Lecture 6 Notes

2024-07-09

RQ: Do rookies turn the ball over more?

```
require(tidyverse)
```

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

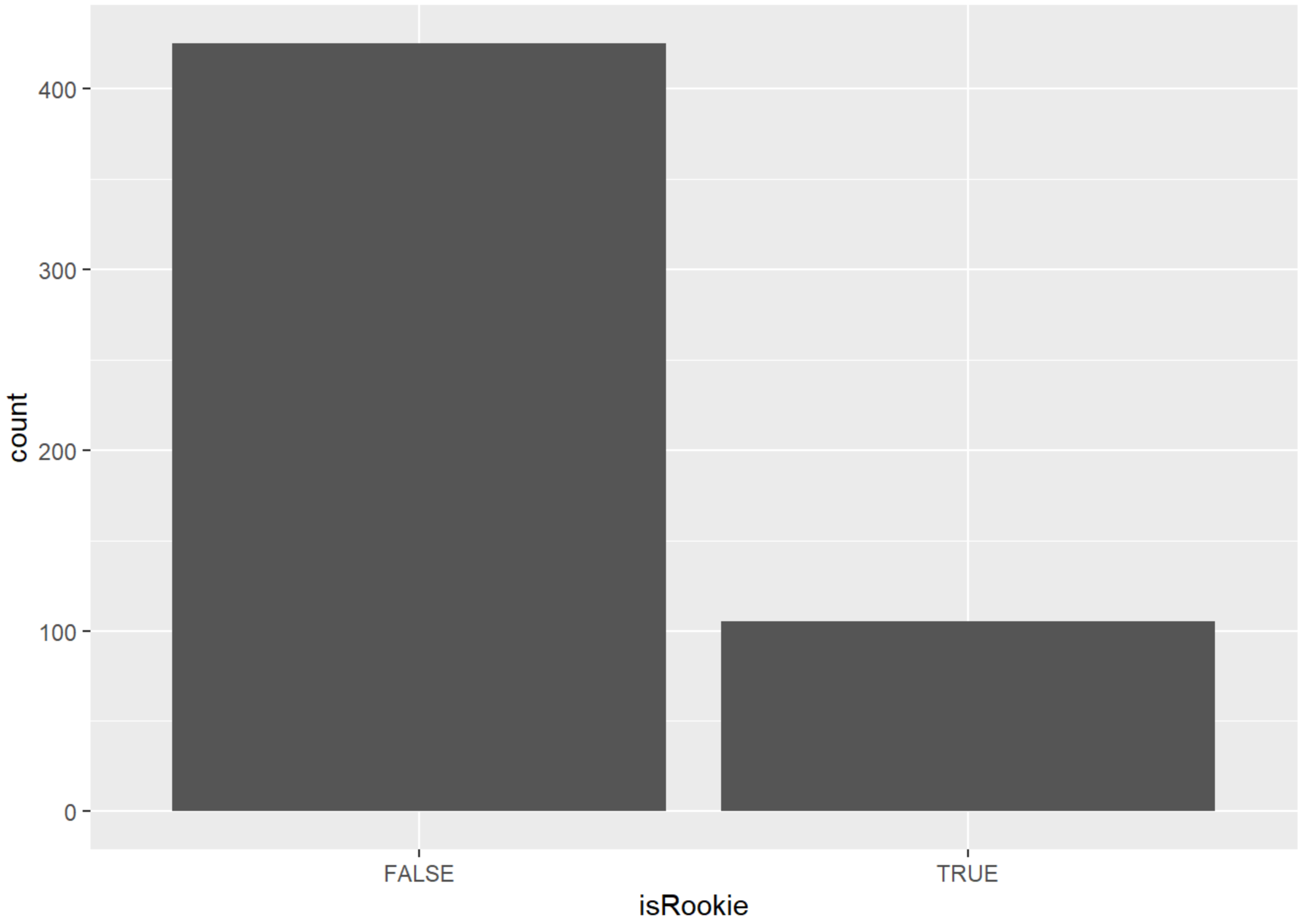
```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
## ── Conflicts ──────────────────────────────────────────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

```
nba <- read_rds('https://github.com/jbisbee1/ISP_Data_Science_2024/raw/main/data/nba_pla
yers_2018.Rds')

glimpse(nba %>% select(tov,isRookie))
```

```
## Rows: 530
## Columns: 2
## $ tov      <dbl> 144, 4, 135, 14, 121, 8, 33, 6, 28, 2, 72, 268, 58, 23, 103, …
## $ isRookie <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, TRUE, TRUE, TRUE, TR…
```

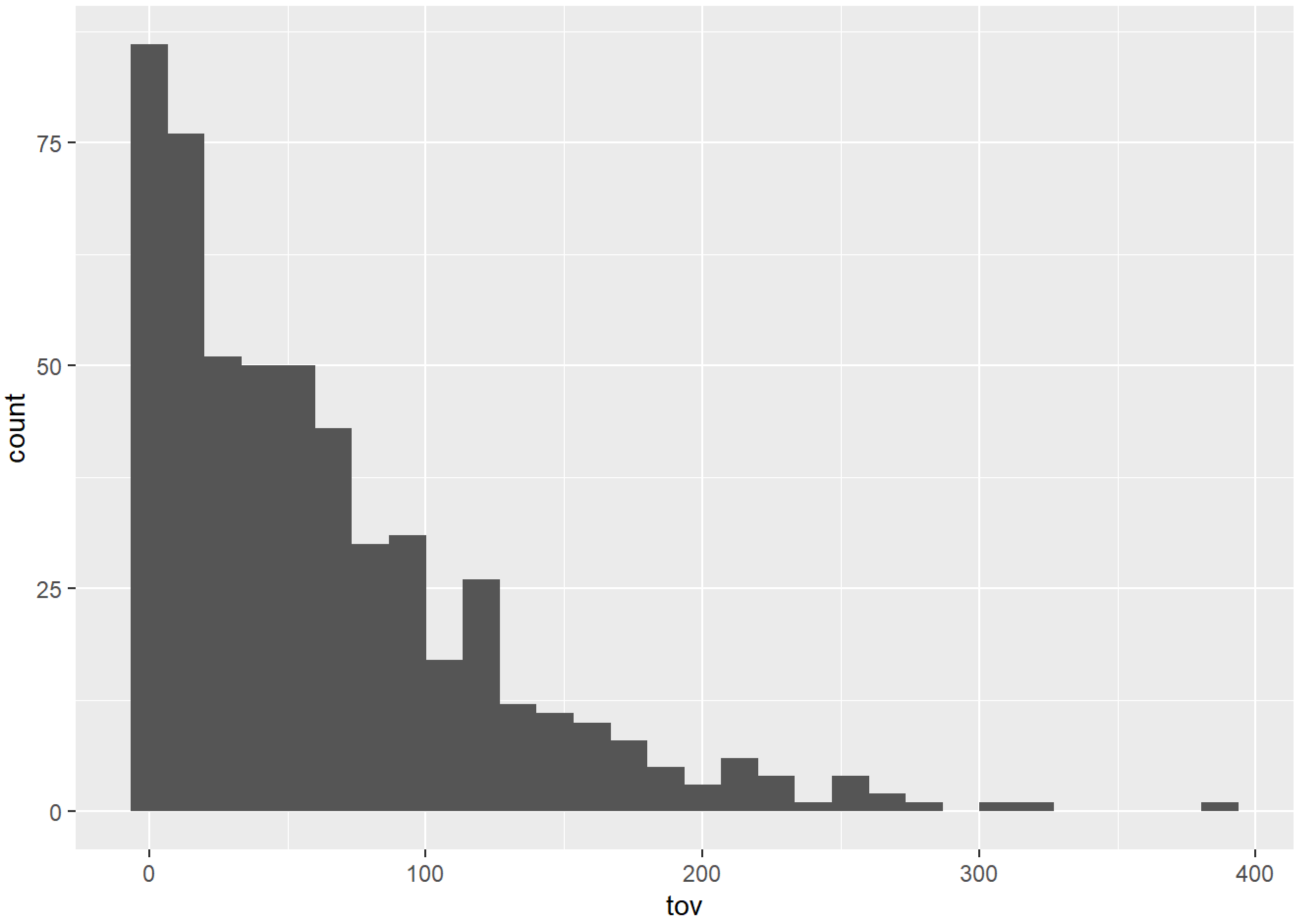
Visualize both X and Y variables

