Multivariate Analysis

Part 3: Uncertainty

Prof. Bisbee

Vanderbilt University

Lecture Date: 2023/10/04

Slides Updated: 2023-09-30

Agenda

1. Uncertainty
2. More NBA data
3. Bootstrap Sampling
4. Applied to Polls

The Missing Ingrediant

- Thus far we have:

  1. Tested whether **selective** schools have **higher SAT scores**: Yes
  2. Tested Trump's theory that **polls were biased against him**: No
  3. Tested whether RDD polls **contact more Trump supporters**: No
  4. Tested whether state polls **accurately predicted the president**: No

- We want to do more than say "Yes" or "No" when answering a Research Question or making a Prediction
- We want to express our **confidence**

What is "confidence"?

- In frequentist statistics:
  - How often your conclusion would be correct if you were able to run an "experiment" many times
  - How often your conclusion would be correct if you were able to observe the world many times
- Research Question: Are NBA players from Tennessee better at shooting free throws than players from UVA?
  - Theory: ??
  - Hypothesis: ??
- Analysis: compare `pctFT` by `org`

NBA Example

```r
require(tidyverse)
```

```r
nba <- read_rds('../data/nba_players_2018.Rds')
glimpse(nba %>% select(org,pctFT))
```

```
## Rows: 530
## Columns: 2
## $ org   <fct> Texas, NA, Other, FC Barcelona Basquet, Kent…
## $ pctFT <dbl> 0.847, 0.700, 0.500, 0.923, 0.735, 0.667, 0.…
```

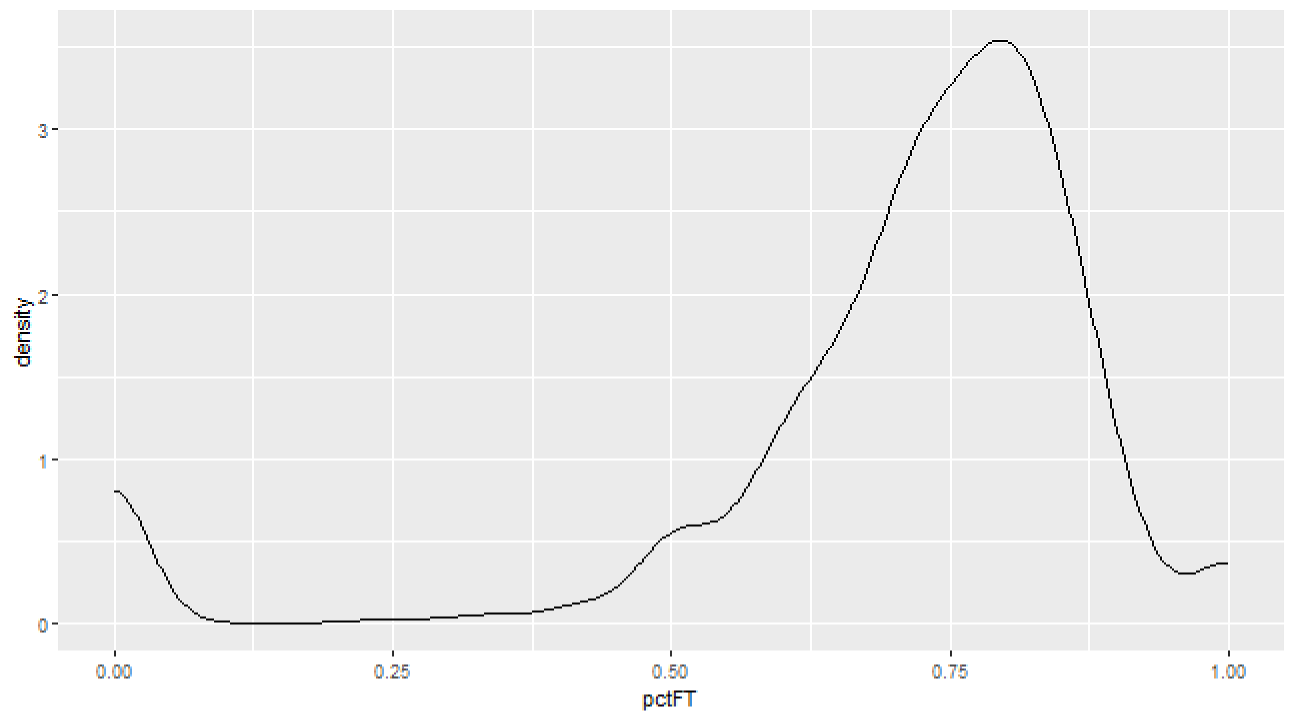
Look

```
summary(nba %>% select(pctFT,org))
```

```
##      pctFT                                   org
##  Min.   :0.0000   Other                        : 85
##  1st Qu.:0.6515   Kentucky                     : 25
##  Median :0.7500   Duke                         : 17
##  Mean   :0.6968   California-Los Angeles: 15
##  3rd Qu.:0.8180   Kansas                       : 11
##  Max.   :1.0000   (Other)                      :220
##                   NA's                         :157
```

