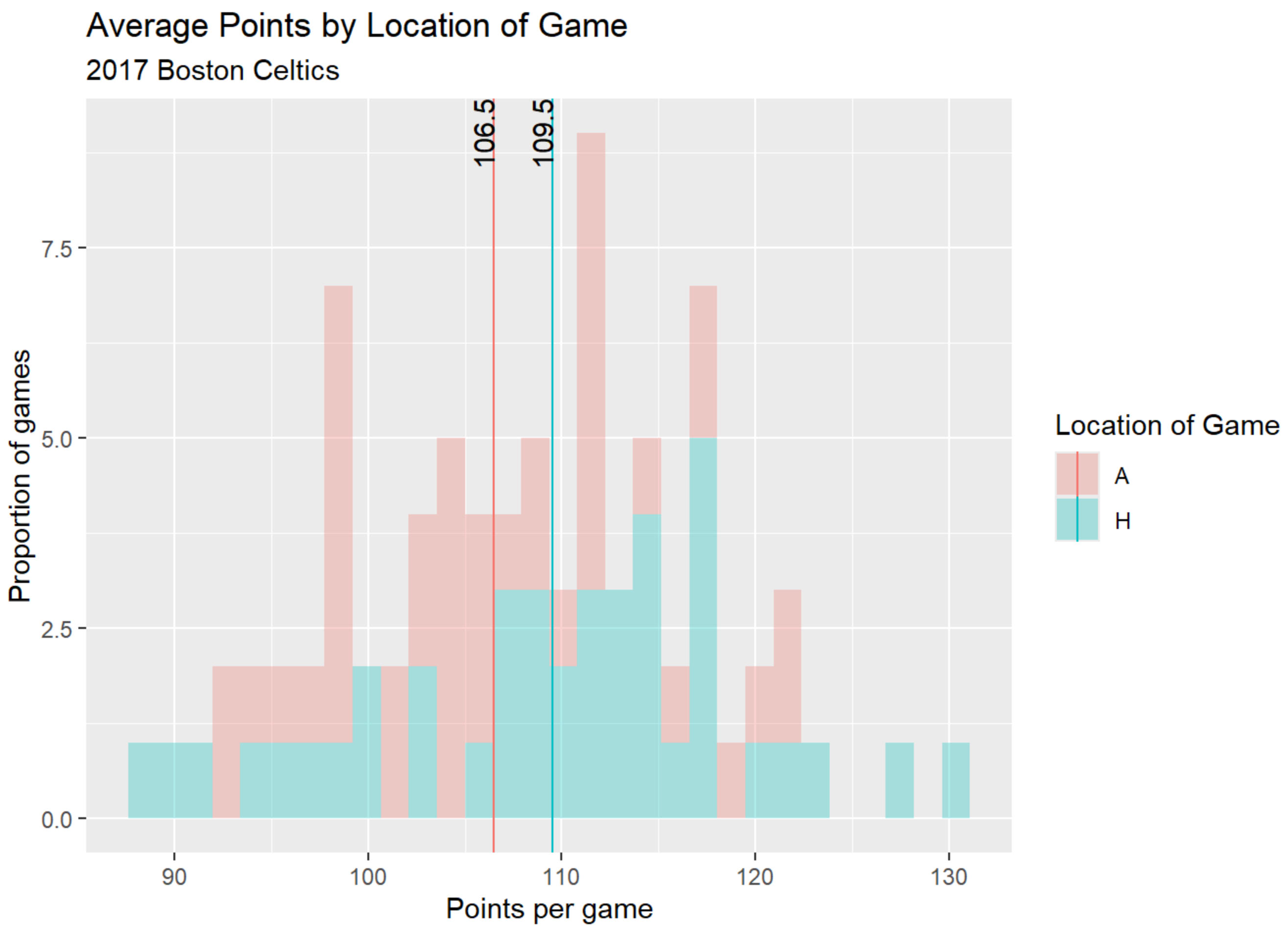
```
## # A tibble: 2 × 2
##   locationGame avg_pts
##   <chr>          <dbl>
## 1 A               107.
## 2 H               110.
```

```
games %>%
  filter(yearSeason == 2017,
         nameTeam == 'Boston Celtics') %>% # Filter to the 2017 season (yearSeason) AND
to the Boston Celtics (nameTeam)
  ggplot(aes(x = pts,color = locationGame)) + # Create a multivariate plot comparing poi
nts scored between home and away games
  geom_density() + # Choose the appropriate geom_... for this plot (i.e., geom_histogram
(), geom_density(), geom_bar(), etc.)
  labs(title = 'Average Points by Location of Game', # Add clear descriptions for the ti
tle, subtitle, axes, and legend
       subtitle = '2017 Boston Celtics',
       x = 'Points per game',
       y = 'Proportion of games',
       color = 'Location of Game') +
  geom_vline(data = vertLines,aes(xintercept = avg_pts,color = locationGame)) + # EC: ad
d vertical lines for the average points scored at home and away.
  annotate(geom = 'text',x = vertLines$avg_pts,y = Inf,label = round(vertLines$avg_pts,
1),vjust = 0,hjust = 1,angle = 90) # EC: label the vertical lines
```



```
games %>%
  filter(yearSeason == 2017,
         nameTeam == 'Boston Celtics') %>%
  ggplot(aes(x = pts,fill = locationGame)) +
  geom_histogram(alpha = .3) +
  labs(title = 'Average Points by Location of Game',
       subtitle = '2017 Boston Celtics',
       x = 'Points per game',
       y = 'Proportion of games',
       fill = 'Location of Game',
       color = 'Location of Game') +
  geom_vline(data = vertLines,aes(xintercept = avg_pts,color = locationGame)) +
  annotate(geom = 'text',x = vertLines$avg_pts,y = Inf,label = round(vertLines$avg_pts,
1),vjust = 0,hjust = 1,angle = 90)
```