```
# X
nba %>%
  ggplot(aes(x = isRookie)) +
  geom_bar()
```



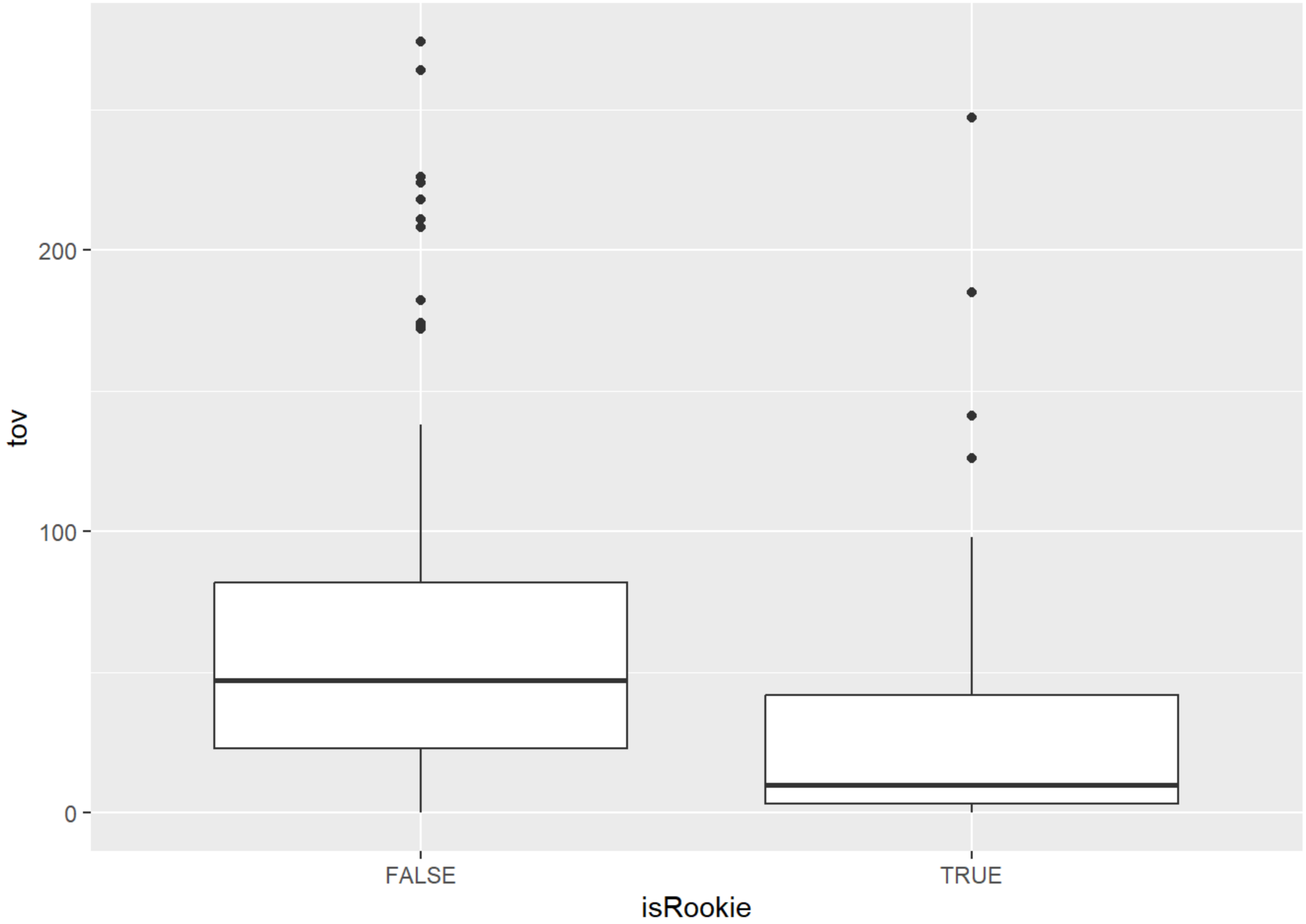
```
# Y
nba %>%
  ggplot(aes(x = tov)) +
  geom_histogram()
```

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



Bootstrap sampling

```
set.seed(123)
simulated_season <- nba %>%
  select(namePlayer,isRookie,tov) %>%
  sample_n(size = 200,
           replace = T)

simulated_season %>%
  ggplot(aes(x = isRookie,
             y = tov)) +
  geom_boxplot()
```



```
simulated_season %>%
  group_by(isRookie) %>%
  summarise(avg_tov = mean(tov,na.rm=T))
```

```
## # A tibble: 2 × 2
##   isRookie avg_tov
##   <lgl>      <dbl>
## 1 FALSE       64.5
## 2 TRUE        36.7
```

New function: for() loop

```
set.seed(123)
bootstrap_result <- NULL #instantiate an empty object
for(i in 1:100) {
  # Simulate a new season
  simulated_season <- nba %>%
  select(namePlayer,isRookie,tov) %>%
  sample_n(size = 200,
           replace = T)

  # Answer research question
  answer <- simulated_season %>%
    group_by(isRookie) %>%
    summarise(avg_tov = mean(tov,na.rm=T))

  answer <- answer %>%
    mutate(simulation_number = i)

  # Save result to bootstrap_result object
  bootstrap_result <- bootstrap_result %>%
    bind_rows(answer)
}

bootstrap_result
```

```
## # A tibble: 200 × 3
##    isRookie avg_tov simulation_number
##    <lgl>      <dbl>             <int>
##  1 FALSE       64.5                 1
##  2 TRUE        36.7                 1
##  3 FALSE       75.5                 2
##  4 TRUE        39.2                 2
##  5 FALSE       69                   3
##  6 TRUE        30.4                 3
##  7 FALSE       62.8                 4
##  8 TRUE        30.6                 4
##  9 FALSE       66.9                 5
## 10 TRUE        27.2                 5
## # ℹ 190 more rows
```

Calculate confidence / certainty

- Idea: confidence = # of sim realities that support research question / # of sim realities

```
bootstrap_result
```

```
## # A tibble: 200 × 3
##    isRookie avg_tov simulation_number
##    <lgl>      <dbl>             <int>
##  1 FALSE       64.5                 1
##  2 TRUE        36.7                 1
##  3 FALSE       75.5                 2
##  4 TRUE        39.2                 2
##  5 FALSE       69                   3
##  6 TRUE        30.4                 3
##  7 FALSE       62.8                 4
##  8 TRUE        30.6                 4
##  9 FALSE       66.9                 5
## 10 TRUE        27.2                 5
## # ℹ 190 more rows
```