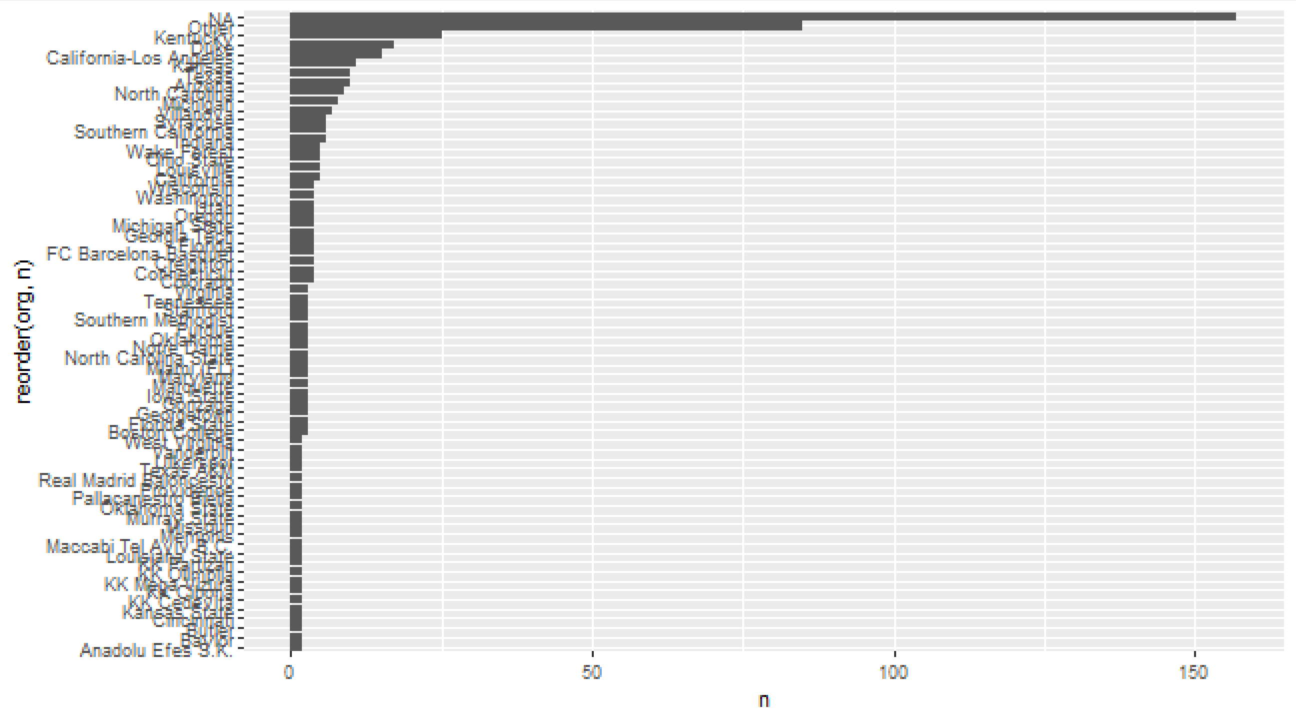
Visualize: Univariate Y

```
nba %>%
  ggplot(aes(x = pctFT)) +
  geom_density()
```