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



> I chose a geom_density that was colored by the location of the game. I could have also chosen a histogram.

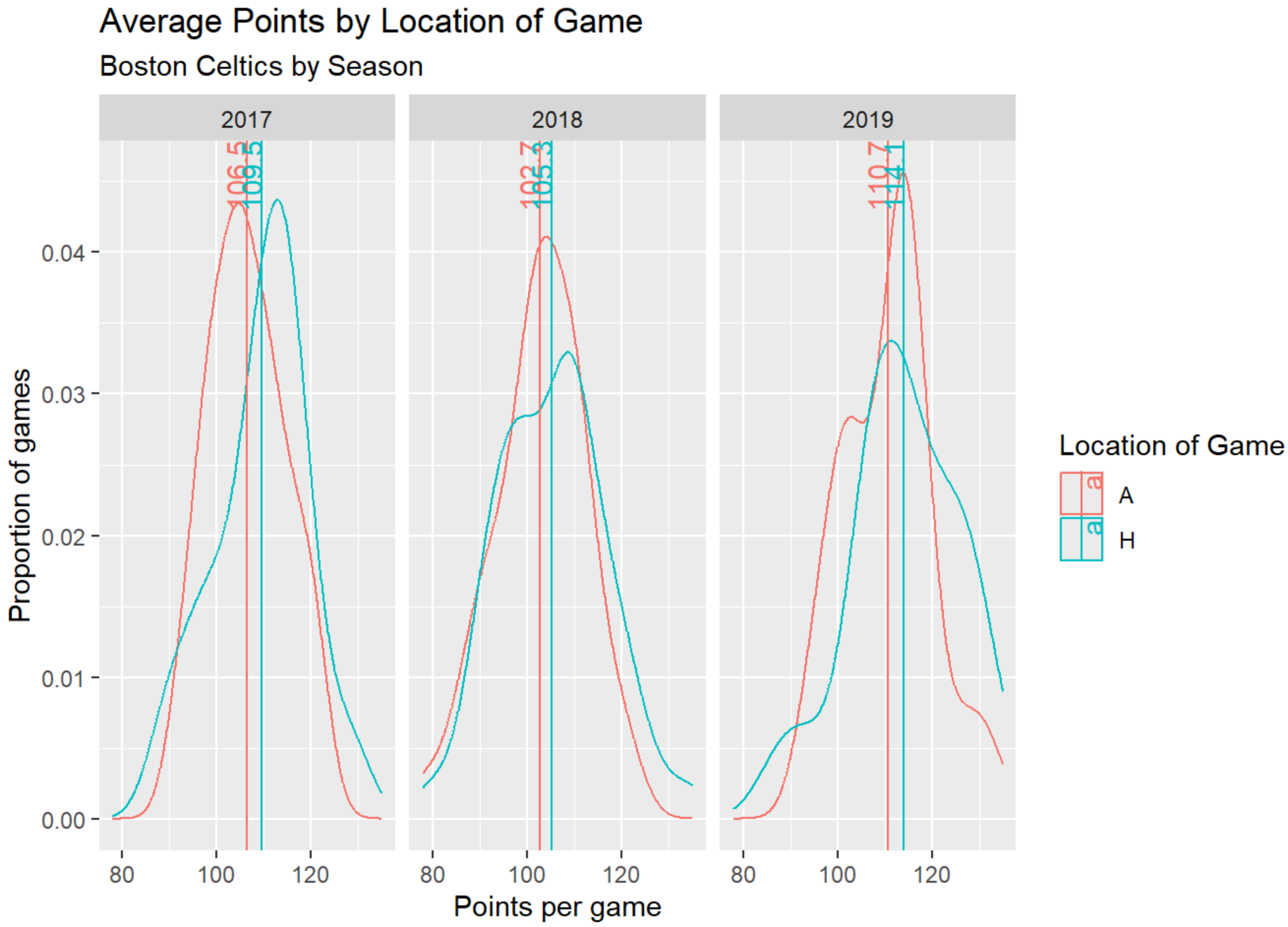
Question 6 [1 point]