```
# New function: pivot_wider()
final_answer <- bootstrap_result %>%
  pivot_wider(names_from = "isRookie",
              values_from = "avg_tov",
              names_prefix = "rookie") %>%
  mutate(rookie_better = ifelse(rookieFALSE > rookieTRUE,
                                1,
                                0))

final_answer %>%
  count(rookie_better)
```

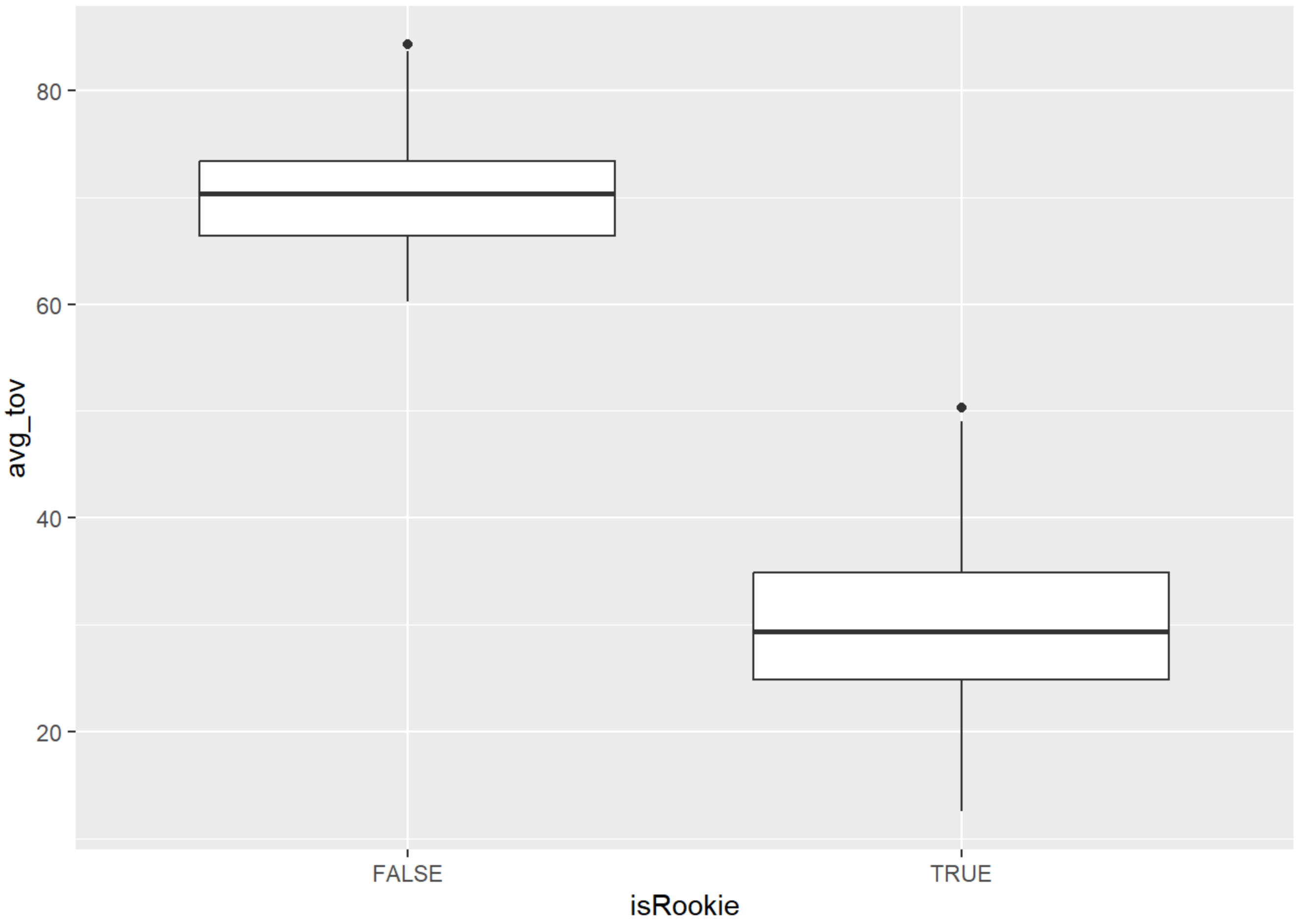
```
## # A tibble: 1 × 2
##   rookie_better     n
##           <dbl> <int>
## 1             1   100
```

```
final_answer %>%
  count(rookie_better) %>%
  mutate(confidence = prop.table(n))
```

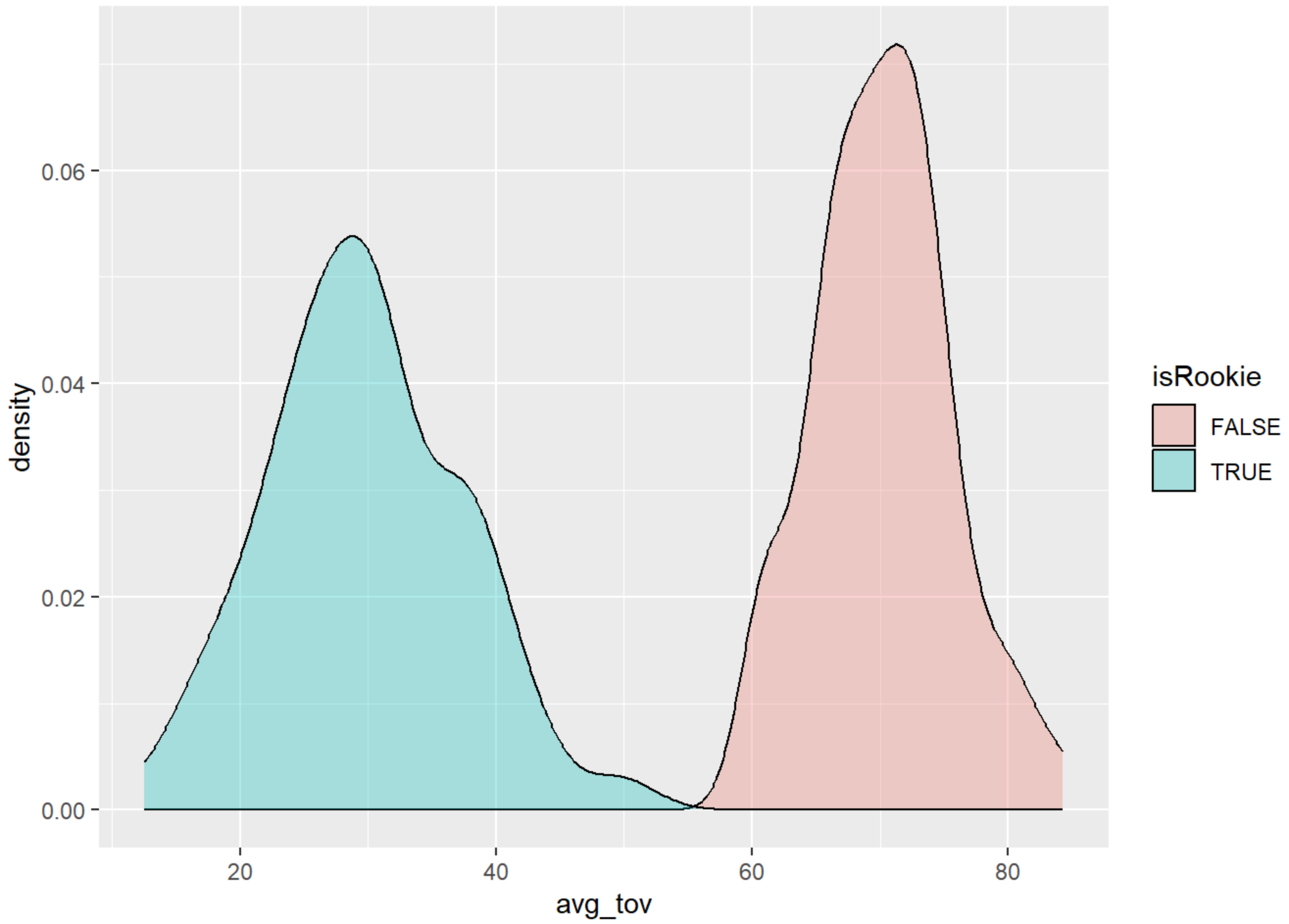
```
## # A tibble: 1 × 3
##   rookie_better     n confidence
##           <dbl> <int>      <dbl>
## 1             1   100          1
```

Visualizing Bootstrapped Results