Visualize: Univariate X

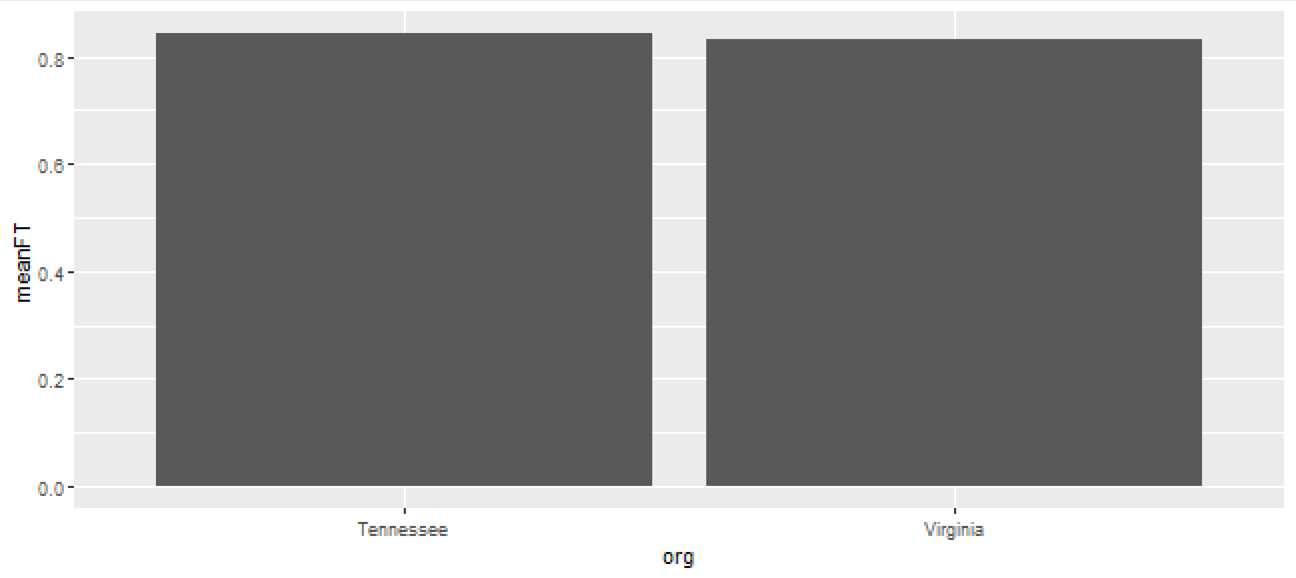
```r
nba %>%
  count(org) %>%
  ggplot(aes(x = n,y = reorder(org,n))) +
  geom_bar(stat = 'identity')
```



Visualize: Multivariate

- Option #1: `summarise()` data prior to plotting

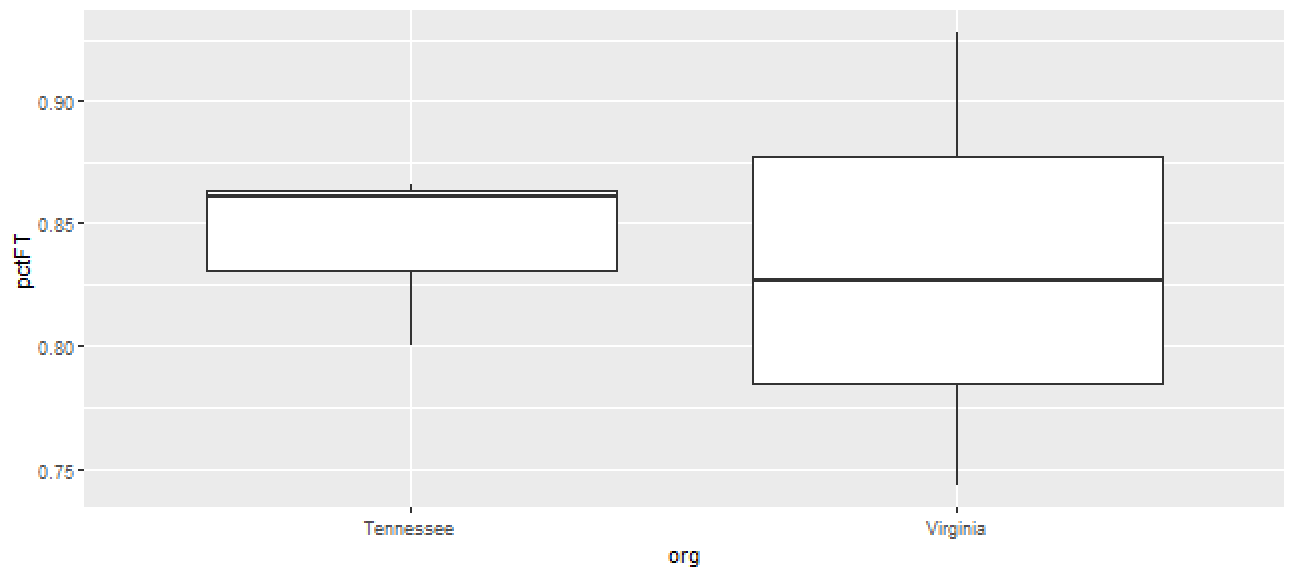
```
nba %>%
  filter(org %in% c('Tennessee','Virginia')) %>%
  group_by(org) %>% summarise(meanFT = mean(pctFT,na.rm=T)) %>%
  ggplot(aes(x = org,y = meanFT)) +
  geom_bar(stat = 'identity')
```



