Uncertainty Part 2

Sports Analytic Mania

Prof. Bisbee

Vanderbilt University

Slides Updated: 2024-01-09

Agenda

1. Uncertainty
2. More NBA data
3. Bootstrap Sampling

Sports Analytics

- Previously, we looked at players
  - Specifically, `isRookie` and `pts`
  - But could try **many** other ideas
- Useful if we want a job scouting talent
- But what if we want to advise actual games?
  - **Game Data**!

Other NBA Data

- Load the `game_summary.Rds` data

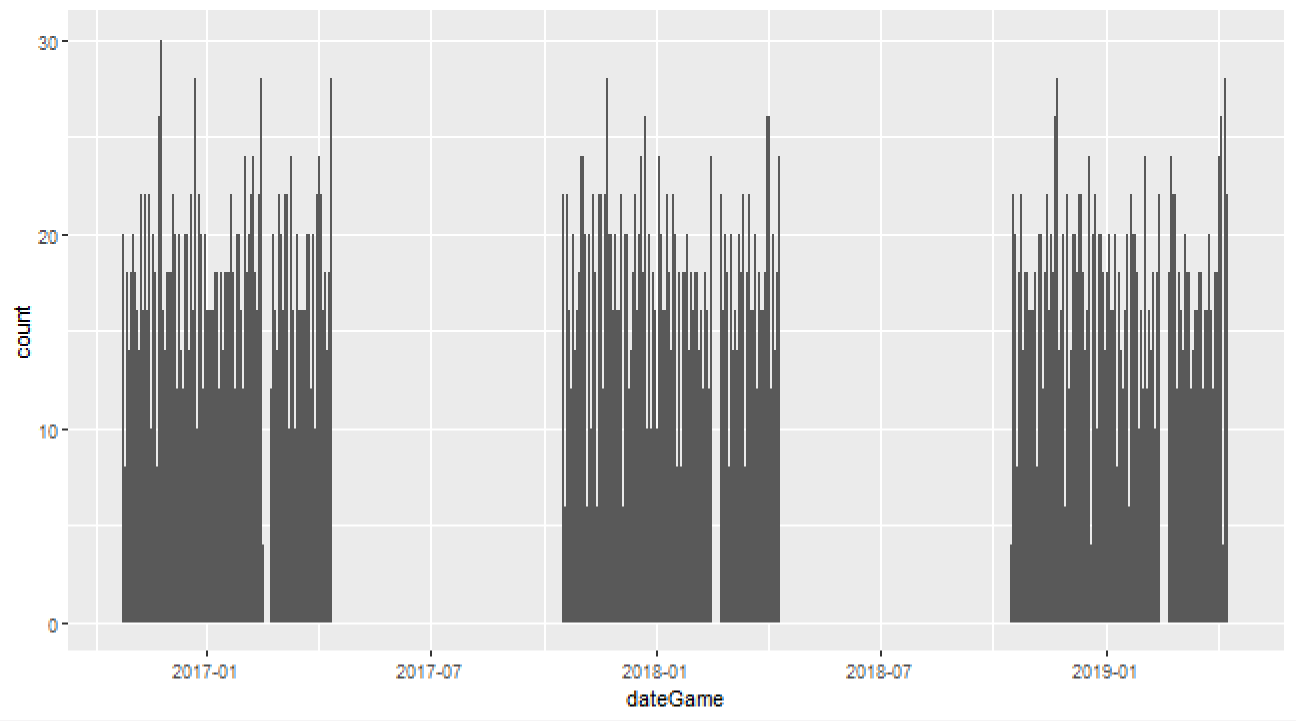
```
require(tidyverse)
gms <-
read_rds('https://github.com/jbisbee1/DS1000_S2024/raw/main/data/game_s
gms
```

```
## # A tibble: 7,380 × 16
##     idGame yearSeason dateGame   idTeam nameTeam locationGame
##      <dbl>      <int> <date>      <dbl> <chr>    <chr>
##  1 2.16e7       2017 2016-10-25 1.61e9 Clevela… H
##  2 2.16e7       2017 2016-10-25 1.61e9 New Yor… A
##  3 2.16e7       2017 2016-10-25 1.61e9 Portlan… H
##  4 2.16e7       2017 2016-10-25 1.61e9 Utah Ja… A
##  5 2.16e7       2017 2016-10-25 1.61e9 Golden … H
##  6 2.16e7       2017 2016-10-25 1.61e9 San Ant… A
##  7 2.16e7       2017 2016-10-26 1.61e9 Miami H… A
##  8 2.16e7       2017 2016-10-26 1.61e9 Orlando… H
##  9 2.16e7       2017 2016-10-26 1.61e9 Dallas … A
## 10 2.16e7       2017 2016-10-26 1.61e9 Indiana… H
## # ℹ 7,370 more rows
## # ℹ 10 more variables: tov <dbl>, pts <dbl>, treb <dbl>,
```

Other NBA Data

- Contains data on every game played between 2016 and 2019

```
gms %>%
  ggplot(aes(x = dateGame)) +
  geom_bar(stat = 'count')
```