Now recreate the same plot for the 2018, 2019, and combined seasons. Imagine that you work for the Celtics organization and Brad Stevens (the GM), asks you if the team scores more points at home or away? Based on your analysis, what would you tell him?

```
# By season
(vertLines <- games %>%
filter(nameTeam == 'Boston Celtics') %>% # Filter to the Boston Celtics (nameTeam)
  group_by(locationGame,yearSeason) %>% # Group by the location (locationGame) and the s
eason (yearSeason)
  summarise(avg_pts = mean(pts,na.rm=T))) # Calculate the average points (pts)
```

```
## `summarise()` has grouped output by 'locationGame'. You can override using the
## `.groups` argument.
## # A tibble: 6 × 3
## # Groups:   locationGame [2]
##   locationGame yearSeason avg_pts
##   <chr>             <int>   <dbl>
## 1 A                  2017    107.
## 2 A                  2018    103.
## 3 A                  2019    111.
## 4 H                  2017    110.
## 5 H                  2018    105.
## 6 H                  2019    114.
```

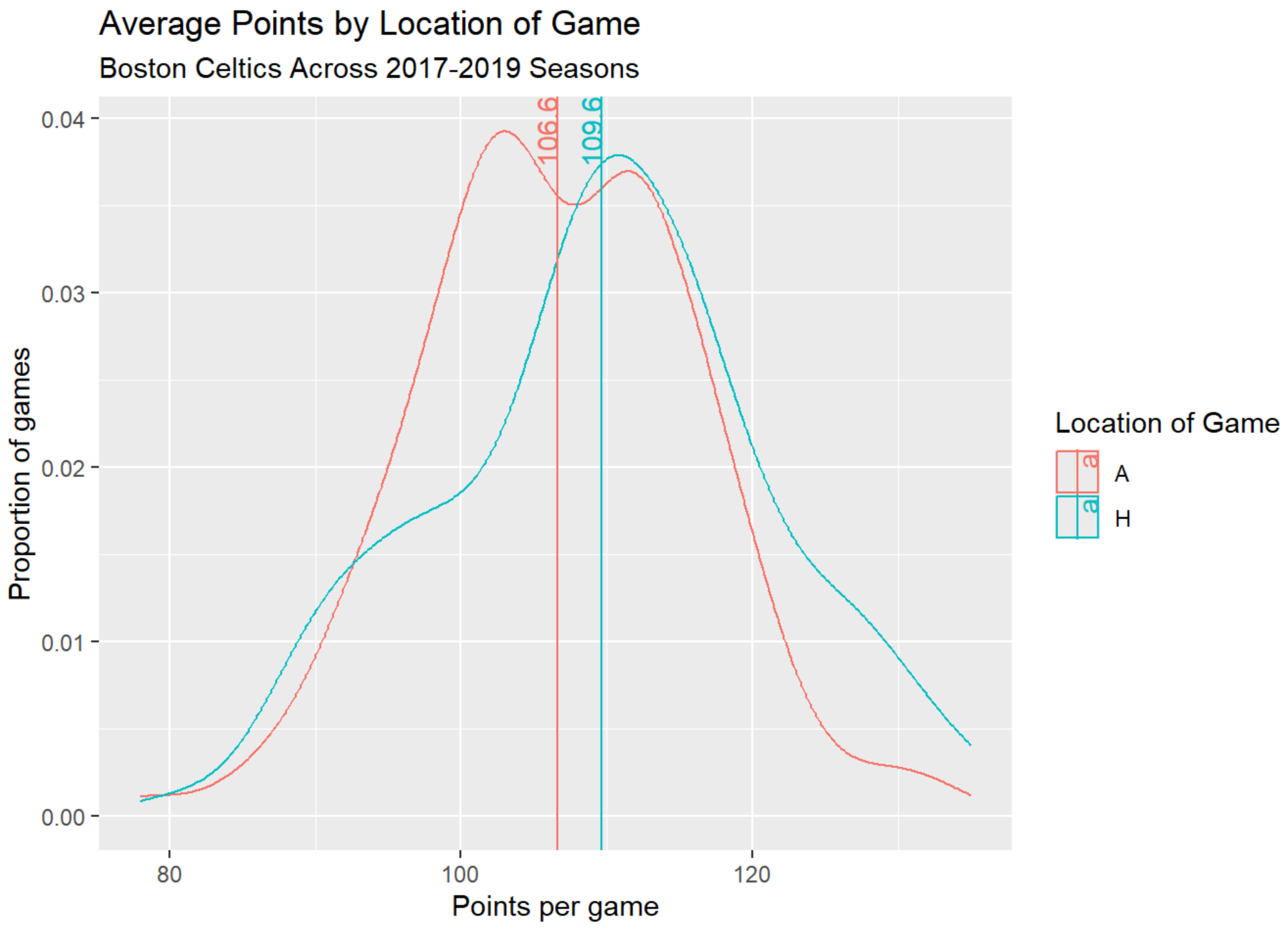
```
games %>%
  filter(nameTeam == 'Boston Celtics') %>% # Filter to the 2017 season (yearSeason) AND to the Boston Celtics (nameTeam)
  ggplot(aes(x = pts,color = locationGame)) + # Create a multivariate plot comparing points scored between home and away games
  geom_density() + # Choose the appropriate geom_... for this plot (i.e., geom_histogram(), geom_density(), geom_bar(), etc.)
  labs(title = 'Average Points by Location of Game', # Add clear descriptions for the title, subtitle, axes, and legend
       subtitle = 'Boston Celtics by Season',
       x = 'Points per game',
       y = 'Proportion of games',
