Lecture 7

2024-07-10

Biden and Trump Bias in Polling

```
require(tidyverse)
```

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

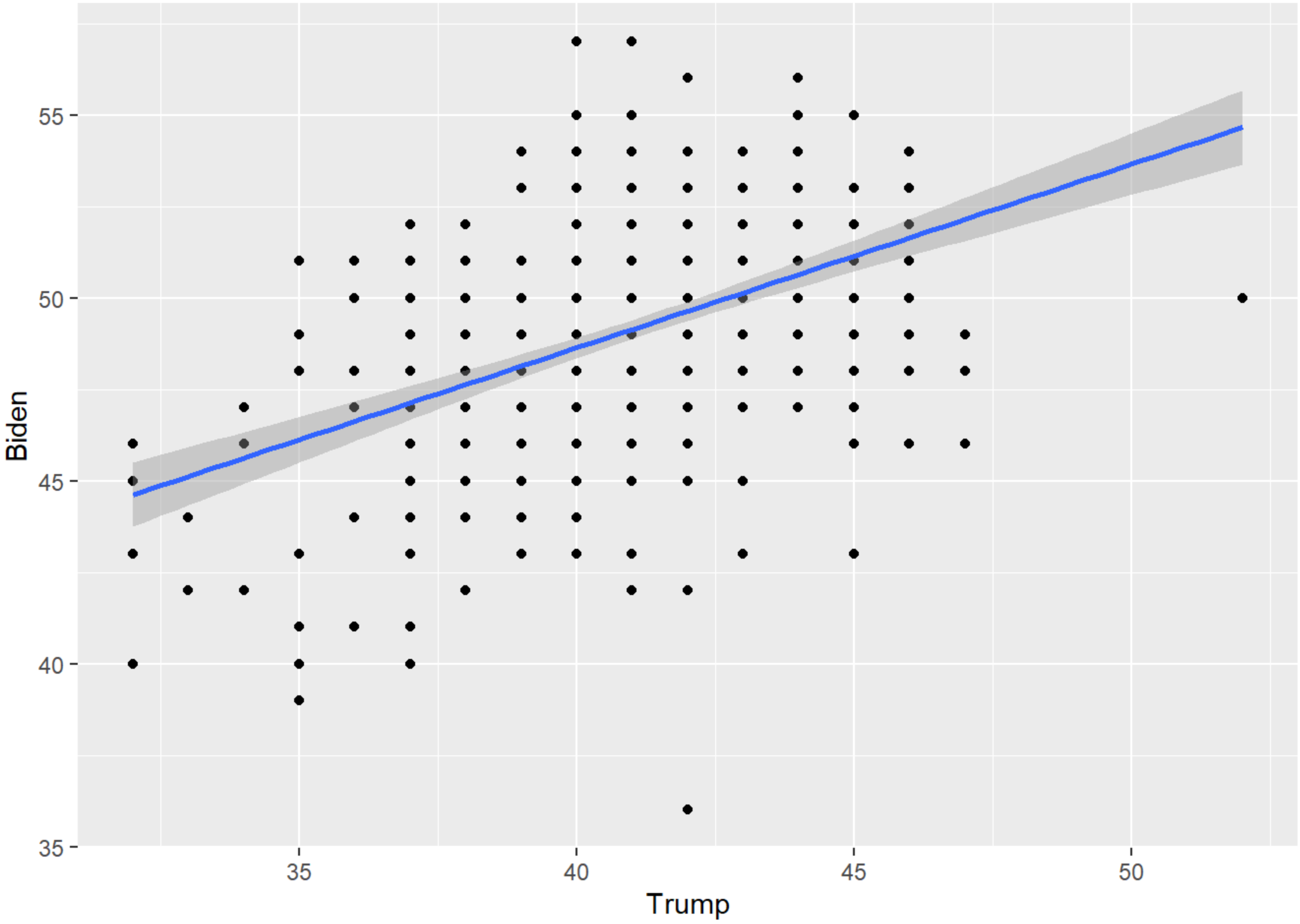
```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── 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
```

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

Plot multivariate visualization

```
pres %>%
  ggplot(aes(x = Trump,
             y = Biden)) +
  geom_point() +
  geom_smooth(method = 'lm')
```

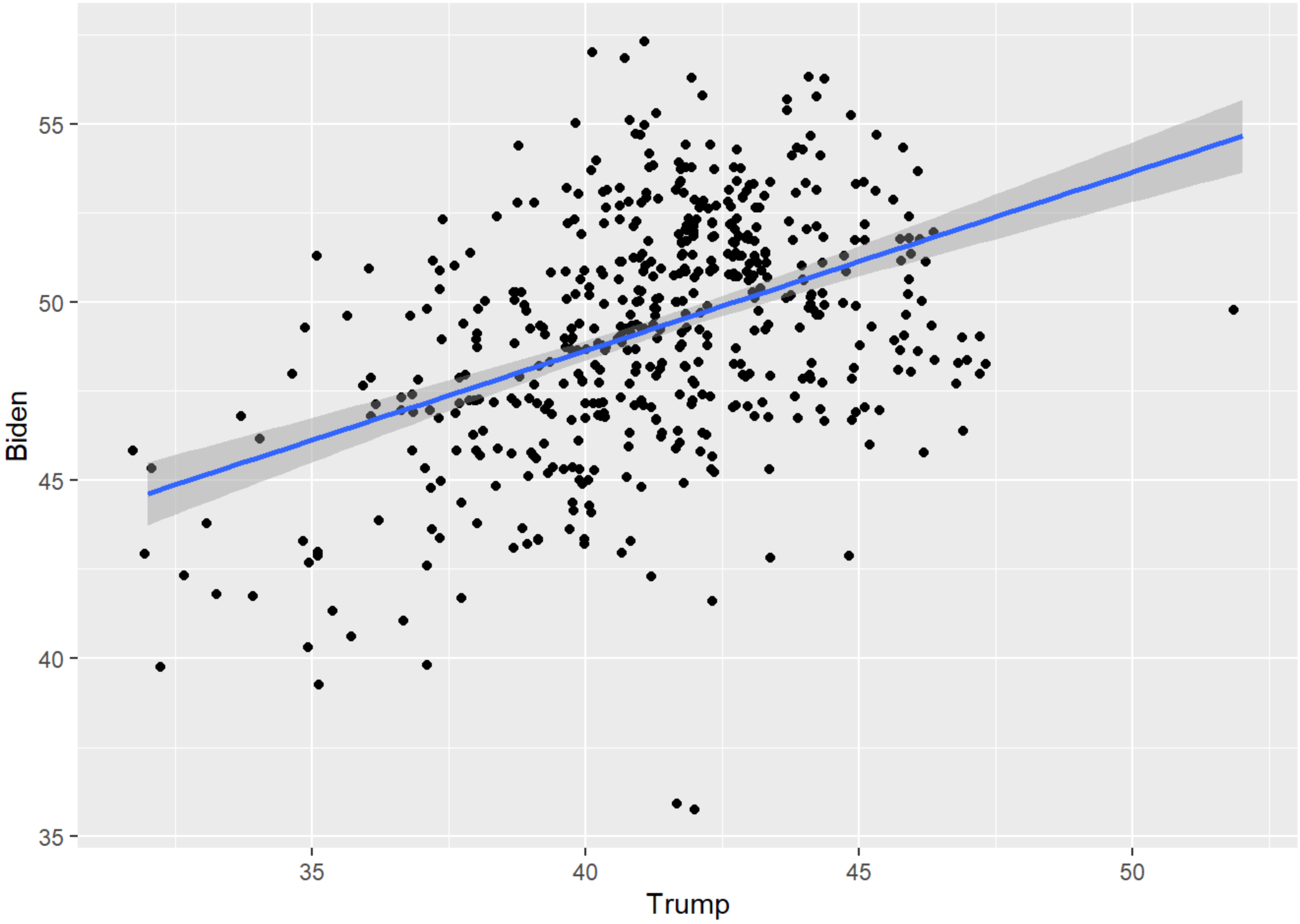
```
## `geom_smooth()` using formula = 'y ~ x'
```