Other NBA Data

```
glimpse(gms)
```

```
## Rows: 7,380
## Columns: 16
## $ idGame       <dbl> 21600001, 21600001, 21600002, 2160000…
## $ yearSeason   <int> 2017, 2017, 2017, 2017, 2017, 2017, 2…
## $ dateGame     <date> 2016-10-25, 2016-10-25, 2016-10-25, …
## $ idTeam       <dbl> 1610612739, 1610612752, 1610612757, 1…
## $ nameTeam     <chr> "Cleveland Cavaliers", "New York Knic…
## $ locationGame <chr> "H", "A", "H", "A", "H", "A", "A", "H…
## $ tov          <dbl> 14, 18, 12, 11, 16, 13, 10, 11, 15, 1…
## $ pts          <dbl> 117, 88, 113, 104, 100, 129, 108, 96,…
## $ treb         <dbl> 51, 42, 34, 31, 35, 55, 52, 45, 49, 5…
## $ oreb         <dbl> 11, 13, 5, 6, 8, 21, 16, 15, 10, 8, 1…
## $ pctFG        <dbl> 0.4833077, 0.3220769, 0.4310000, 0.51…
## $ pctFT        <dbl> 0.7500000, 0.8055000, 1.0000000, 1.00…
## $ teamrest     <dbl> 120, 120, 120, 120, 120, 120, 120, 12…
## $ second_game  <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FA…
## $ isWin        <lgl> TRUE, FALSE, TRUE, FALSE, FALSE, TRUE…
## $ ft_80        <dbl> 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0…
```

Codebook

Name	Description
idGame	Unique game id
yearSeason	Which season? NBA uses ending year so 2016-17 = 2017
dateGame	Date of the game
idTeam	Unique team id
nameTeam	Team Name
locationGame	Game location, H=Home, A=Away
tov	Total turnovers
pts	Total points
treb	Total rebounds
pctFG	Field Goal Percentage
teamrest	How many days since last game for team
pctFT	Free throw percentage
isWin	Won? TRUE or FALSE
ft_80	Team scored more than 80 percent of free throws

Codebook

- Which of these are categorical? Which are continuous?
  - Remember the **process**!
- `isWin` as an ordered binary

```
gms %>%
  count(isWin)
```

```
## # A tibble: 2 × 2
##   isWin     n
##   <lgl> <int>
## 1 FALSE  3690
## 2 TRUE   3690
```

Codebook

- The same number for wins and losses?

```
gms %>%
  select(idGame,nameTeam,dateGame,locationGame,isWin) %>% head()
```

```
## # A tibble: 6 × 5
##     idGame nameTeam              dateGame   locationGame isWin
##      <dbl> <chr>                 <date>     <chr>        <lgl>
## 1 21600001 Cleveland Cavaliers 2016-10-25 H            TRUE
## 2 21600001 New York Knicks     2016-10-25 A            FALSE
## 3 21600002 Portland Trail Bla… 2016-10-25 H            TRUE
## 4 21600002 Utah Jazz           2016-10-25 A            FALSE
## 5 21600003 Golden State Warri… 2016-10-25 H            FALSE
## 6 21600003 San Antonio Spurs   2016-10-25 A            TRUE
```

- Each row is a **team-game** pair
  - I.e., the Cavs hosted the Knicks on October 25, 2016 and won!

The Knicks

Science

- What predicts winning?
    - Points? (more is better)
    - Turnovers? (less is better)
    - Rebounds? (more is better)
- How confident are we?

```
gms %>%
  group_by(isWin) %>%
  summarise(avgTO = mean(tov))
```

```
## # A tibble: 2 × 2
##   isWin avgTO
##   <lgl> <dbl>
## 1 FALSE  13.9
## 2 TRUE   13.1
```

Turnovers and Winning

- On average, winning teams have ~1 fewer turnover than losing teams
- FSNoR: is this *always* the case?

```
gms %>%
  filter(yearSeason == 2017) %>%
  group_by(isWin) %>%
  summarise(avgTO = mean(tov))
```

```
## # A tibble: 2 × 2
##   isWin avgTO
##   <lgl> <dbl>
## 1 FALSE  13.8
## 2 TRUE   12.9
```

Turnovers and Winning

- On average, winning teams have ~1 fewer turnover than losing teams
- FSNoR: is this *always* the case?

```
gms %>%
  filter(yearSeason == 2018) %>%
  group_by(isWin) %>%
  summarise(avgTO = mean(tov))
```

```
## # A tibble: 2 × 2
##   isWin avgTO
##   <lgl> <dbl>
## 1 FALSE  14.1
## 2 TRUE   13.3
```

Turnovers and Winning

- On average, winning teams have ~1 fewer turnover than losing teams
- FSNoR: is this *always* the case?

```
gms %>%
  group_by(isWin,yearSeason) %>%
  summarise(avgTO = mean(tov)) %>%
  spread(isWin,avgTO,sep = '_')
```

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

```
## # A tibble: 3 × 3
##   yearSeason isWin_FALSE isWin_TRUE
##        <int>       <dbl>      <dbl>
## 1       2017        13.8       12.9
## 2       2018        14.1       13.3
## 3       2019        13.9       13.1
```

Turnovers and Winning

- On average, winning teams have ~1 fewer turnover than losing teams
- FSNoR: is this *always* the case?
    - Not literally (numbers change)
    - But practically?
- How **confident** are we in making this claim?
    - In each season, the average turnovers of winning teams are roughly 1 lower than the average turnovers of losing teams
    - Use **bootstrap sampling** to express this more concretely!