Visualize: Multivariate

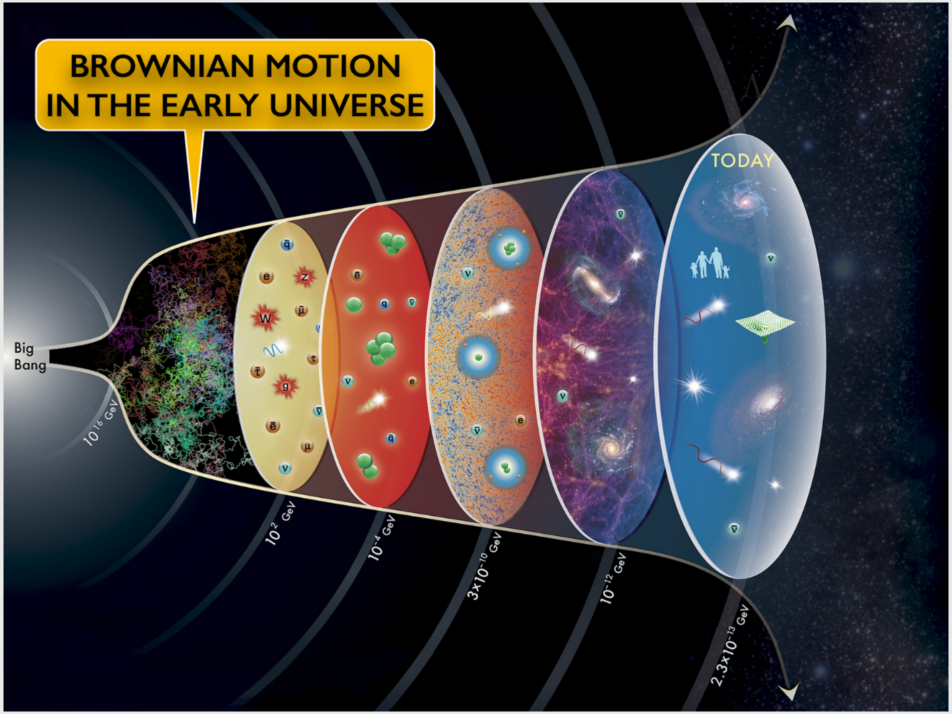
- Option #2: plot raw data

```
nba %>%
  filter(org %in% c('Tennessee','Virginia')) %>%
  ggplot(aes(x = org,y = pctFT)) +
  geom_boxplot()
```



Uncertainty

- Are players from Tennessee **better** at free throws than players from UVA?
- Big philosophical step back
  - We live in a stochastic universe!