       color = 'Location of Game') +
  facet_wrap(~yearSeason) + # Create separate panels for each season (facet_wrap())
  geom_vline(data = vertLines,aes(xintercept = avg_pts,color = locationGame)) +
  geom_text(data = vertLines,aes(x = avg_pts,y = Inf,color = locationGame,label = round(avg_pts,1)),
            vjust = 0,hjust = 1,angle = 90)
```



```
# Over all seasons combined
(vertLines <- games %>%
filter(nameTeam == 'Boston Celtics') %>% # Filter to the Boston Celtics (nameTeam)
  group_by(locationGame) %>% # Group by the location (locationGame)
  summarise(avg_pts = mean(pts,na.rm=T))) # Calculate the average points (pts)
```

```
## # A tibble: 2 × 2
##   locationGame avg_pts
##   <chr>          <dbl>
## 1 A               107.
## 2 H               110.
```

```
games %>%
  filter(nameTeam == 'Boston Celtics') %>% # Filter to the 2017 season (yearSeason) AND
to the Boston Celtics (nameTeam)
  ggplot(aes(x = pts,color = locationGame)) + # Create a multivariate plot comparing poi
nts scored between home and away games
  geom_density() + # Choose the appropriate geom_... for this plot (i.e., geom_histogram
(), geom_density(), geom_bar(), etc.)
  labs(title = 'Average Points by Location of Game', # Add clear descriptions for the ti
tle, subtitle, axes, and legend
       subtitle = 'Boston Celtics Across 2017-2019 Seasons',
       x = 'Points per game',
       y = 'Proportion of games',
       color = 'Location of Game') +
  geom_vline(data = vertLines,aes(xintercept = avg_pts,color = locationGame)) +
  geom_text(data = vertLines,aes(x = avg_pts,y = Inf,color = locationGame,label = round
(avg_pts,1)),
            vjust = 0,hjust = 1,angle = 90)
```