Looping

```r
set.seed(123)
bs_tov <- NULL
for(i in 1:1000) {
  bs_tov <- gms %>%
    sample_n(size = 100,replace = T) %>%
    group_by(isWin) %>%
    summarise(avgTO = mean(tov)) %>%
    bind_rows(bs_tov)
}
bs_tov %>% head()
```

```
## # A tibble: 6 × 2
##   isWin avgTO
##   <lgl> <dbl>
## 1 FALSE  13.6
## 2 TRUE   13.3
## 3 FALSE  13.9
## 4 TRUE   13.0
## 5 FALSE  14.1
## 6 TRUE   13.0
```

Bootstrapped Estimates vs Data

```
bs_tov %>%
  group_by(isWin) %>%
  summarise(bs_est = mean(avgTO))
```

```
## # A tibble: 2 × 2
##   isWin bs_est
##   <lgl>  <dbl>
## 1 FALSE   13.9
## 2 TRUE    13.1
```

```
gms %>%
  group_by(isWin) %>%
  summarise(data_est = mean(tov))
```

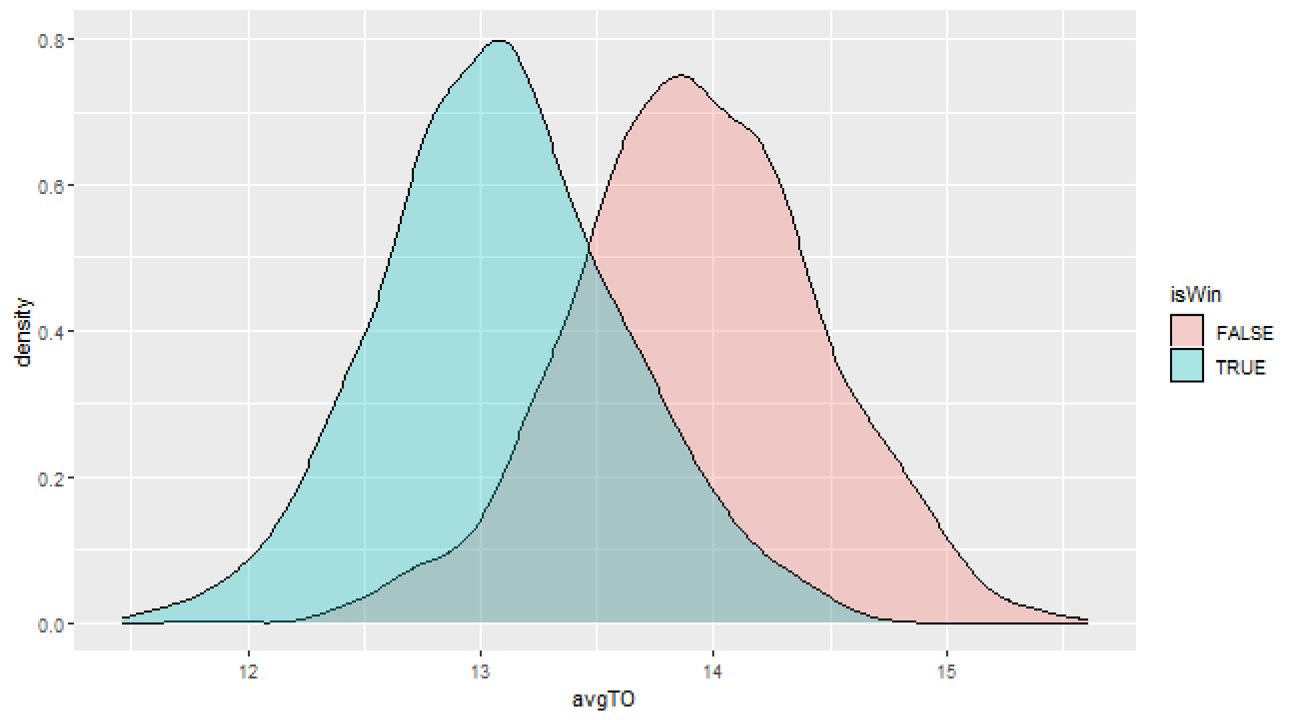
```
## # A tibble: 2 × 2
##   isWin data_est
##   <lgl>    <dbl>
## 1 FALSE     13.9
## 2 TRUE      13.1
```

Bootstrapped Estimates vs Data

- They're identical!
    - In theory, bootstrapped samples converge on true values
    - ...where "true" is the full data
- So then why bother with bootstrapping?
- **Uncertainty!**

Plot Distributions of Bootstraps

```
bs_tov %>%
  ggplot(aes(x = avgTO,fill = isWin)) +
  geom_density(alpha = .3)
```



Generalizability

- What if we only used one season?
  - Do we think our conclusions would "generalize" (i.e., apply to) other seasons?
  - For example, is the turnover-win relationship the same in the 2017 season as the 2018 season?
  - What about the 2019 season?
  - Why or why not?
- Demonstrate using the 2017 data