Fixing the visualization to show multiple polls on the same coordinates

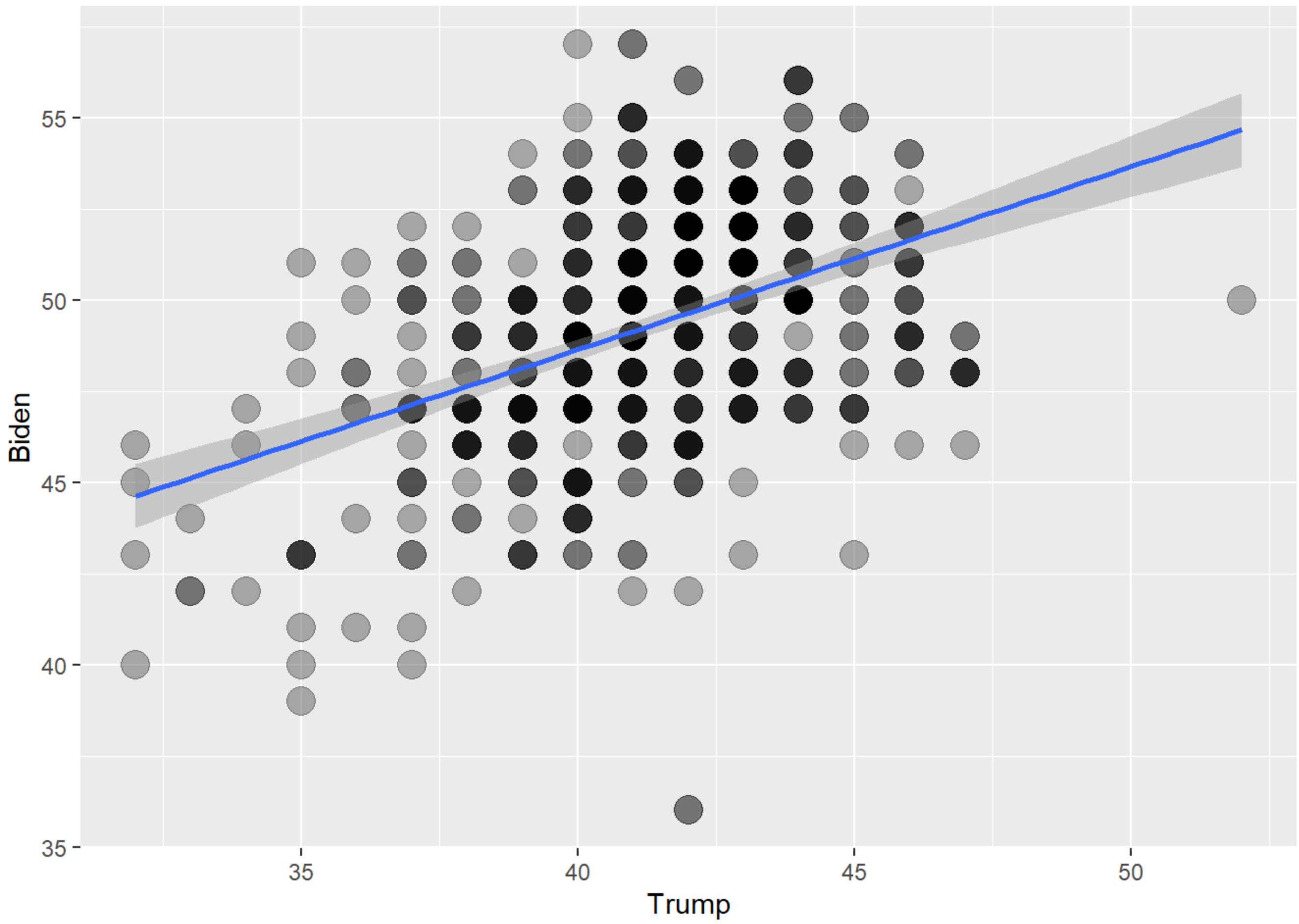
```
# Method 1: geom_jitter()
pres %>%
  ggplot(aes(x = Trump,
             y = Biden)) +
  geom_jitter() +
  geom_smooth(method = 'lm')
```

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



```
# Method 2: size + alpha
pres %>%
  ggplot(aes(x = Trump,
             y = Biden)) +
  geom_point(alpha = .3,size = 5) +
  geom_smooth(method = 'lm')
```

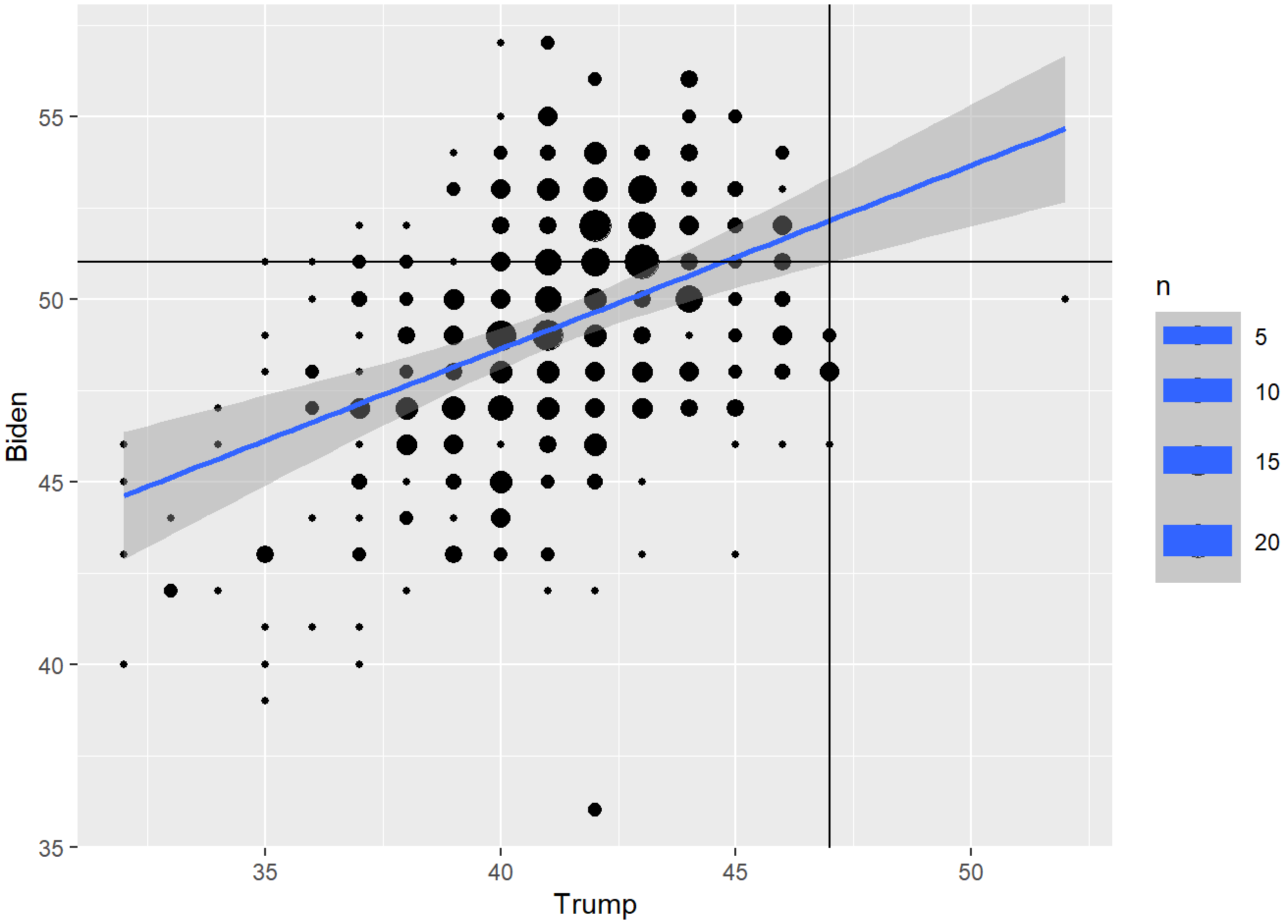
```
## `geom_smooth()` using formula = 'y ~ x'
```



```
# Method 3: count() + size
View(pres)
p1 <- pres %>%
  count(Trump,Biden) %>%
  ggplot(aes(x = Trump,
             y = Biden,
             size = n)) +
  geom_point() +
  geom_smooth(method = 'lm',aes(weight = n)) +
  geom_vline(xintercept = 47) +
  geom_hline(yintercept = 51)

p1
```

```
## 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?
```