```
bootstrap_result %>%
  ggplot(aes(x = isRookie,
             y = avg_tov)) +
  geom_boxplot()
```



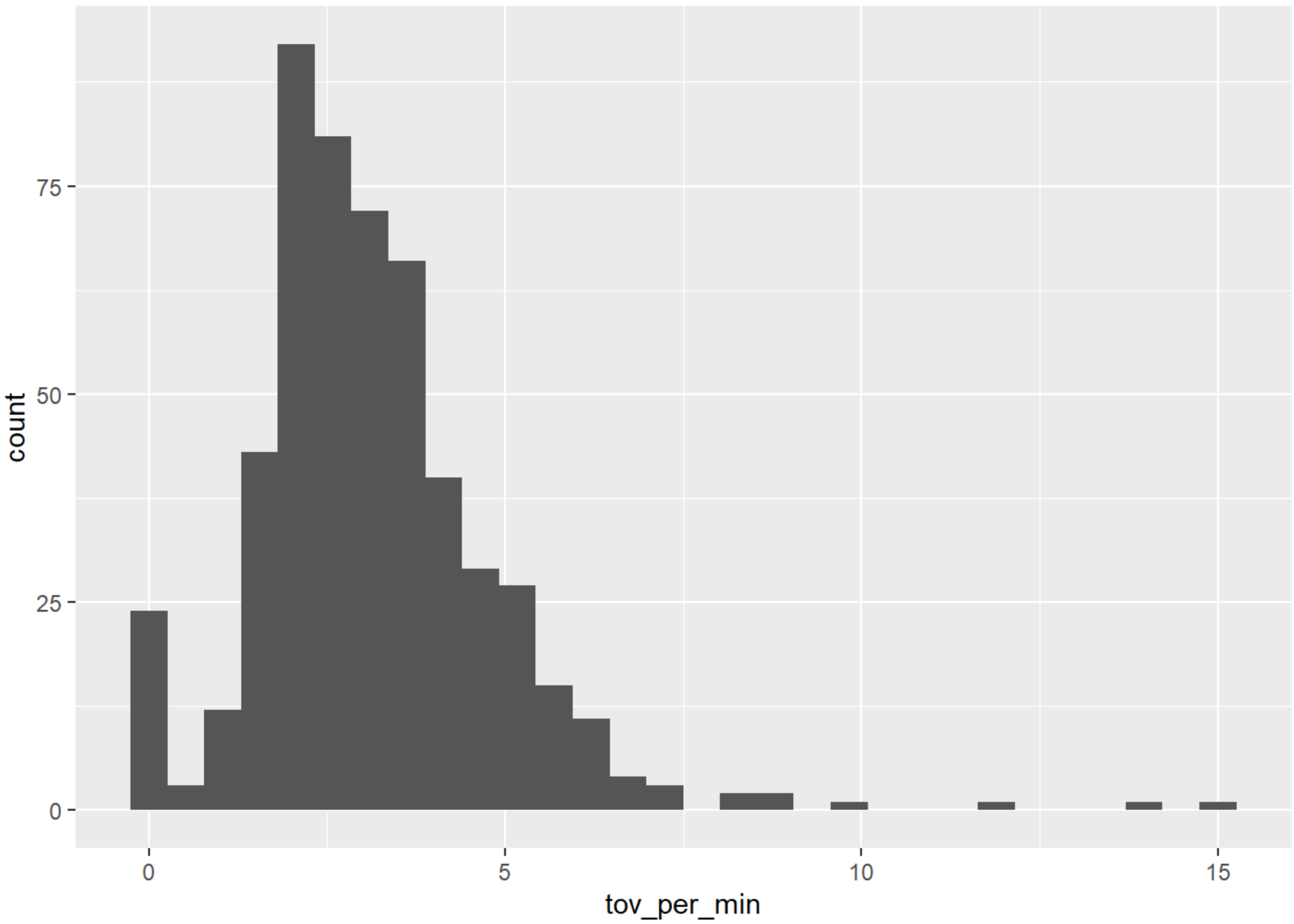
```
# geom_density()
bootstrap_result %>%
  ggplot(aes(x = avg_tov,
             fill = isRookie)) +
  geom_density(alpha = .3)
```



Our methods are right but our MEASURE is wrong

```
nba <- nba %>%
  mutate(tov_per_min = tov*60 / minutes)