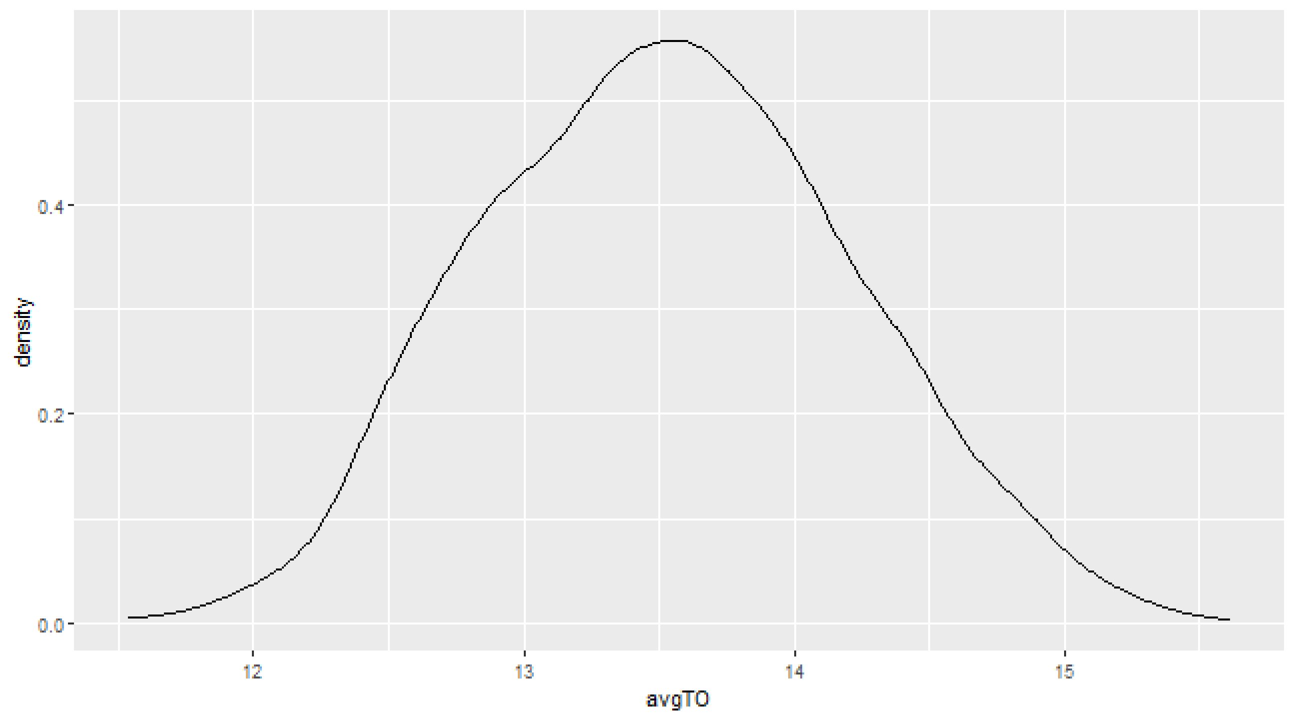
Generalizability

- Bootstrap + `group_by`

```
bsRes <- NULL

for(i in 1:500) {  # Only 500 simulations this time
  bsRes <- gms %>%
    group_by(yearSeason) %>% #<< Group by the season
    sample_n(size = 100,replace = T) %>% #<< Get 100 observations per season
    group_by(yearSeason,isWin) %>% #<< Then calculate mean tov by season AND win
    summarise(avgTO = mean(tov,na.rm=T),.groups = 'drop') %>%
    ungroup() %>%
    mutate(bsInd = i) %>%
    bind_rows(bsRes)

}
```

Plotting the results

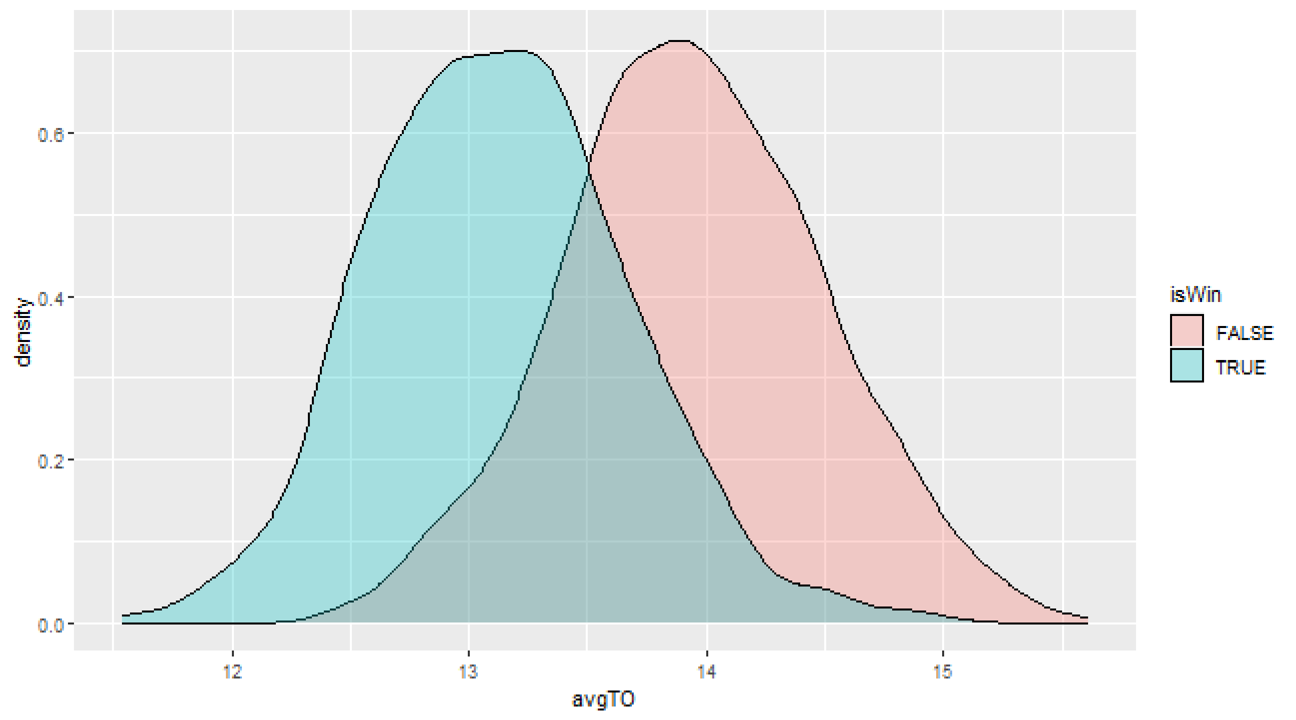
```
bsRes %>%
  ggplot(aes(x = avgTO)) +
  geom_density(alpha = .3)
```



- Is this answering our question?