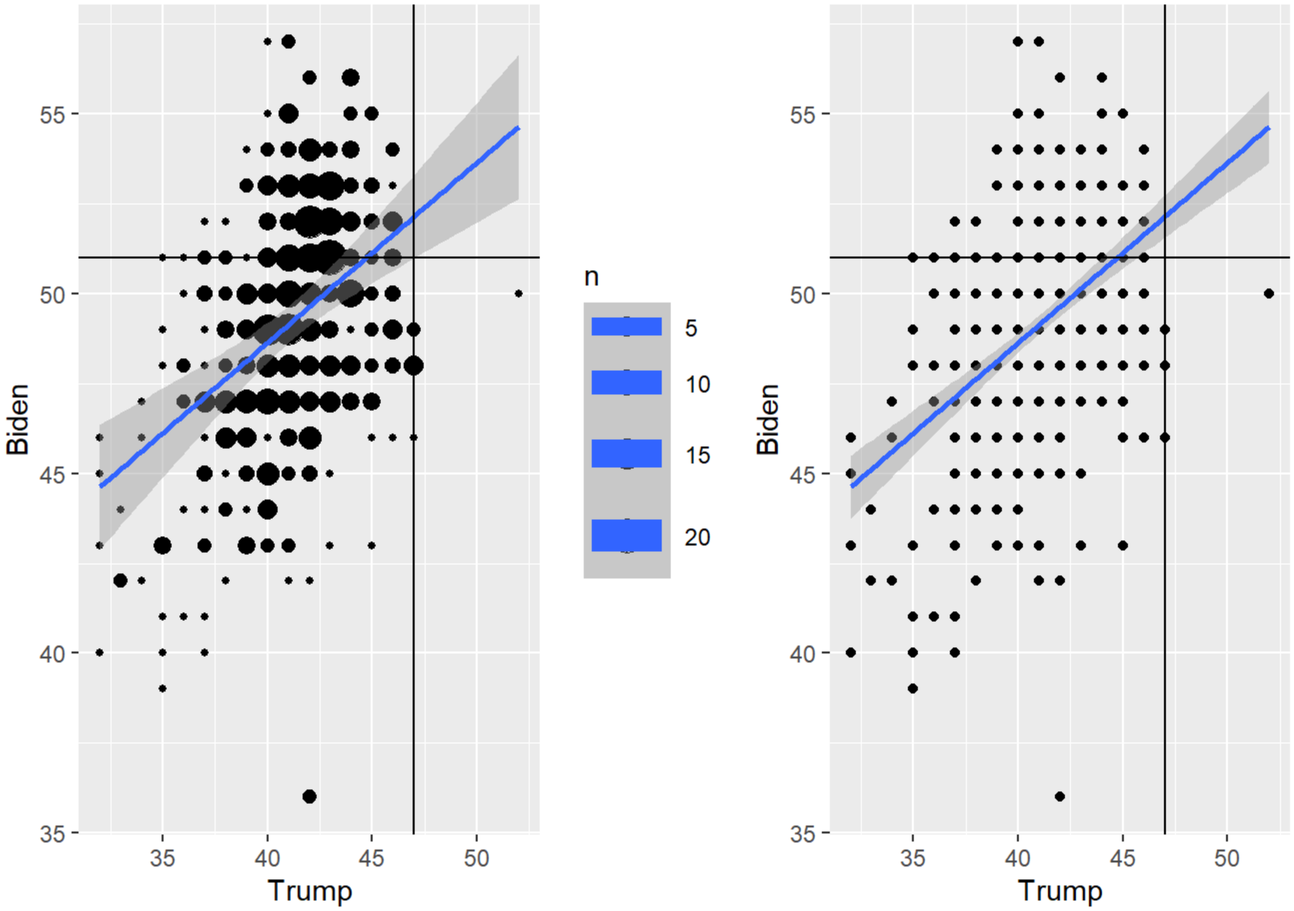
```
p2 <- pres %>%
  ggplot(aes(x = Trump,y = Biden)) +
  geom_point() +
  geom_smooth(method = 'lm')  +
  geom_vline(xintercept = 47) +
  geom_hline(yintercept = 51)

require(patchwork)
```

```
## Loading required package: patchwork
```

```
p1 + p2
```

```
## `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?
```