nba %>%
  ggplot(aes(x = tov_per_min)) +
  geom_histogram()
```

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


count
75
50
25
0
0
5
10
15
tov_per_min

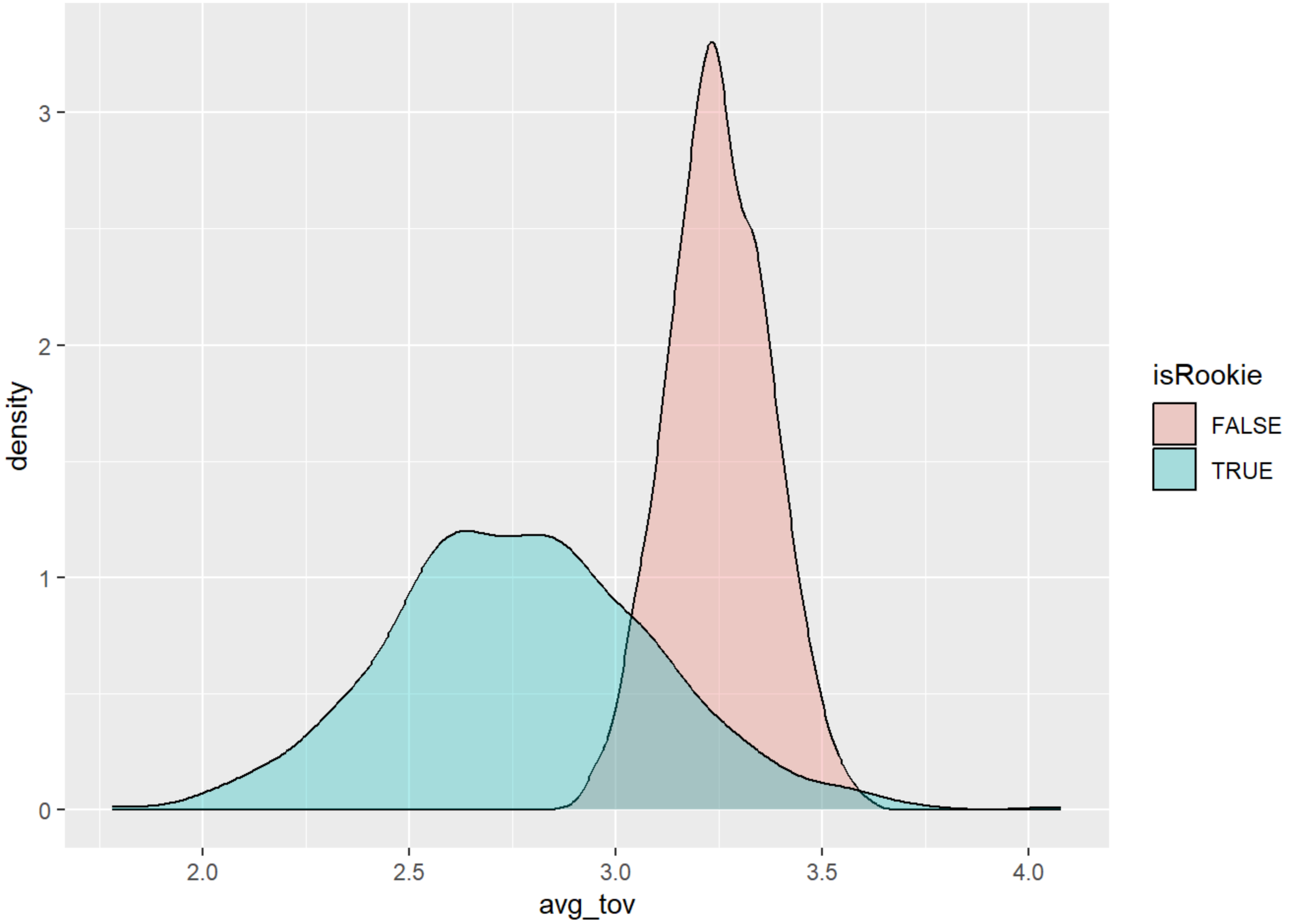
```
# Redo analysis
set.seed(123)
bootstrap_result <- NULL #instantiate an empty object
for(i in 1:1000) {
  # Simulate a new season
  simulated_season <- nba %>%
  select(namePlayer,isRookie,tov_per_min) %>%
  sample_n(size = 200,
           replace = T)

  # Answer research question
  answer <- simulated_season %>%
    group_by(isRookie) %>%
    summarise(avg_tov = mean(tov_per_min,na.rm=T))

  answer <- answer %>%
    mutate(simulation_number = i)

  # Save result to bootstrap_result object
  bootstrap_result <- bootstrap_result %>%
    bind_rows(answer)
}

final_answer <- bootstrap_result %>%
  pivot_wider(names_from = "isRookie",
              values_from = "avg_tov",
              names_prefix = "rookie") %>%
  mutate(rookie_better = ifelse(rookieFALSE > rookieTRUE,
                                1,
                                0))

final_answer %>%
  count(rookie_better)
```

```
## # A tibble: 2 × 2
##   rookie_better     n
##           <dbl> <int>
## 1             0    83
## 2             1   917
```

```
final_answer %>%
  count(rookie_better) %>%
  mutate(confidence = prop.table(n))
```

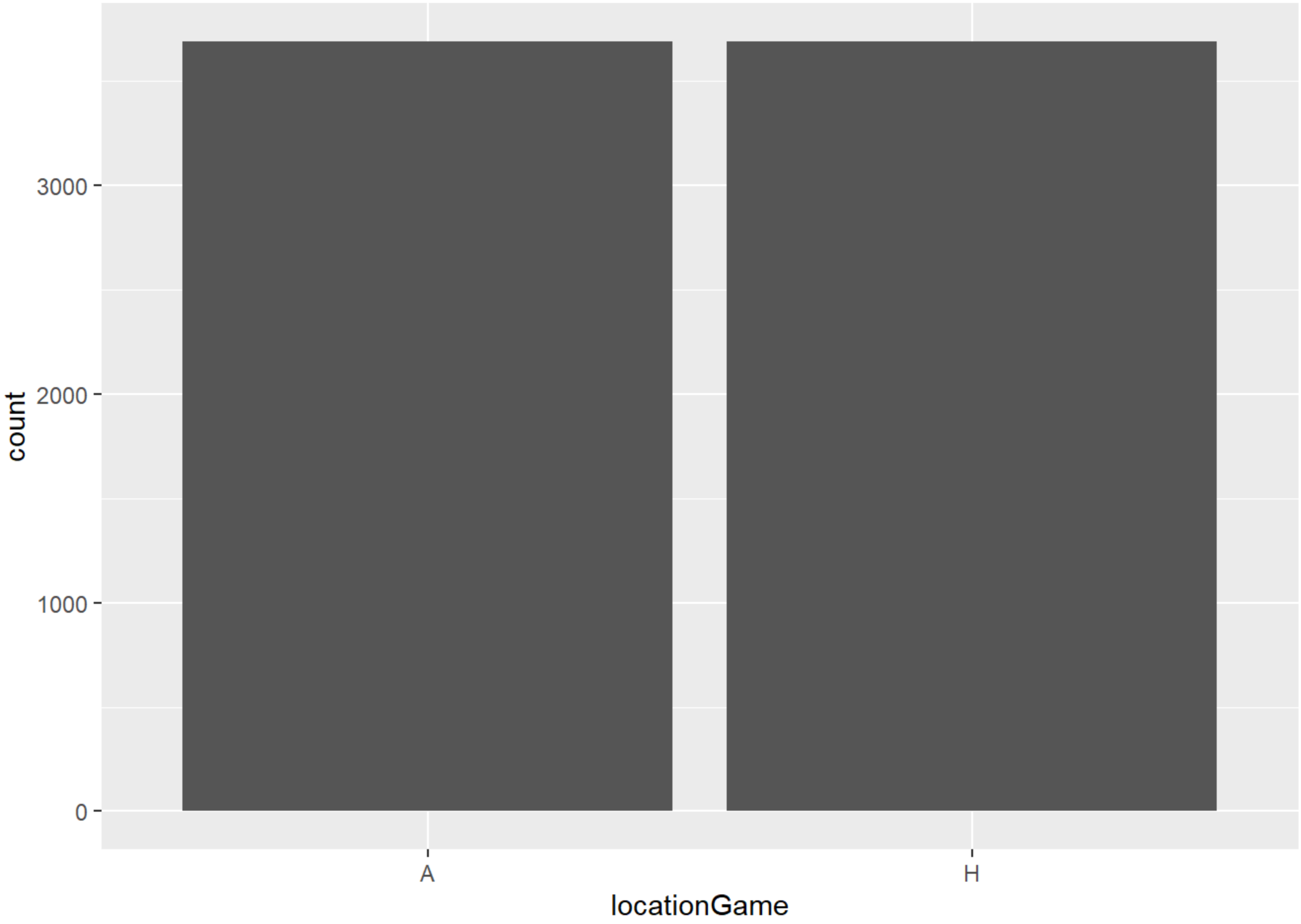
```
## # A tibble: 2 × 3
##   rookie_better     n confidence
##           <dbl> <int>      <dbl>
## 1             0    83      0.083
## 2             1   917      0.917
```