Plotting the results

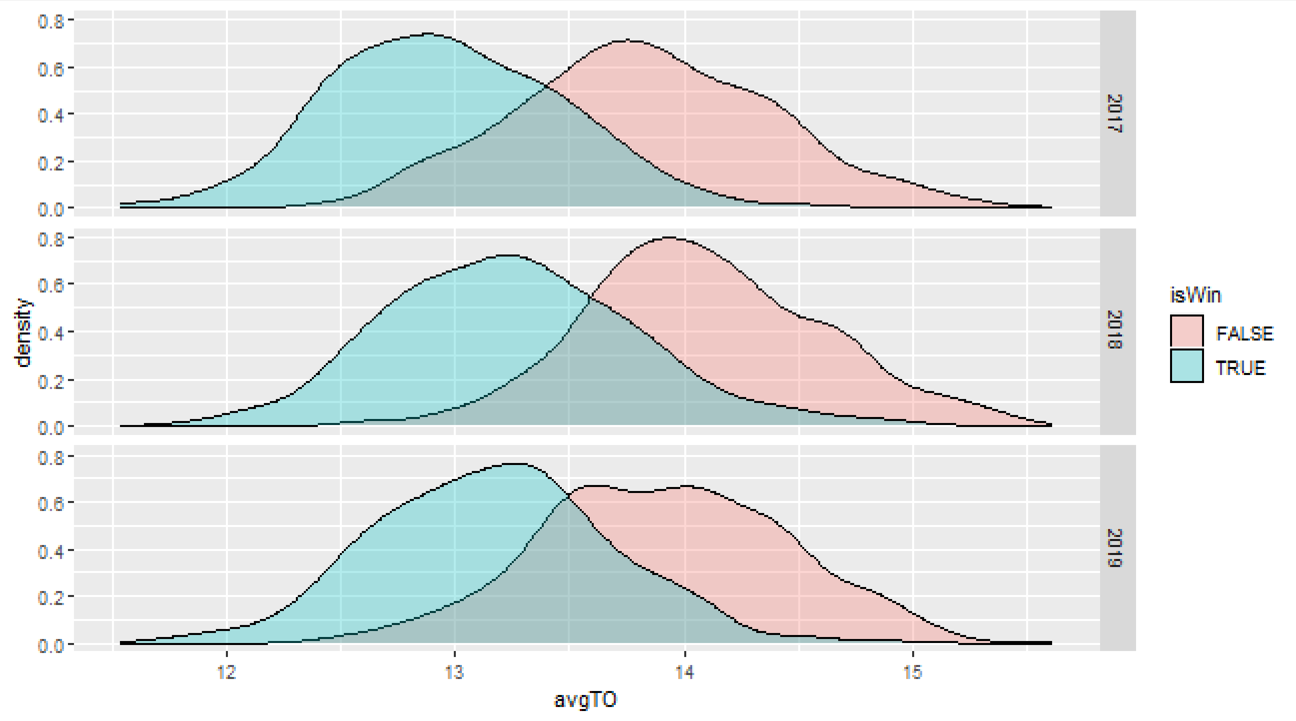
```
bsRes %>%
  ggplot(aes(x = avgTO,fill = isWin)) +
  geom_density(alpha = .3)
```



- Is this answering our question?

Plotting the results

```
bsRes %>%
  ggplot(aes(x = avgTO,fill = isWin)) +
  geom_density(alpha = .3) +
  facet_grid(yearSeason~.)
```