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

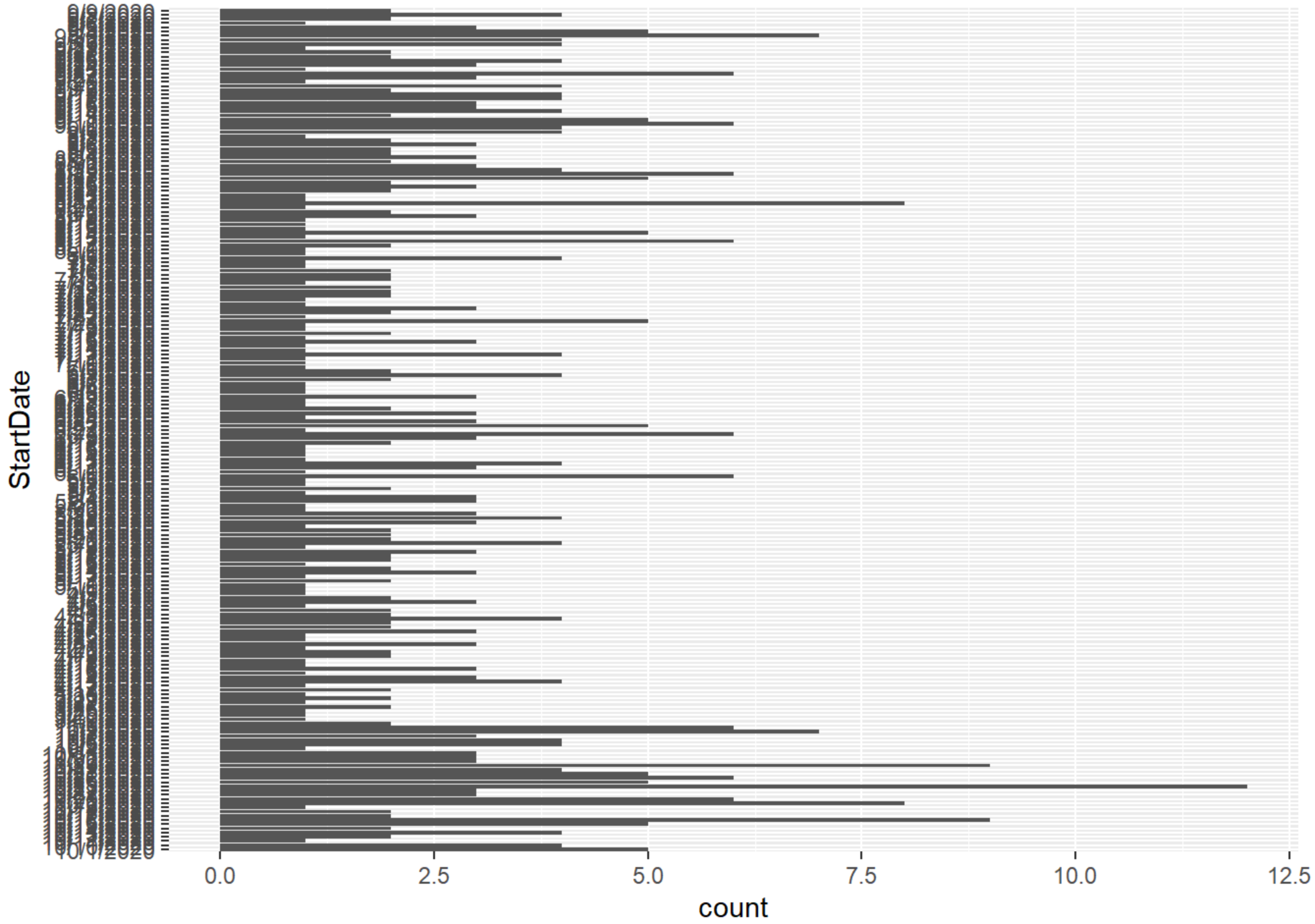


Looking at time

```
# Idea #1: It is a categorical variable
# Step 1: Look
pres %>%
  select(StartDate)
```

```
## # A tibble: 528 × 1
##    StartDate
##    <chr>
##  1 10/31/2020
##  2 10/31/2020
##  3 10/29/2020
##  4 11/1/2020
##  5 11/1/2020
##  6 10/30/2020
##  7 10/31/2020
##  8 10/30/2020
##  9 10/29/2020
## 10 10/29/2020
## # ℹ 518 more rows
```

```
# Step 2: Univariate Visualization
pres %>%
  ggplot(aes(y = StartDate)) +
  geom_bar()
```



```
# Conclusion: text DOESN'T WORK

# Idea #2: numeric
# pres %>%
#   ggplot(aes(x = StartDate)) +
#   geom_histogram()

# Conclusion: numeric DOESN'T WORK
```

New function: as.Date()

```
example_date <- "2024-07-10"

exam_date <- "2024-07-11"

#exam_date - example_date

as.Date(exam_date) - as.Date(example_date)
```

```
## Time difference of 1 days
```

```
# American date example
today_date <- "07/10/2024"

as.Date(today_date)
```

```
## [1] "0007-10-20"
```

```
as.Date(today_date,"%m/%d/%Y")
```