```
bootstrap_result %>%
  ggplot(aes(x = avg_tov,
             fill = isRookie)) +
  geom_density(alpha = .3)
```



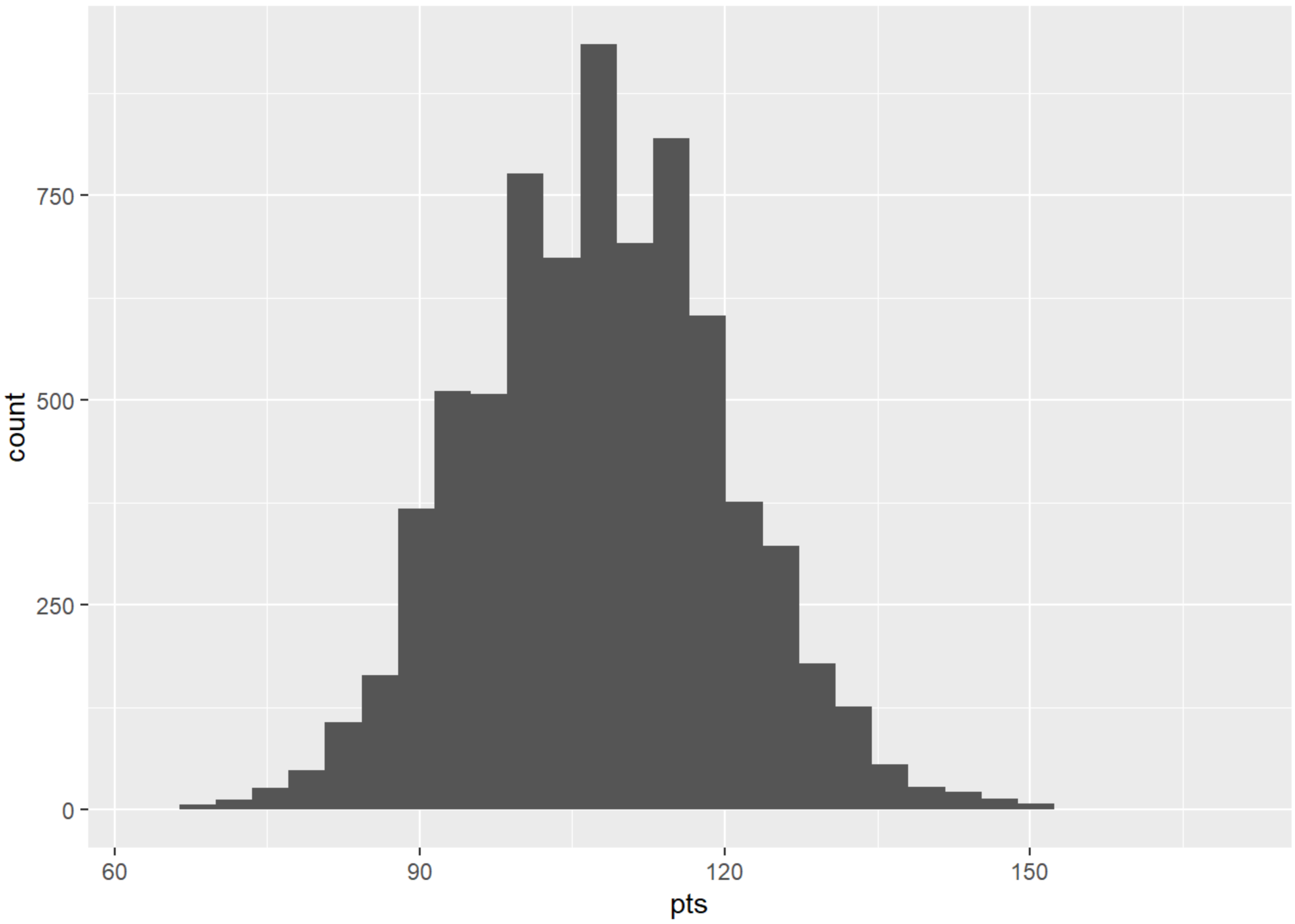
New data: game_summary.Rds

```
games <- read_rds('https://github.com/jbisbee1/ISP_Data_Science_2024/raw/main/data/game_
summary.Rds')

View(games)

# X
games %>%
  ggplot(aes(x = locationGame)) +
  geom_bar()
```



```
#Y
games %>%
  ggplot(aes(x = pts)) +
  geom_histogram()
```

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

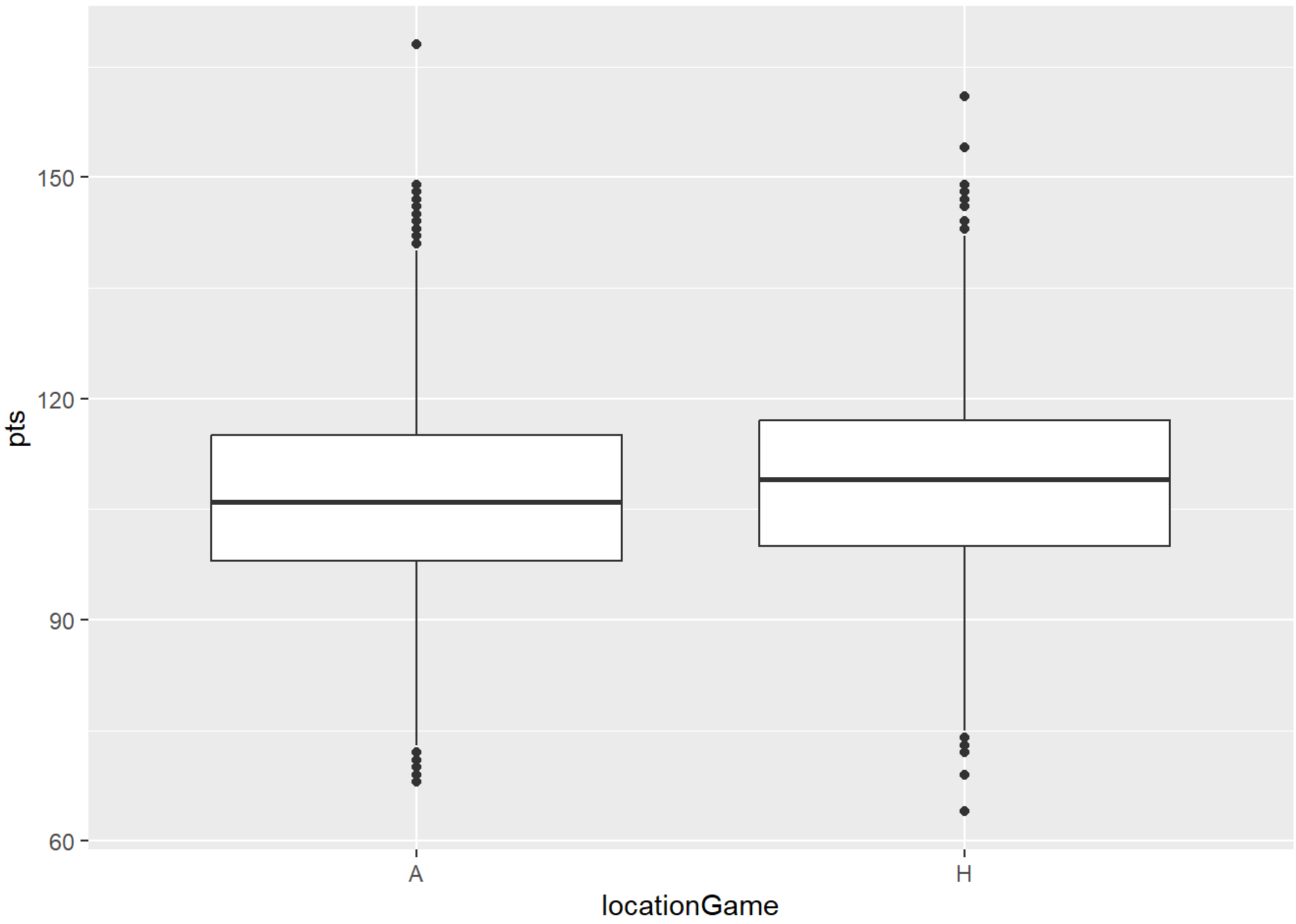