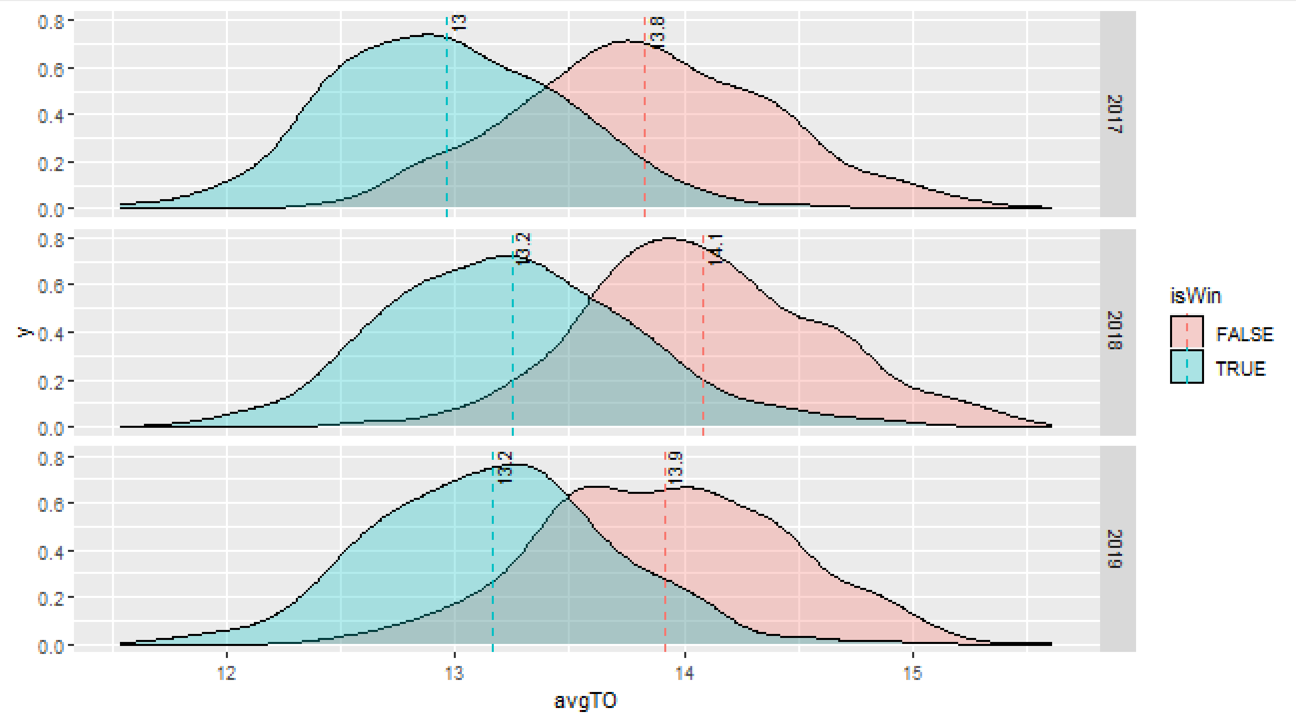
Plotting the results

```
p  <- bsRes %>%
  ggplot(aes(x = avgTO,fill = isWin)) +
  geom_density(alpha = .3) +
  geom_vline(data = bsRes %>%
               group_by(yearSeason,isWin) %>%
               summarise(avgTO = mean(avgTO,na.rm=T)),
             aes(xintercept = avgTO,color = isWin),linetype =
'dashed') +
  geom_text(data = bsRes %>%
               group_by(yearSeason,isWin) %>%
               summarise(avgTO = mean(avgTO,na.rm=T)),
             aes(x = avgTO,y = Inf,label = round(avgTO,1)),hjust =
1.1,vjust = 1.1,size = 3,angle = 90) +
  facet_grid(yearSeason~.)
```

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

Plotting the results

```
p
```



Summarizing further

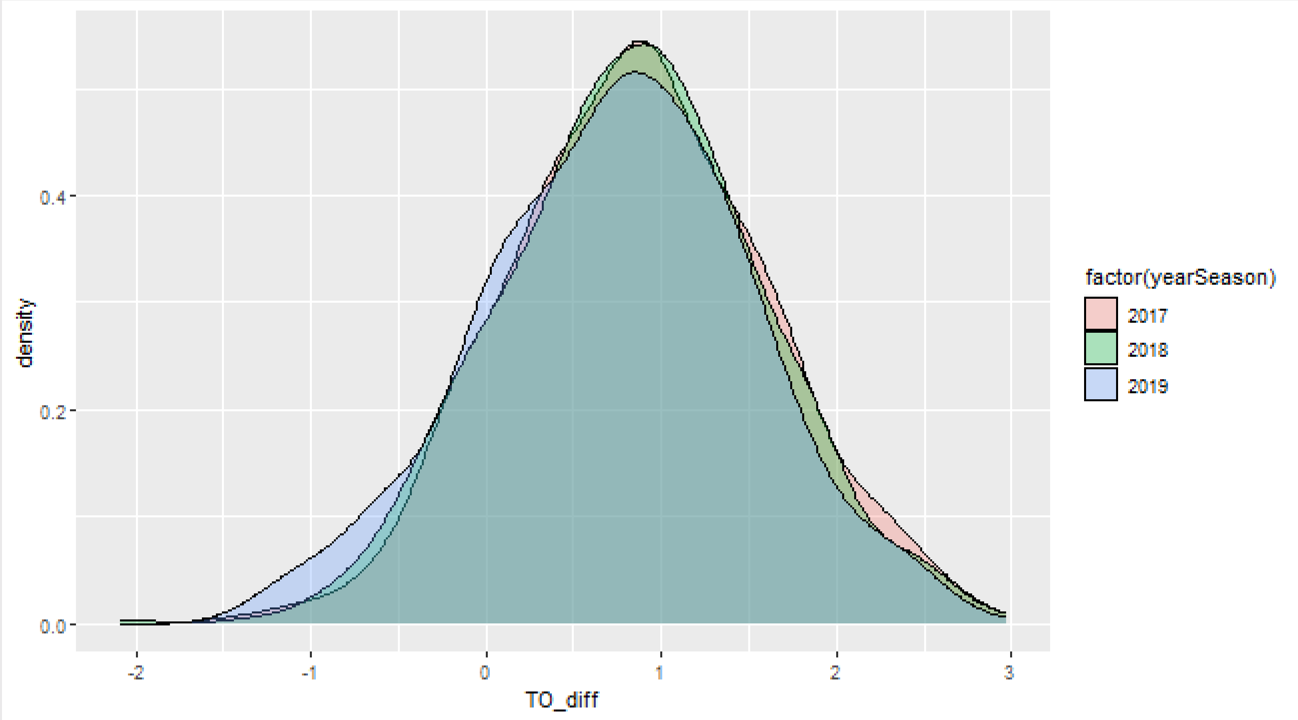
- We are *actually* interested in whether winning teams turnover the ball less
  - Science: never forget your theory / hypothesis!
- So let's actually calculate this!
- The `spread` command to create two columns

```
bsRes %>%
  spread(isWin,avgTO,sep = '_') %>%
  mutate(TO_diff = isWin_FALSE - isWin_TRUE)
```

```
## # A tibble: 1,500 × 5
##    yearSeason bsInd isWin_FALSE isWin_TRUE TO_diff
##         <int> <int>       <dbl>      <dbl>   <dbl>
##  1       2017     1        14.3       13.1  1.16
##  2       2017     2        14.1       12.5  1.60
##  3       2017     3        13.6       13.9 -0.285
##  4       2017     4        13.6       12.3  1.34
##  5       2017     5        14.1       13.4  0.739
##  6       2017     6        14.3       12.9  1.47
##  7       2017     7        13.4       13.4 -0.0161
```