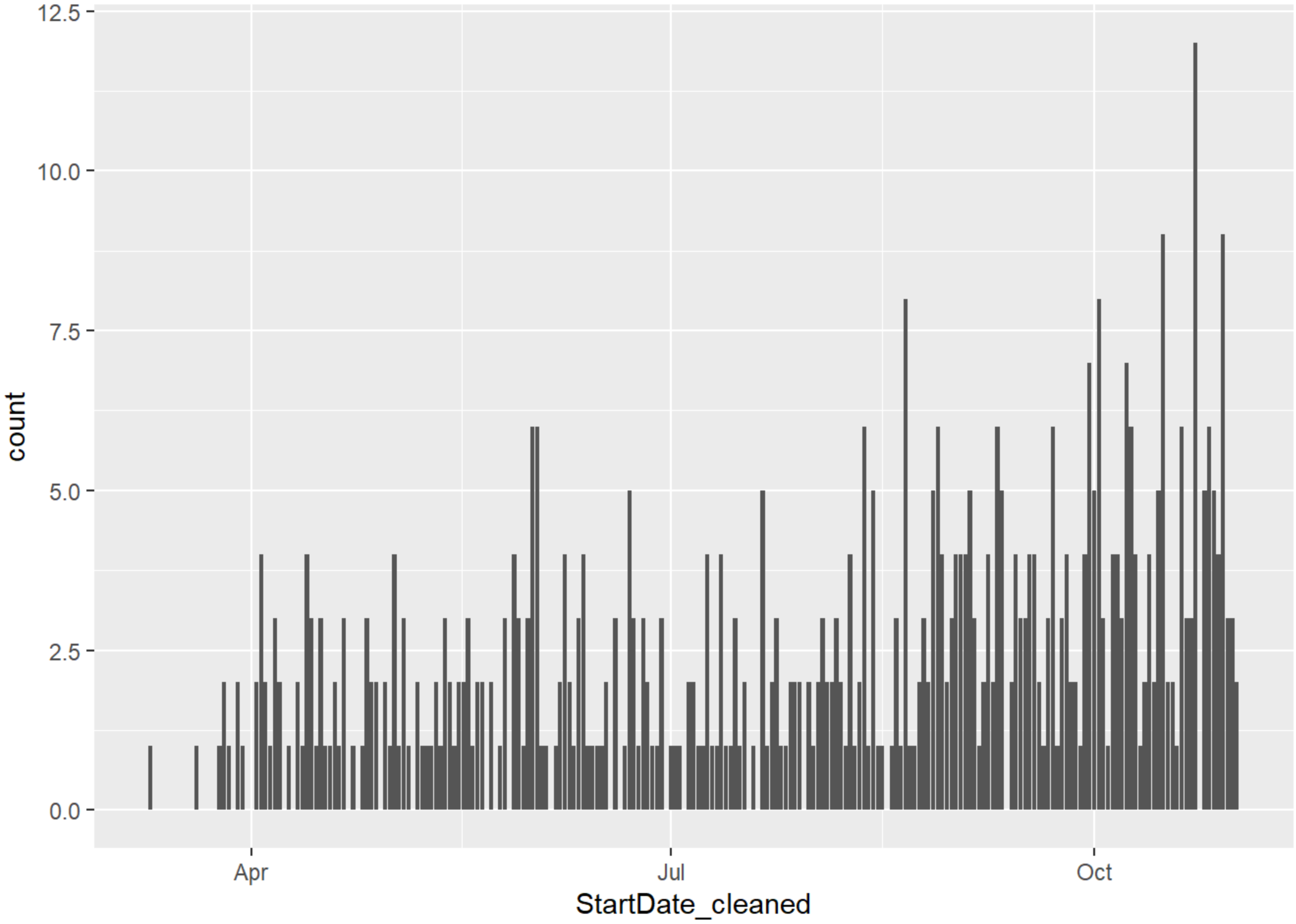
```
## [1] "2024-07-10"
```

Using new function in our data

```
"Jul 10, 2024"
```

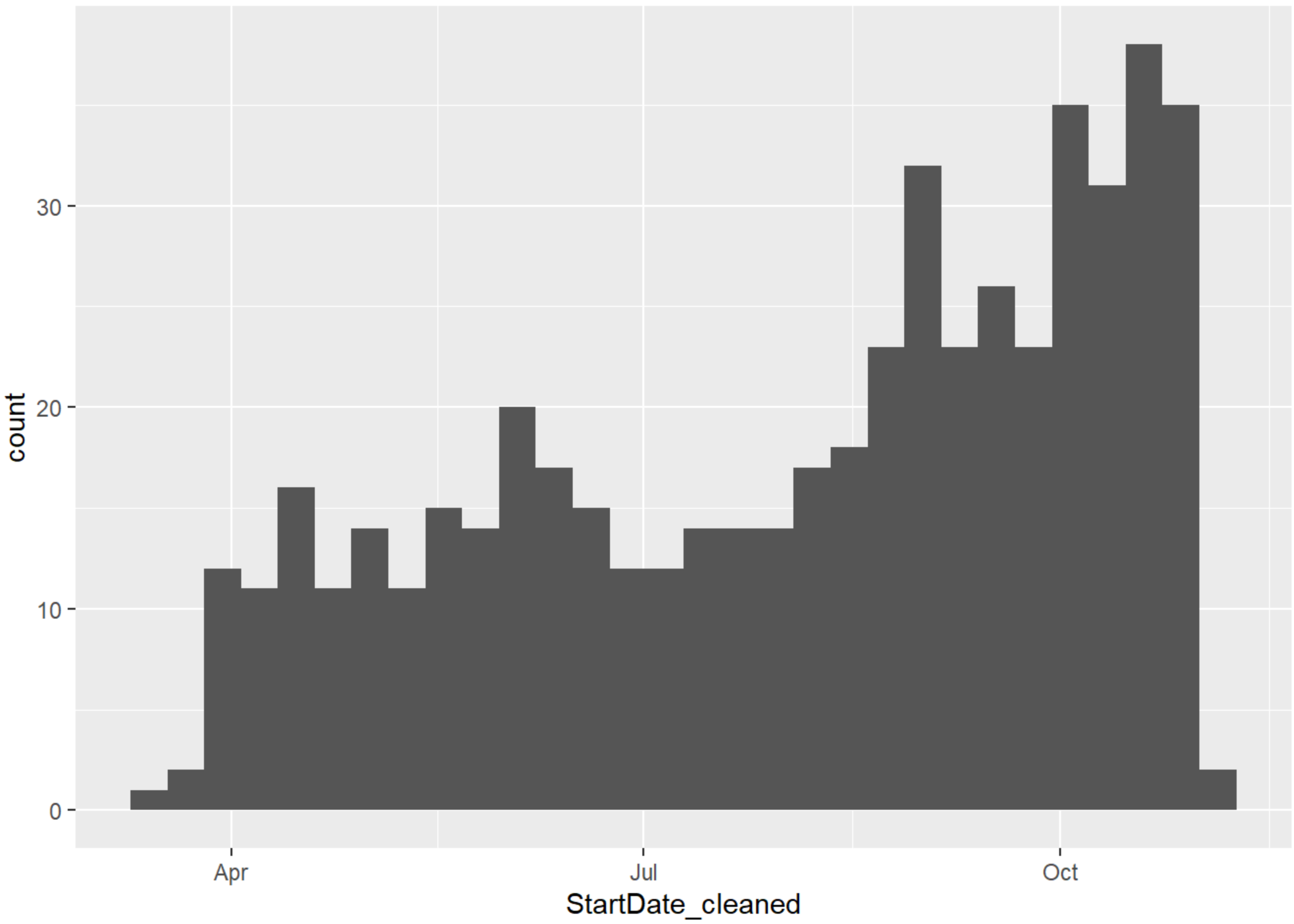
```
## [1] "Jul 10, 2024"
```

```
pres <- pres %>%
  mutate(StartDate_cleaned = as.Date(StartDate,'%m/%d/%Y'))

pres %>%
  ggplot(aes(x = StartDate_cleaned)) +
  geom_bar()
```



```
# Treat as numeric
pres %>%
  ggplot(aes(x = StartDate_cleaned)) +
  geom_histogram()
```

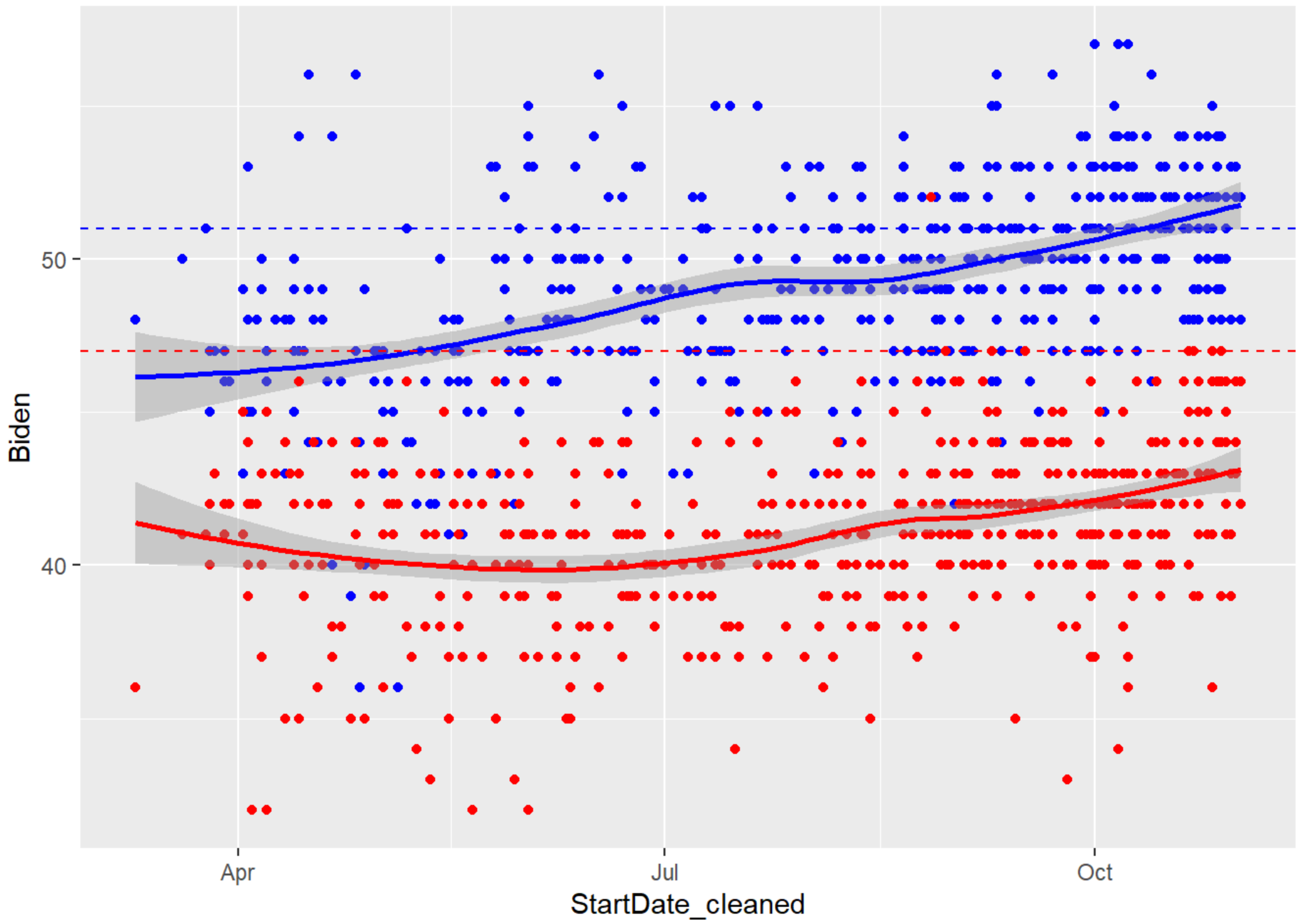
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Multivariate:

```
pres %>%
  ggplot(aes(x = StartDate_cleaned,
             y = Biden)) +
  geom_point(color = 'blue') +
  geom_smooth(color = 'blue') +
  geom_hline(yintercept = 51,
             color = 'blue',
             linetype='dashed') +
  geom_point(color = 'red',
             aes(y = Trump)) +
  geom_smooth(color = 'red',
              aes(y = Trump)) +
  geom_hline(yintercept = 47,
             color = 'red',
             linetype='dashed')
```