```
summary(games$pts)
```

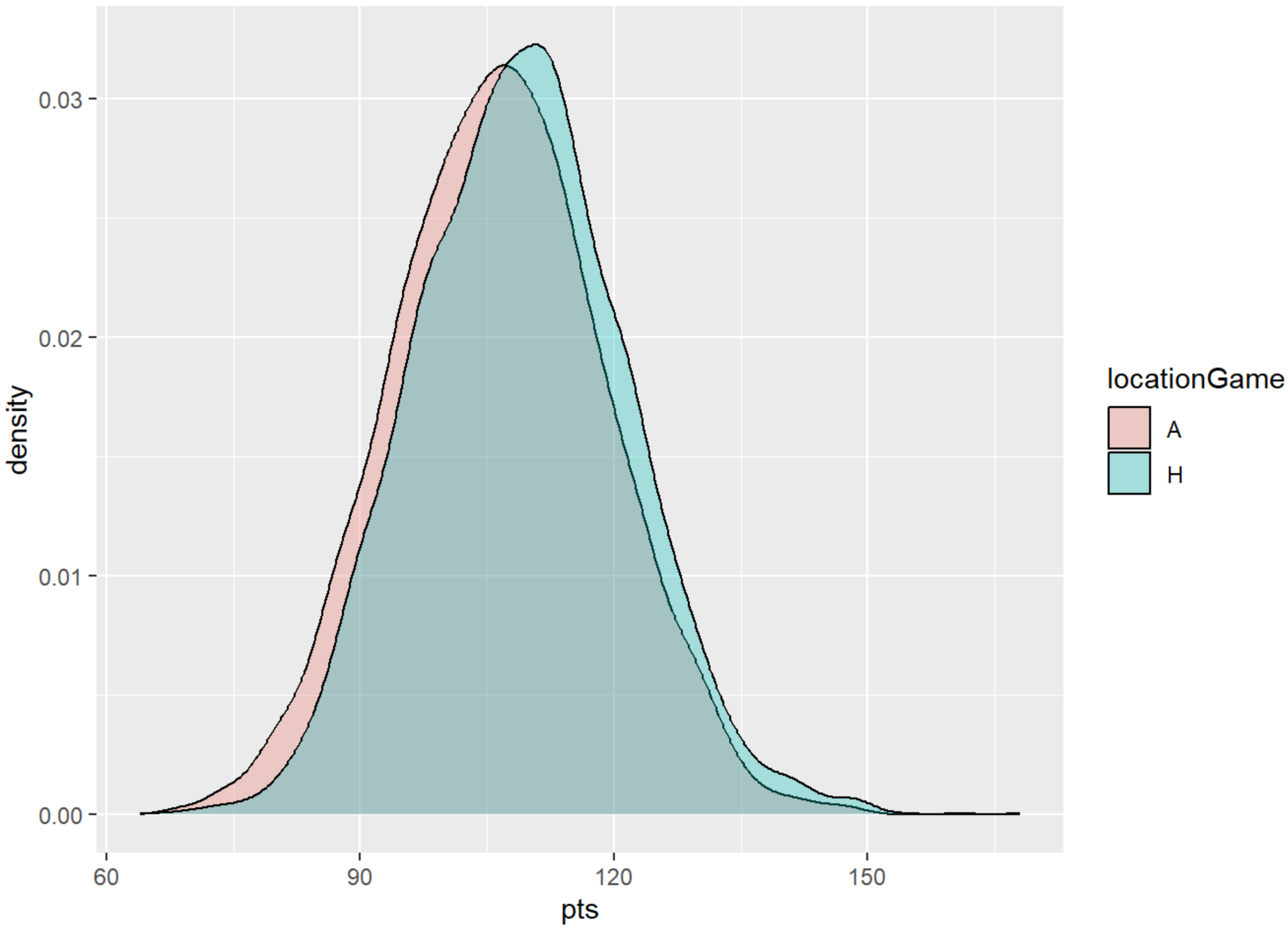
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    64.0    99.0   108.0   107.7   116.0   168.0 
```

Multivariate Visualization

```
# Boxplot
games %>%
  ggplot(aes(x = locationGame,
             y = pts)) +
  geom_boxplot()
```



```
# Density + fill
games %>%
  ggplot(aes(x = pts,
             fill = locationGame)) +
  geom_density(alpha = .3)
```