> The Celtics scored more points at home games than away games for every season in the data, as well as when combining all the seasons together. Based on this analysis, I would tell Brad Stevens that the Celtics score more points at home games than at away games. Overall, the difference is equivalent to roughly one 3-point shot: 106.6 points at away games and 109.6 points at home games.

Question 7 [1 point]

Brad Stevens thanks you for your answer, but is a well-trained statistician in his own right, and wants to know how confident you are in your claim. Bootstrap sample the data 1,000 times to provide him with a more sophisticated answer. How confident are you in your conclusion that the Celtics score more points at home games than away games? Make sure to `set.seed(123)` *to ensure you get the same answer every time you* `knit` *your code!*

```
set.seed(123) # Set the seed!
forBS <- games %>% # To make things easier, create a new data object that is filtered to just the Celtics
    filter(nameTeam == 'Boston Celtics') # Filter to the Celtics (nameTeam)

bsRes <- NULL # Instantiate an empty object to store data from the loop
for(i in 1:1000) { # Loop 1,000 times
  bsRes <- forBS %>%
    sample_n(size = nrow(forBS),replace = T) %>% # Sample the data with replacement using all possible rows
    group_by(locationGame) %>% # Group by the location of the game (locationGame)
    summarise(avg_pts = mean(pts,na.rm=T)) %>% # Calculate the average points (pts)
    ungroup() %>% # Best practices!
    spread(locationGame,avg_pts) %>% # Spread the data to get one column for average points at home and another for average points away
    mutate(diff = H - A, # Calculate the difference between home and away points
           bsInd = i) %>% # Save the bootstrap index
    bind_rows(bsRes) # Append the result to the empty object from line 133
}

# Calculate the confidence
bsRes %>%
  summarise(confidence = mean(diff > 0), # Calculate the proportion of bootstrap simulations where the home points are greater than the away points
            avg_diff = mean(diff)) # Calculate the overall average difference
```

```
## # A tibble: 1 × 2
##   confidence avg_diff
##        <dbl>    <dbl>
## 1      0.992     2.93
```

> I am 99.2% confident in my conclusion that the Celtics score more points at home games than away games. Furthermore, the average difference is about 3 points (2.93) over the 1,000 bootstrapped simulations.

Question 8 [2 points]

Re-do this analysis for three other statistics of interest to Brad: total rebounds (treb), turnovers (tov), and field goal percent (pctFG). **NOT GRADED:** *Do you notice anything strange in these results? What might explain it?*

```r
bsRes <- NULL # Instantiate an empty object to store data from the loop
for(i in 1:1000) { # Loop 1,000 times
  bsRes <- forBS %>%
    sample_n(size = nrow(forBS),replace = T) %>% # Sample the data with replacement using all possible rows
    group_by(locationGame) %>% # Group by the location of the game (locationGame)
    summarise(avg_reb = mean(treb,na.rm=T), # Calculate the average total rebounds (treb)
              avg_tov = mean(tov,na.rm=T), # Calculate the average turnovers (tov)
              avg_pctFG = mean(pctFG,na.rm=T)) %>% # Calculate the average field goal shooting percentage (pctFG)
    ungroup() %>% # Best practices!
    pivot_wider(names_from = locationGame, # Pivot wider to get each measure in its own colunm for homme and away games
                values_from = c('avg_reb','avg_tov','avg_pctFG')) %>% # Use the values from the variables you created above
    mutate(diff_reb = avg_reb_H - avg_reb_A, # Calculate the difference between home and away total rebounds
           diff_tov = avg_tov_H - avg_tov_A, # Calculate the difference between home and away turnovers
           diff_pctFG = avg_pctFG_H - avg_pctFG_A, # Calculate the difference between home and away field goal percentages
           bsInd = i) %>% # Save the bootstrap index
    bind_rows(bsRes) # Append the result to the empty object from line 165
}

# Calculate the confidence
bsRes %>%
  summarise(confidence_reb = mean(diff_reb > 0),
            confidence_tov = mean(diff_tov > 0),
            confidence_pctFG = mean(diff_pctFG > 0))
```

```
## # A tibble: 1 × 3
##   confidence_reb confidence_tov confidence_pctFG
##            <dbl>          <dbl>            <dbl>
## 1          0.994          0.923            0.885
```