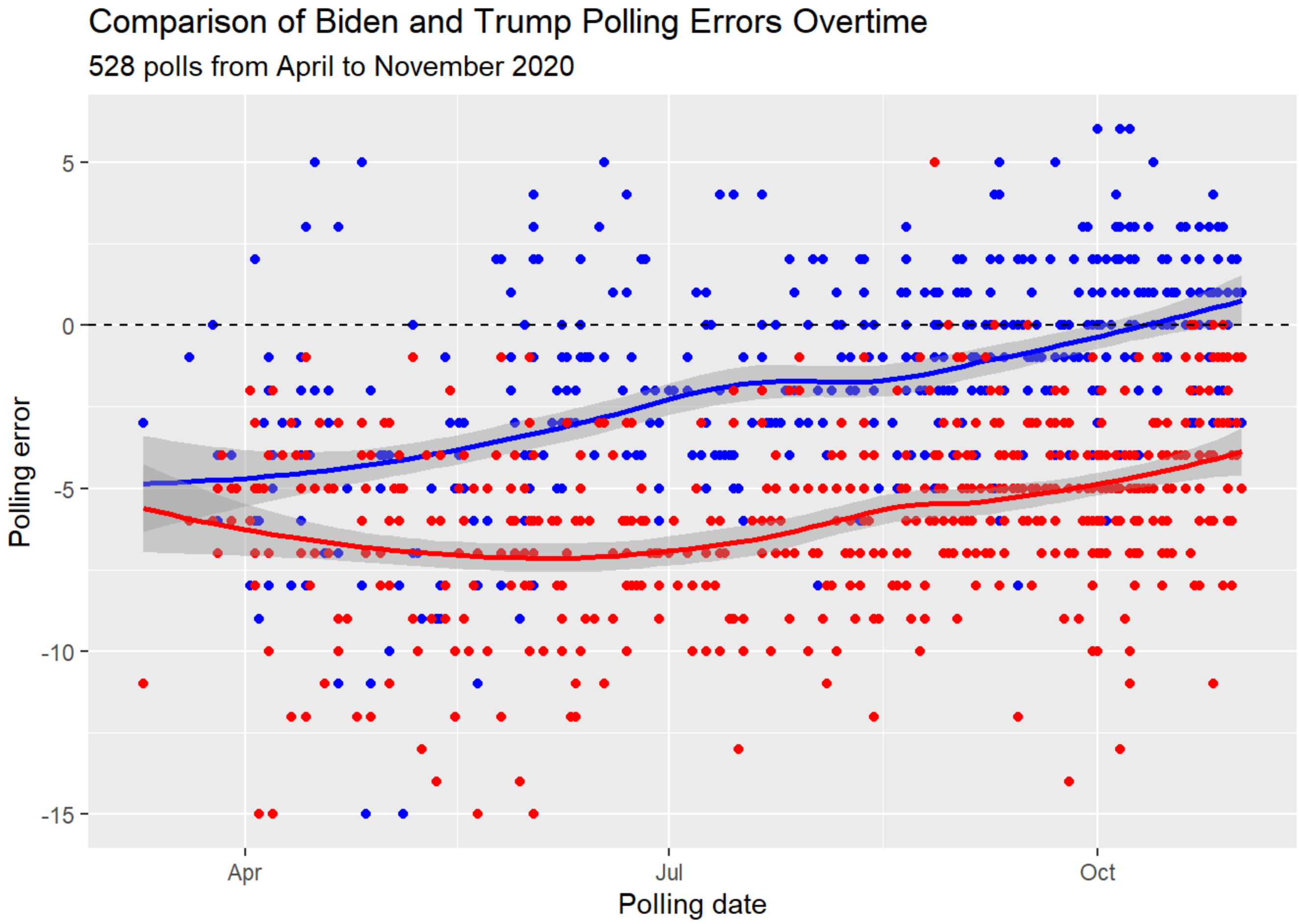
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Creating new variable: prediction error

```
# Create such that + ==> overestimate, - ==> underestimate
pres <- pres %>%
  mutate(bidenError = Biden - DemCertVote,
         trumpError = Trump - RepCertVote)

# Recreate plot
pres %>%
  ggplot(aes(x = StartDate_cleaned,
             y = bidenError)) +
  geom_point(color = 'blue') +
  geom_smooth(color = 'blue') +
  geom_point(color = 'red',
             aes(y = trumpError)) +
  geom_smooth(color = 'red',
              aes(y = trumpError)) +
  geom_hline(yintercept = 0,linetype = 'dashed') +
  labs(x = 'Polling date',
       y = 'Polling error',
       title = 'Comparison of Biden and Trump Polling Errors Overtime',
       subtitle = '528 polls from April to November 2020')
```

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



Introducing midterm exam data

```
fn <- read_rds("https://github.com/jbisbee1/ISP_Data_Science_2024/raw/main/data/fn_clean
ed_final.rds")

fn
```

```
## # A tibble: 957 × 24
##    placed mental_state eliminations assists revives accuracy  hits head_shots
##     <dbl> <chr>               <dbl>   <dbl>   <dbl>    <dbl> <dbl>      <dbl>
##  1     17 sober                   2       0       0   0.194     10          1
##  2     41 sober                   0       2       0   0.324     17          0
##  3     36 high                    3       0       0   0.337     38          0
##  4     28 high                    1       4       0   0.105     22          3
##  5      3 high                    3       2       1   0.622     49         18
##  6     15 high                    0       1       0   0.0582     4          3
##  7      9 high                    2       2       0   0.265     43          2
##  8     29 high                    3       2       0   0.272     14          3
##  9     11 high                    4       0       1   0.383     53         13
## 10      1 high                    1       2       0   0.328     27          0
## # ℹ 947 more rows
## # ℹ 16 more variables: distance_traveled <dbl>, materials_gathered <dbl>,
## #   materials_used <dbl>, damage_taken <dbl>, damage_to_players <dbl>,
## #   damage_to_structures <dbl>, won <dbl>, player <int>, gameId <int>,
## #   startTime <dttm>, sessionId <int>, lagSess <dbl>, delta <dbl>,
## #   startTime2 <dttm>, gameIdSession <int>, gameIdSession2 <dbl>
```