Confidence calculation via bootstrap

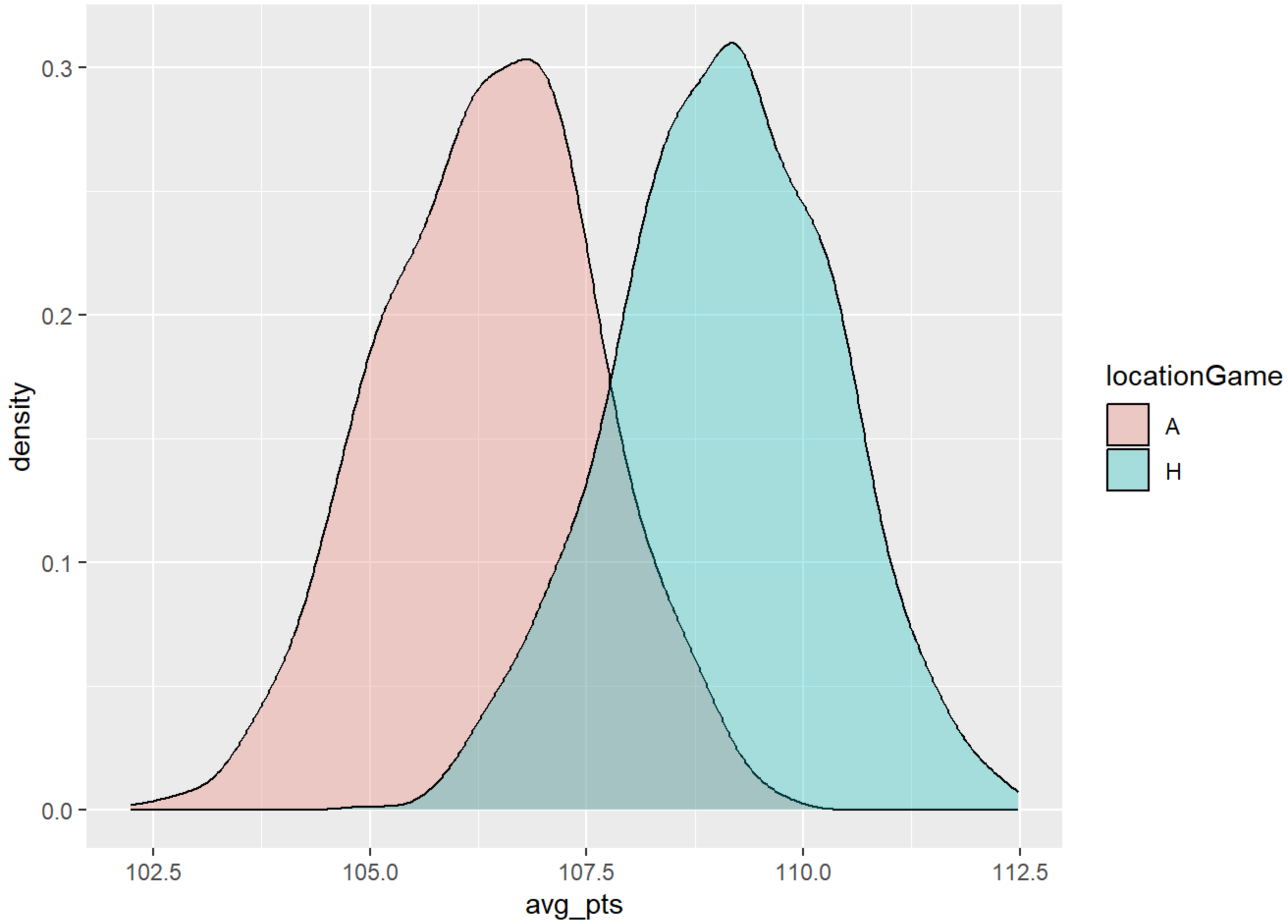
```
set.seed(123)
bootstrap_result <- NULL #instantiate an empty object
for(i in 1:1000) {
  # Simulate a new season
  simulated_season <- games %>%
  select(locationGame,pts) %>%
  sample_n(size = 200,
           replace = T)

  # Answer research question
  answer <- simulated_season %>%
    group_by(locationGame) %>%
    summarise(avg_pts = mean(pts,na.rm=T))

  answer <- answer %>%
    mutate(simulation_number = i)

  # Save result to bootstrap_result object
  bootstrap_result <- bootstrap_result %>%
    bind_rows(answer)
}

bootstrap_result
```

```
## # A tibble: 2,000 × 3
##    locationGame avg_pts simulation_number
##    <chr>          <dbl>             <int>
##  1 A               108.                 1
##  2 H               109.                 1
##  3 A               105.                 2
##  4 H               109.                 2
##  5 A               108.                 3
##  6 H               111.                 3
##  7 A               105.                 4
##  8 H               108.                 4
##  9 A               105.                 5
## 10 H               112.                 5
## # ℹ 1,990 more rows
```

```
final_answer <- bootstrap_result %>%
  pivot_wider(names_from = "locationGame",
              values_from = "avg_pts",
              names_prefix = "location_") %>%
  mutate(home_court_advantage = ifelse(location_H > location_A,
                                1,
                                0))

final_answer %>%
  count(home_court_advantage) %>%
  mutate(confidence = prop.table(n))
```

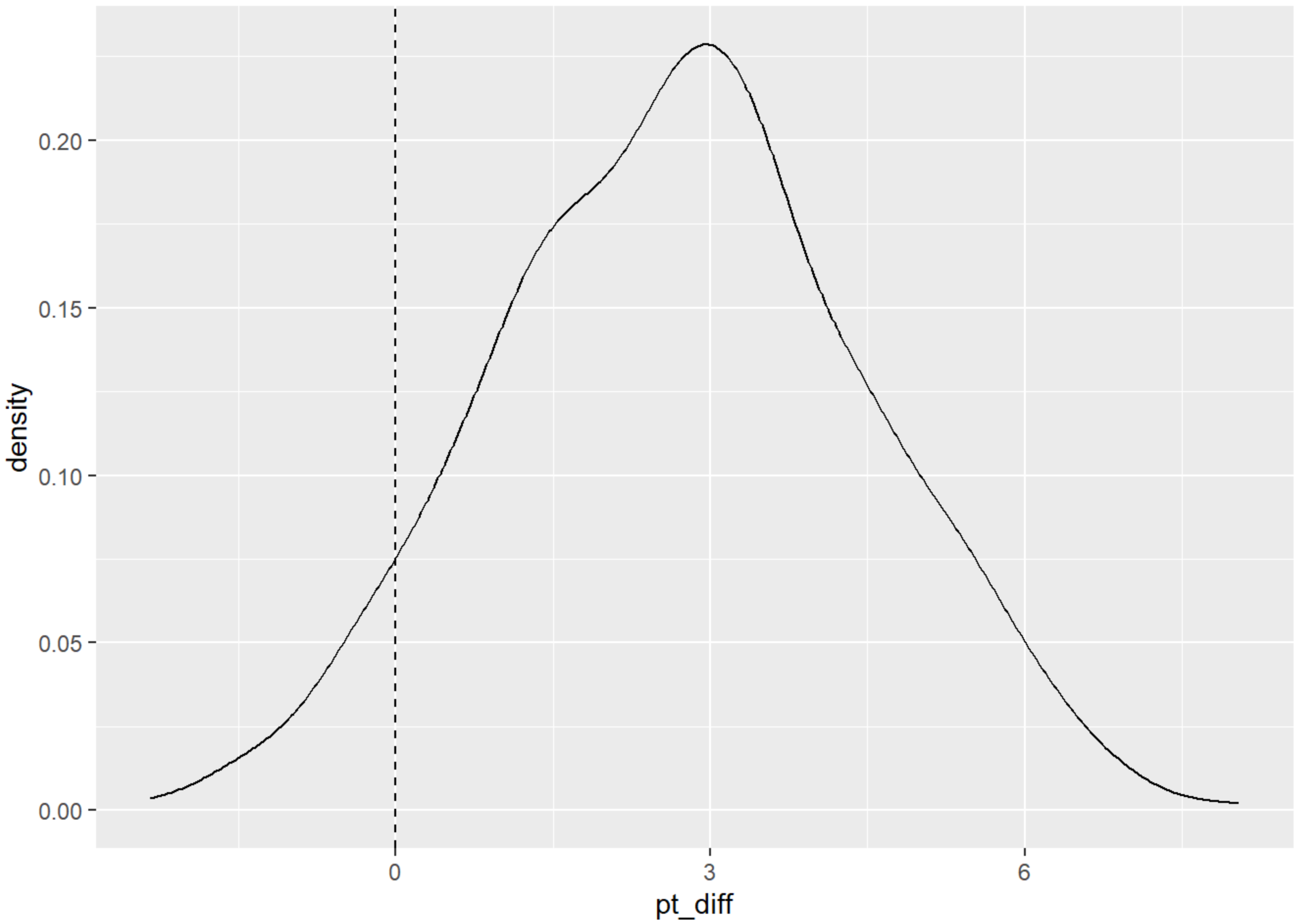
```
## # A tibble: 2 × 3
##   home_court_advantage     n confidence
##                  <dbl> <int>      <dbl>
## 1                    0    68      0.068
## 2                    1   932      0.932
```

```
bootstrap_result %>%
  ggplot(aes(x = avg_pts,
             fill = locationGame)) +
  geom_density(alpha = .3)
```



One final visualization of bootstrapped results

```
final_answer %>%
  mutate(pt_diff = location_H - location_A) %>%
  ggplot(aes(x = pt_diff)) +
  geom_density() +
  geom_vline(xintercept = 0,linetype = 'dashed')
```


0.20
0.15
0.10
0.05
0.00
density
0
3
6
pt_diff