> I am 99.4% confident that the Celtics rebound more at home games than away games. I am 92.3% confident that they turn over the ball more at home games than away games. And I am 88.5% confident that they shoot more accurately at home than away games. **NOT GRADED:** These results are surprising since turnovers are theoretically bad for a basketball team, yet we find that the Celtics have more turnovers at home games than away games. This might be due to a faster pace of play, where the Celtics move the ball around more, providing more opportunityies for points and rebounds, but also more turnovers.

Extra Credit 2 [2 points]

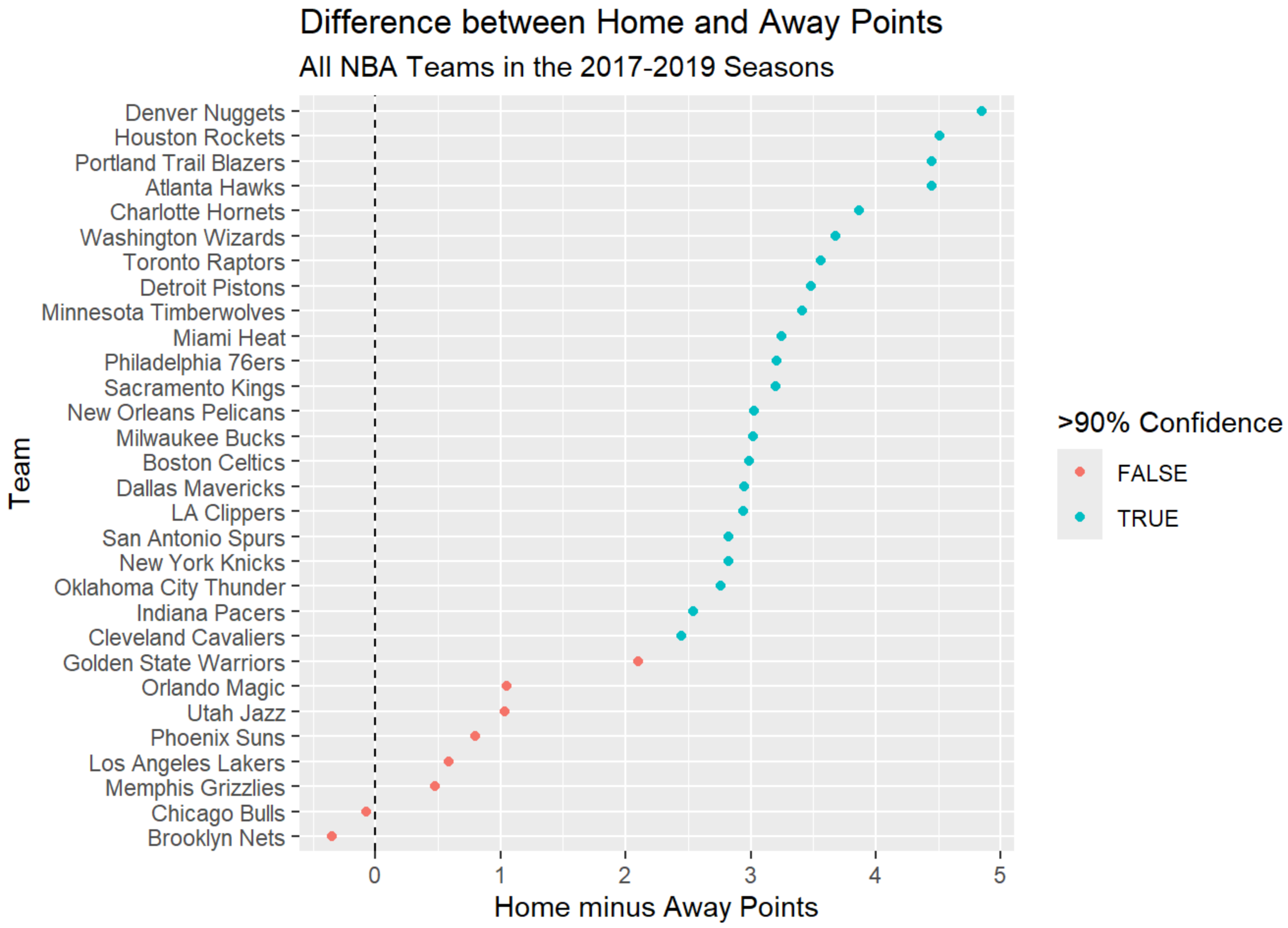
Now Brad is asking for a similar analysis of other teams. Calculate the difference between home and away points for every team in the league and prepare a summary table that includes both the average difference for each team, as well as your confidence about whether the difference is not zero. Based on these data, would you argue that there is an **overall** *home court advantage in terms of points across the NBA writ large? Visualize these summary results by plotting the difference on the x-axis, the teams (reordered) on the y-axis, and the points colored by whether you are more than 90% confident in your answer. What does it mean to have less than 50% confidence?*

```r
bsRes <- NULL # Instantiate an empty object to store data from the loop
for(i in 1:1000) { # Loop 1,000 times
  bsRes <- games %>%
    group_by(nameTeam) %>%
    sample_n(size = n(),replace = T) %>% # Sample the data with replacement using all possible rows
    group_by(locationGame,nameTeam) %>% # Group by the location of the game (locationGame)
    summarise(avg_pts = mean(pts,na.rm=T),.groups = 'drop') %>% # Calculate the average turnovers (tov)
    pivot_wider(id_cols = nameTeam,
                names_from = locationGame, # Pivot wider to get each measure in its own colunm for homme and away games