Generalizability

```
bsRes %>%
  spread(isWin,avgTO,sep = '_') %>%
  mutate(TO_diff = isWin_FALSE - isWin_TRUE) %>%
  ggplot(aes(x = TO_diff,fill = factor(yearSeason))) +
  geom_density(alpha = .3)
```

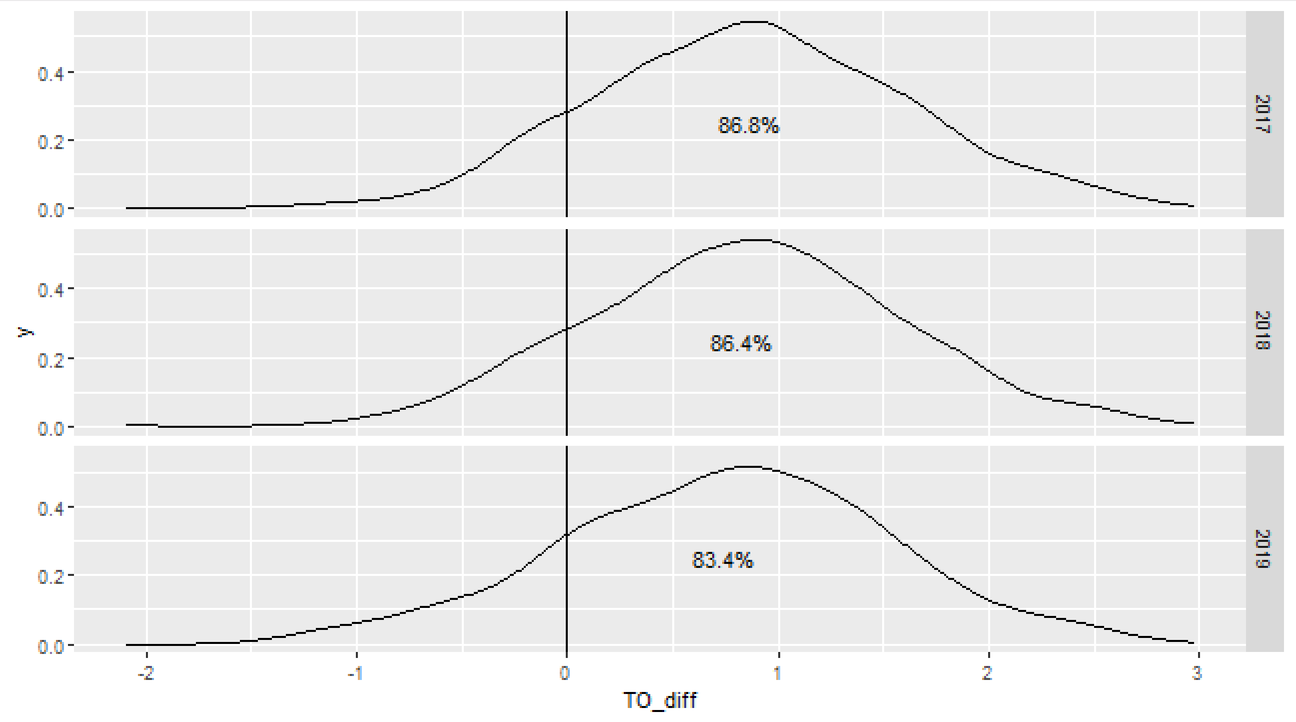


Comparing across seasons

```
p <- bsRes %>%
  spread(isWin,avgTO,sep = '_') %>%
  mutate(TO_diff = isWin_FALSE - isWin_TRUE) %>%
  ggplot(aes(x = TO_diff,group = yearSeason)) +
  geom_density(alpha = .3) +
  geom_vline(xintercept = 0) +
  geom_text(data = bsRes %>%
              spread(isWin,avgTO,sep = '_') %>%
              mutate(TO_diff = isWin_FALSE - isWin_TRUE) %>%
              group_by(yearSeason) %>%
              summarise(conf = mean(TO_diff > 0),
                        TO_diff = mean(TO_diff),
                        y = .25),
             aes(x = TO_diff,y = y,label =
paste0(round(conf*100,1),'%'))) +
  facet_grid(yearSeason ~.)
```