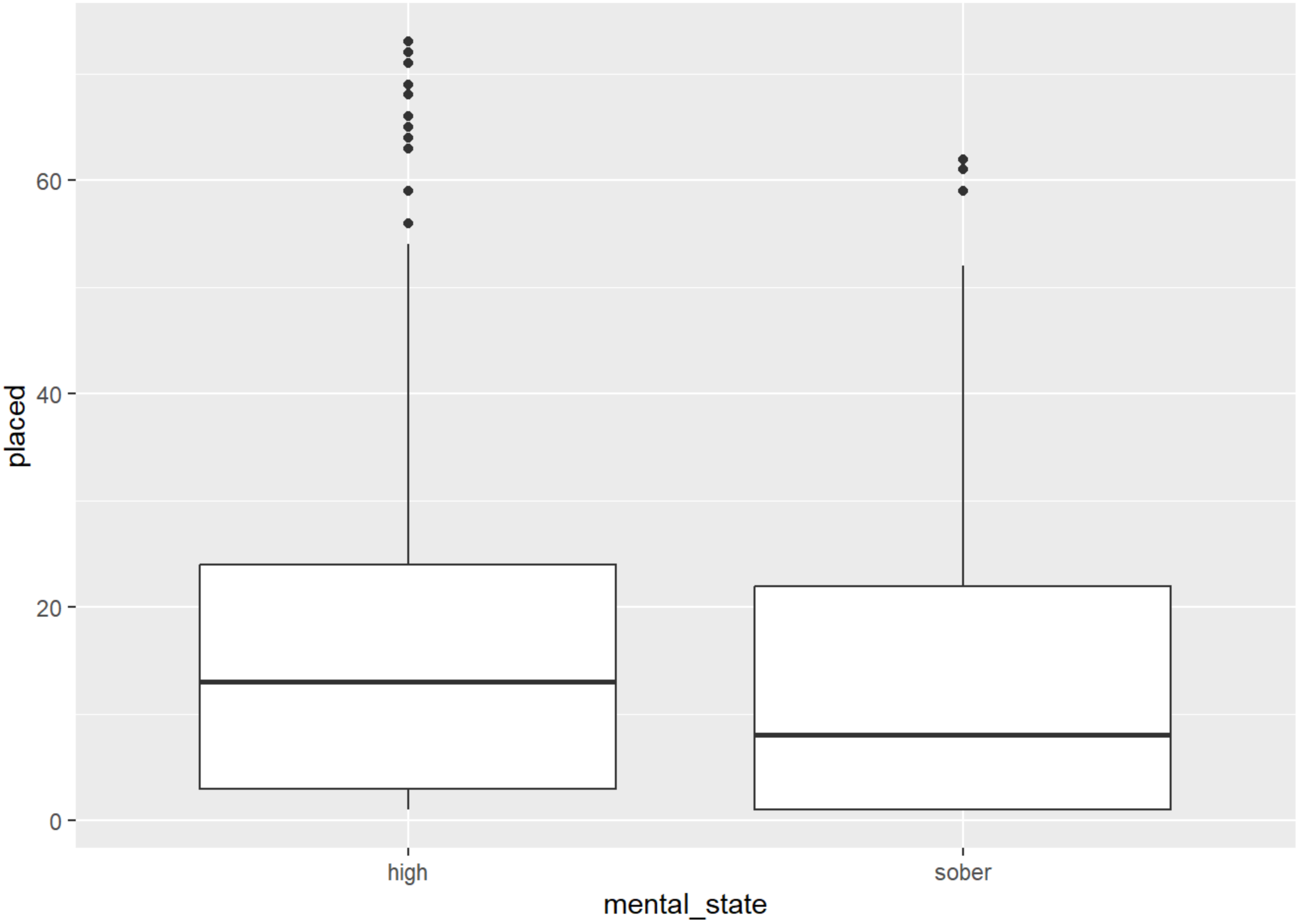
```
summary(fn)
```

```
##      placed      mental_state        eliminations      assists
##  Min.   : 1.00   Length:957         Min.   :0.000   Min.   :0.00
##  1st Qu.: 1.00   Class :character   1st Qu.:1.000   1st Qu.:0.00
##  Median :11.00   Mode  :character   Median :2.000   Median :1.00
##  Mean   :14.58                      Mean   :2.526   Mean   :1.51
##  3rd Qu.:24.00                      3rd Qu.:4.000   3rd Qu.:2.00
##  Max.   :73.00                      Max.   :9.000   Max.   :7.00
##
##     revives          accuracy           hits          head_shots
##  Min.   :0.0000   Min.   :0.0000   Min.   :  0.00   Min.   : 0.000
##  1st Qu.:0.0000   1st Qu.:0.1736   1st Qu.: 13.00   1st Qu.: 1.000
##  Median :0.0000   Median :0.2469   Median : 25.00   Median : 3.000
##  Mean   :0.4086   Mean   :0.2605   Mean   : 29.95   Mean   : 4.829
##  3rd Qu.:1.0000   3rd Qu.:0.3256   3rd Qu.: 39.00   3rd Qu.: 6.000
##  Max.   :4.0000   Max.   :0.9472   Max.   :118.00   Max.   :36.000
##
##  distance_traveled materials_gathered materials_used    damage_taken
##  Min.   :   0      Min.   :   0.0     Min.   :   0.0   Min.   :  0
##  1st Qu.: 357      1st Qu.:  52.0     1st Qu.:  16.0   1st Qu.:153
##  Median : 725      Median : 204.0     Median :  66.0   Median :217
##  Mean   :1140      Mean   : 399.6     Mean   : 132.5   Mean   :245
##  3rd Qu.:1516      3rd Qu.: 459.0     3rd Qu.: 148.0   3rd Qu.:322
##  Max.   :4766      Max.   :3098.0     Max.   :1788.0   Max.   :749
##
##  damage_to_players damage_to_structures      won             player
##  Min.   :   0.0    Min.   :    0        Min.   :0.0000   Min.   :-5
##  1st Qu.: 334.0    1st Qu.: 1001        1st Qu.:0.0000   1st Qu.:-3
##  Median : 499.0    Median : 2068        Median :0.0000   Median : 0
##  Mean   : 581.7    Mean   : 3217        Mean   :0.3041   Mean   : 0
##  3rd Qu.: 764.0    3rd Qu.: 4159        3rd Qu.:1.0000   3rd Qu.: 3
##  Max.   :1693.0    Max.   :19308        Max.   :1.0000   Max.   : 5
##
##      gameId     startTime                        sessionId        lagSess
##  Min.   : 1   Min.   :2020-04-10 16:46:06.26   Min.   :1.000   Min.   :1.000
##  1st Qu.:22   1st Qu.:2020-04-18 02:42:45.60   1st Qu.:1.000   1st Qu.:1.000
##  Median :44   Median :2020-04-25 10:29:43.91   Median :2.000   Median :2.000
##  Mean   :44   Mean   :2020-04-25 23:17:50.25   Mean   :2.042   Mean   :2.018
##  3rd Qu.:66   3rd Qu.:2020-05-03 11:39:01.86   3rd Qu.:3.000   3rd Qu.:3.000
##  Max.   :87   Max.   :2020-05-12 11:36:13.09   Max.   :4.000   Max.   :4.000
##
##      delta            startTime2                     gameIdSession
##  Min.   :   36.67   Min.   :2020-04-10 17:05:06.74   Min.   : 1.00
##  1st Qu.:  633.12   1st Qu.:2020-04-18 02:54:33.15   1st Qu.: 8.00
##  Median :  824.78   Median :2020-04-25 22:28:54.59   Median :15.00
##  Mean   : 2868.84   Mean   :2020-04-25 23:40:50.49   Mean   :15.66
##  3rd Qu.: 1061.13   3rd Qu.:2020-05-03 11:39:31.61   3rd Qu.:22.00
##  Max.   :86633.67   Max.   :2020-05-12 11:36:13.09   Max.   :44.00
##                     NA's   :1
##  gameIdSession2
##  Min.   :   1.0
##  1st Qu.:  64.0
##  Median : 225.0
##  Mean   : 346.8
##  3rd Qu.: 484.0
##  Max.   :1936.0
##
```

```
fn %>%
  count(mental_state)
```

```
## # A tibble: 2 × 2
##   mental_state     n
##   <chr>        <int>
## 1 high           462
## 2 sober          495
```

```
View(fn)

fn %>%
  ggplot(aes(x = mental_state,
             y = placed)) +
  geom_boxplot()
```