Uncertainty

- Are players from Tennessee **better** at free throws than players from UVA?
- Populations versus samples
    - Intro stats: uncertainty due to **sample**

Uncertainty

- Big philosophical step back
    - We live in a stochastic universe!
- What does **better** mean?
    - Theory: An innate quality in greater abundance
    - Prediction: If we had to bet on who scores more FTs, who do we choose?
- How **confident** would we be with this bet?

Uncertainty

- If the universe is inherently stochastic, we are inherently uncertain
  - We THINK UT players are better FT shooters, but not 100% certain
- How to measure this?
  - Run 100 experimental seasons
  - Record FT percentage for players from UVA and UT for each season
  - Calculate how many times UT players have a better percentage than UVA players
- 90 seasons out of 100 → 90% confident / certainty
- 100 seasons out of 100 → 100%?
- **FUNDAMENTAL STOCHASTIC NATURE OF REALITY (FSNoR)**

Uncertainty

- Running 100 experimental seasons is impossible
    1. We are not Adam Silver
    2. Even if we were Adam Silver, 100 seasons = a century of basketball!

Uncertainty

- Running 100 experimental seasons is impossible

    1. We are not Adam Silver
    2. Even if we were Adam Silver, 100 seasons = a century of basketball!
    3. If we were God? 100 seasons with the same players?

- *STILL wouldn't be 100% certain due to* ***FSNoR***

    - (**F**undamental **S**tochastic **N**ature **o**f **R**eality)

Uncertainty

- But we are data scientists
- Take 1 season of basketball but sample it randomly
- **Bootstrap sampling**
- Theory: By mimicking the sampling process, we can simulate a God experiment
  - (NB: this goes much deeper. Uncertainty from bootstrap combines FSNoR + sampling uncertainty.)
- Practice: `sample_n()` + `for()` loops

Bootstrap Demo Step 1

- One randomly sampled player via `sample_n(size,replace)`
  - `size`: how many samples (from 1 to all observations)
  - `replace`: whether to put the sample back (`TRUE` or `FALSE`)

```
set.seed(123) # Ensure we can reproduce results exactly

nba %>%
  sample_n(size = 1,replace = T) %>%
select(namePlayer,slugSeason,slugTeam,pctFT)
```

```
## # A tibble: 1 × 4
##   namePlayer    slugSeason slugTeam pctFT
##   <chr>         <chr>      <chr>    <dbl>
## 1 Moritz Wagner 2018-19    LAL      0.811
```

Bootstrap Demo Step 2

- Two randomly sampled players

```
set.seed(123)
nba %>%
  sample_n(size = 1,replace = T) %>%
select(namePlayer,slugSeason,slugTeam,pctFT)
```

```
## # A tibble: 1 × 4
##   namePlayer     slugSeason slugTeam pctFT
##   <chr>          <chr>      <chr>    <dbl>
## 1 Moritz Wagner 2018-19    LAL      0.811
```

```
nba %>%
  sample_n(size = 1,replace = T) %>%
select(namePlayer,slugSeason,slugTeam,pctFT)
```

```
## # A tibble: 1 × 4
##   namePlayer slugSeason slugTeam pctFT
##   <chr>      <chr>      <chr>    <dbl>
## 1 Sam Dekker 2018-19    LAC      0.609
```

Bootstrap Demo Step 2

- OR two randomly sampled players

```
set.seed(123)

nba %>%
  sample_n(size = 2,replace = T) %>%
select(namePlayer,slugSeason,slugTeam,pctFT)
```

```
## # A tibble: 2 × 4
##   namePlayer     slugSeason slugTeam pctFT
##   <chr>          <chr>      <chr>    <dbl>
## 1 Moritz Wagner 2018-19    LAL      0.811
## 2 Sam Dekker    2018-19    LAC      0.609
```