Comparing across seasons

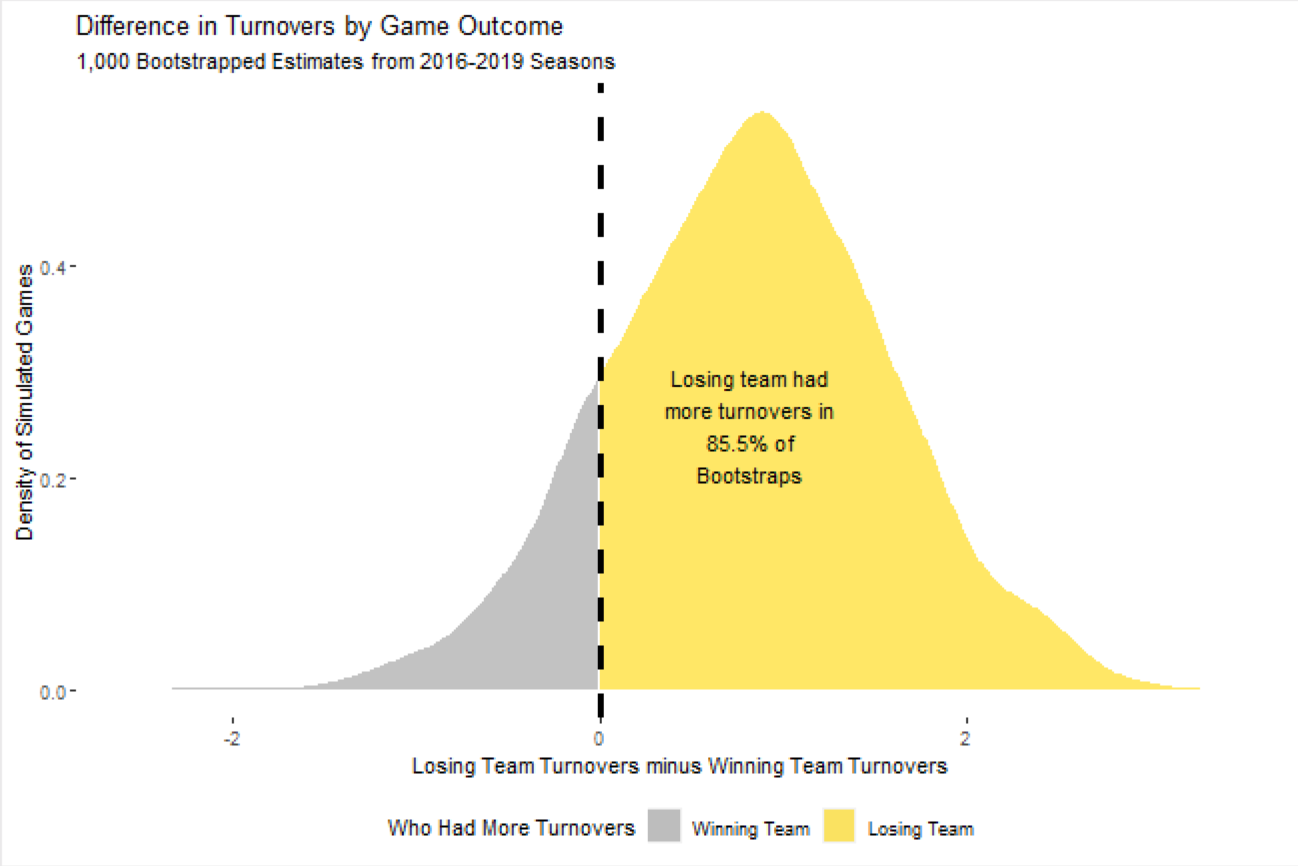
```
p
```



Visualization is **DEEP**

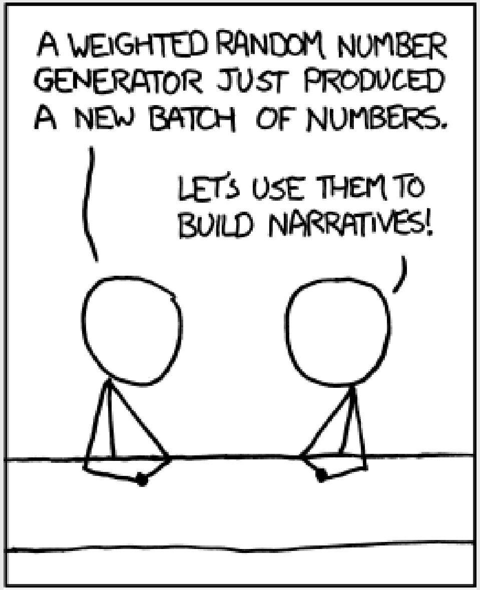
```
toplot <- bsRes %>%
  spread(isWin,avgTO,sep = '_') %>%
  mutate(TO_diff = isWin_FALSE - isWin_TRUE)

tmp <- density(toplot$TO_diff)
p <- data.frame(x = tmp$x,y = tmp$y,
           area = tmp$x >= 0) %>%
  ggplot(aes(x = x,ymin = 0,ymax = y,fill = area)) +
  geom_ribbon(alpha = .6) +
  geom_vline(xintercept = 0,linetype = 'dashed',size = 1.1) +
  annotate(geom = 'text',x = mean(toplot$TO_diff),y = .25,
           label = paste0("Losing team had\nmore turnovers
in\n",round(mean(toplot$TO_diff > 0),3)*100,"% of\nBootstraps"),
           hjust = .5) +
  labs(title = 'Difference in Turnovers by Game Outcome',
       subtitle = '1,000 Bootstrapped Estimates from 2016-2019
Seasons',
       x = 'Losing Team Turnovers minus Winning Team Turnovers',
       y = 'Density of Simulated Games') +
  scale_fill_manual(name = 'Who Had More Turnovers',
                    values = c('grey60','gold'),labels = c('Winning
Team','Losing Team')) +
```

Visualization is **DEEP**



Conclusion

- Anyone can spit stats



ALL SPORTS COMMENTARY

- Data scientists are comfortable with **uncertainty**

Quiz & Homework

- Go to Brightspace and take the **10th** quiz
    - The password to take the quiz is ####
- **Homework:**
    1. Work through ds1000_hw_11.Rmd (regression!)
    2. Finish Problem Set 6 (on Brightspace)