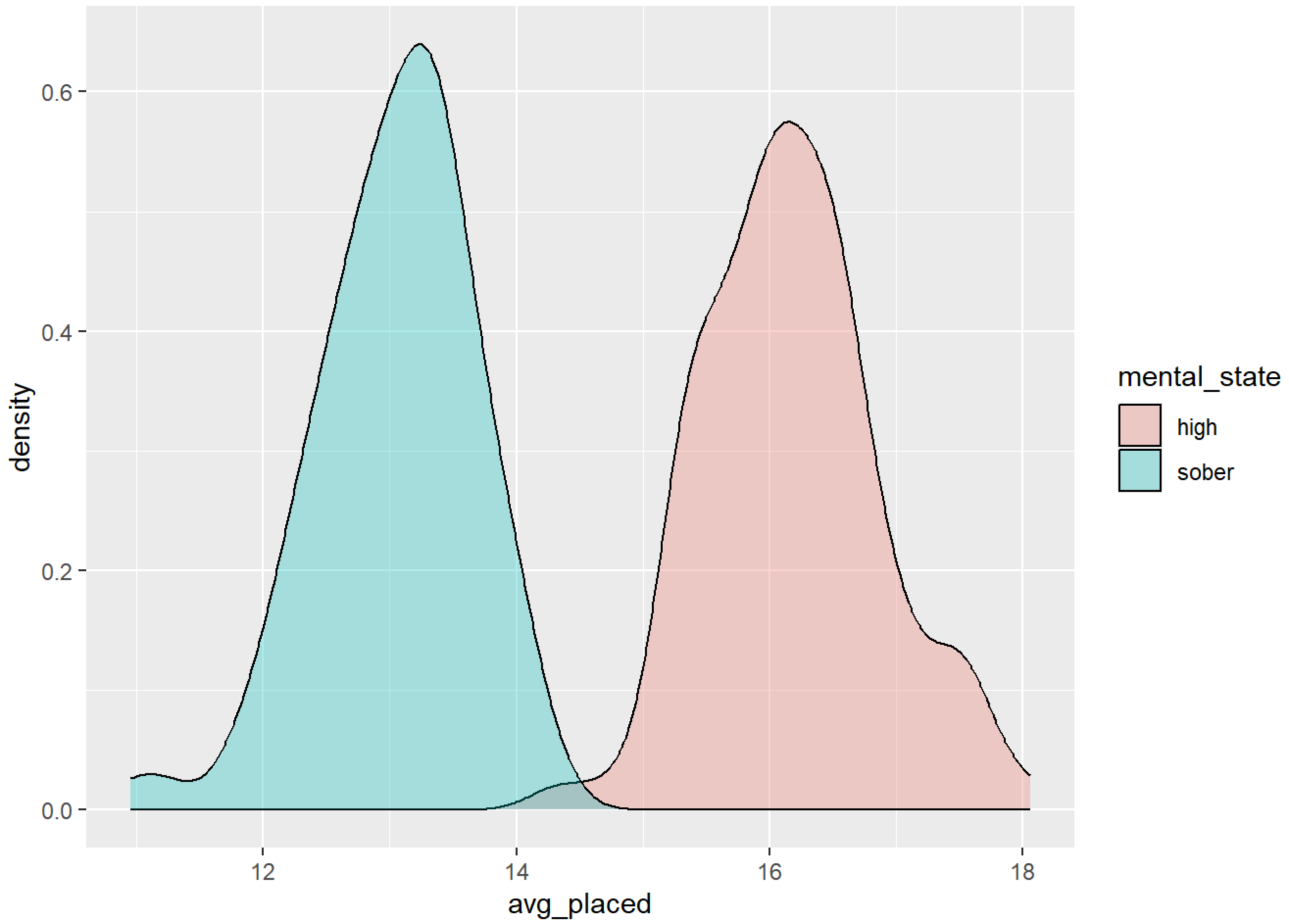


Bootstrap

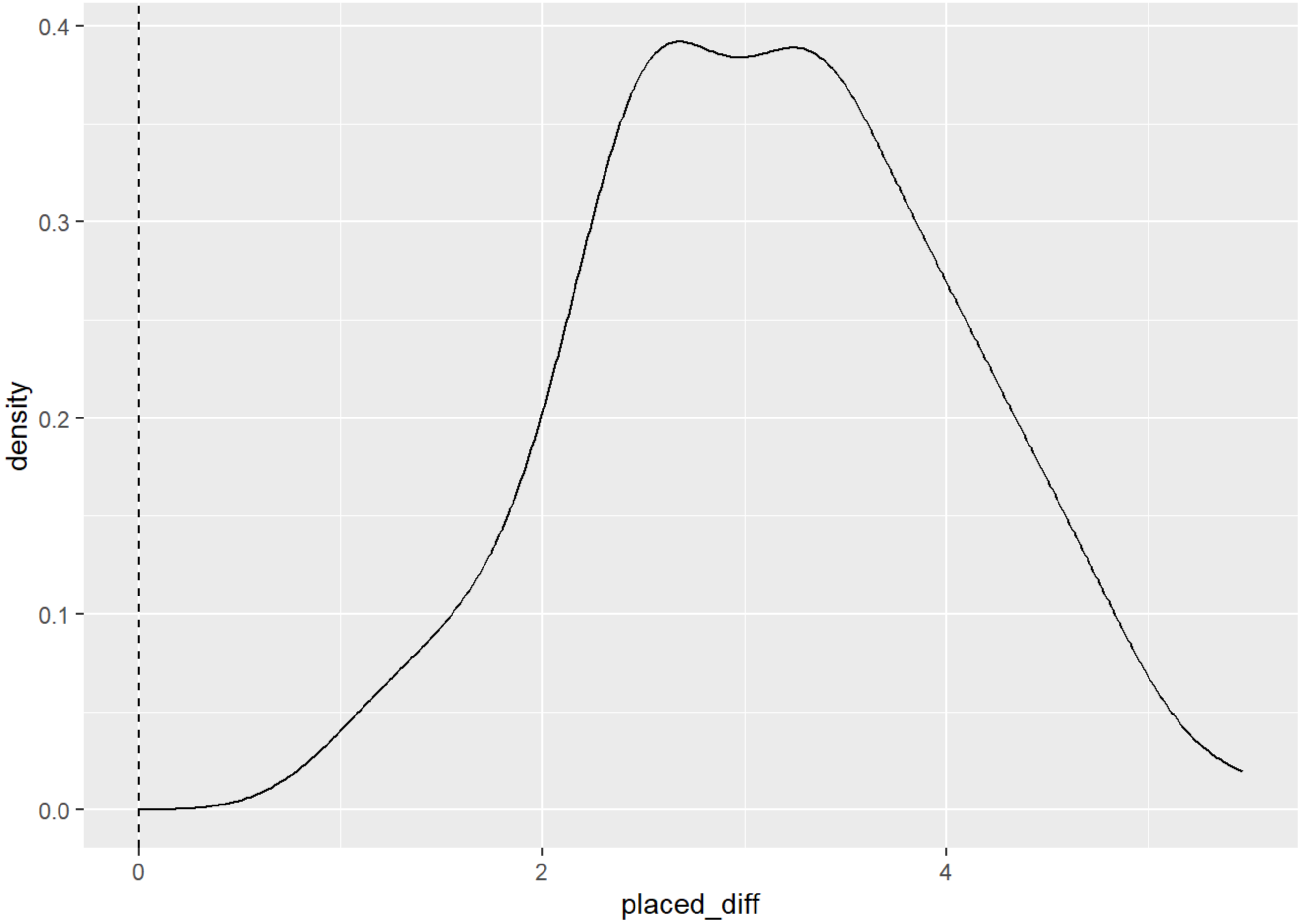
```
set.seed(123)
bsRes <- NULL
for(i in 1:100) {
  simulation <- fn %>%
    sample_n(size = nrow(.),
             replace = T)

  answer <- simulation %>%
    group_by(mental_state) %>%
    summarise(avg_placed = mean(placed)) %>%
    mutate(sim_number = i)

  bsRes <- bsRes %>%
    bind_rows(answer)
}

# Result
bsRes_wide <- bsRes %>%
  pivot_wider(names_from = "mental_state",
              values_from = "avg_placed")

bsRes_wide %>%
  mutate(sober_better = ifelse(sober < high,
                               1,
                               0)) %>%
  summarise(confidence = mean(sober_better))
```

```
## # A tibble: 1 × 1
##   confidence
##        <dbl>
## 1          1
```

```
# Visualize
bsRes %>%
  ggplot(aes(x = avg_placed,
             fill = mental_state)) +
  geom_density(alpha = .3)
```



```
bsRes_wide %>%
  mutate(placed_diff = high - sober) %>%
  ggplot(aes(x = placed_diff)) +
  geom_density() +
  geom_vline(xintercept = 0,linetype = 'dashed')
```


0.4
0.3
0.2
0.1
0.0
density
0
2
4
placed_diff