                values_from = c('avg_pts')) %>% # Use the values from the variables you created above
    mutate(diff = H - A, # Calculate the difference between home and away turnovers
           bsInd = i) %>% # Save the bootstrap index
    bind_rows(bsRes) # Append the result to the empty object from line 165
}

(toplot <- bsRes %>%
  group_by(nameTeam) %>%
  summarise(conf = round(mean(diff > 0),2),
            diff = round(mean(diff),2)))
```

```
## # A tibble: 30 × 3
##    nameTeam               conf  diff
##    <chr>                 <dbl> <dbl>
##  1 Atlanta Hawks          1     4.45
##  2 Boston Celtics         0.99  2.99
##  3 Brooklyn Nets          0.43 -0.35
##  4 Charlotte Hornets      0.99  3.87
##  5 Chicago Bulls          0.5  -0.07
##  6 Cleveland Cavaliers    0.92  2.45
##  7 Dallas Mavericks       0.98  2.95
##  8 Denver Nuggets         1     4.85
##  9 Detroit Pistons        0.99  3.48
## 10 Golden State Warriors  0.88  2.1
## # ℹ 20 more rows
```

```
toplot %>%
  ggplot(aes(x = diff,y = reorder(nameTeam,diff),color = conf > .9 | conf < .1))+
  geom_point() +
  geom_vline(xintercept = 0,linetype = 'dashed') +
  labs(title = 'Difference between Home and Away Points',
       subtitle = 'All NBA Teams in the 2017-2019 Seasons',
       x= 'Home minus Away Points',
       y = 'Team',
       color = '>90% Confidence')
```



> Here we find much stronger evidence that teams generally score more points at home than away games across the NBA. Every team except the Bulls and Nets score more points at home than away, and the majority of these differences we can confidently say are greater than zero at the 90% level. Confidence levels less than 50% mean that teams scored more home points than away points in fewer than 50% of simulated realities. This is equivalent to saying that they scored more away points than home points in more than 50% of simulated realities. In other words, if the confidence is less than 0.5, we can flip the statement and say we are 1-the confidence.