Bootstrap Demo Step 3

- Randomly sample all players: `size = nrow(nba)` (or `nrow(.)`)

```
set.seed(123)

nba %>%
  sample_n(size = nrow(nba),replace = T) %>% # Same as nrow(.)
  select(namePlayer,slugSeason,slugTeam,pctFT)
```

```
## # A tibble: 530 × 4
##    namePlayer         slugSeason slugTeam pctFT
##    <chr>              <chr>      <chr>    <dbl>
##  1 Moritz Wagner      2018-19    LAL      0.811
##  2 Sam Dekker         2018-19    LAC      0.609
##  3 Joe Harris         2018-19    BKN      0.827
##  4 Jonas Valanciunas  2018-19    LAL      0.795
##  5 John Holland       2018-19    CLE      0
##  6 Angel Delgado      2018-19    LAC      0.5
##  7 Donovan Mitchell   2018-19    UTA      0.806
##  8 Damian Jones       2018-19    GSW      0.649
##  9 Luke Kornet        2018-19    NYK      0.826
## 10 Justin Anderson    2018-19    ATL      0.743
## # … with 520 more rows
```

Bootstrap Demo Step 4

- Linking to **confidence**: Do we draw the same conclusion twice?

```
set.seed(123)

# Bootstrapped Season #1
bsSeason1 <- nba %>%
  sample_n(size = nrow(.),replace = T) %>%
  select(org,pctFT) %>%
  mutate(bsSeason = 1)

# Bootstrapped Season #2
bsSeason2 <- nba %>%
  sample_n(size = nrow(.),replace = T) %>%
  select(org,pctFT) %>%
  mutate(bsSeason = 2)
```

Bootstrap Demo Step 4

- Linking to **confidence**: Do we draw the same conclusion twice?

```
bsSeason1 %>%
  filter(org %in% c('Tennessee','Virginia')) %>%
  group_by(org) %>%
  summarise(mean_FT = mean(pctFT))
```

```
## # A tibble: 2 × 2
##   org        mean_FT
##   <fct>        <dbl>
## 1 Tennessee    0.866
## 2 Virginia     0.785
```

```
bsSeason2 %>%
  filter(org %in% c('Tennessee','Virginia')) %>%
  group_by(org) %>%
  summarise(mean_FT = mean(pctFT))
```

```
## # A tibble: 2 × 2
##   org        mean_FT
##   <fct>        <dbl>
```

Bootstrap Demo Step 5

- Want to do this 100 times!
- Use a `for()` loop to make it cleaner
- A `for()` loop repeats the same code multiple times
    - Benefit: don't need to copy and paste a chunk of code 100 times
    - Just put a chunk of code in a loop that repeats 100 times!

```
set.seed(123) # Ensure you'll get the same results each time
bsSeasons <- NULL # Instantiate empty object
for(bsSeason in 1:100) { # Repeat 100 times
  tmpSeason <- nba %>%
    sample_n(size = nrow(.),replace = T) %>% # Sample the data
    select(org,pctFT) %>% # Select variables of interest
    mutate(bsSeasonNumber = bsSeason) # Save the simulation ID
  bsSeasons <- bind_rows(bsSeasons,tmpSeason) # Append to the empty
object!
}
```

Bootstrap to measure Confidence

- Compare UVA and UT's FT percentages in each season

```
bsSeasons %>%
  filter(grepl('Tennessee|^Virginia',org)) %>%
  group_by(bsSeasonNumber,org) %>%
  summarise(mean_ftp = mean(pctFT),.groups = 'drop')
```

```
## # A tibble: 188 × 3
##    bsSeasonNumber org       mean_ftp
##             <int> <fct>        <dbl>
##  1              1 Tennessee    0.866
##  2              1 Virginia     0.785
##  3              2 Tennessee    0.866
##  4              2 Virginia     0.799
##  5              3 Tennessee    0.816
##  6              3 Virginia     0.827
##  7              4 Tennessee    0.847
##  8              4 Virginia     0.852
##  9              5 Tennessee    0.852
## 10              5 Virginia     0.836
## # … with 178 more rows
```

Bootstrap to measure Confidence

- Compare UVA and UT's FT percentages in each season

```
bsSeasons %>%
  filter(grepl('Tennessee|^Virginia',org)) %>%
  group_by(bsSeasonNumber,org) %>%
  summarise(mean_ftp = mean(pctFT),.groups = 'drop') %>%
  spread(org,mean_ftp)
```

```
## # A tibble: 100 × 3
##    bsSeasonNumber Tennessee Virginia
##             <int>     <dbl>    <dbl>
##  1              1     0.866    0.785
##  2              2     0.866    0.799
##  3              3     0.816    0.827
##  4              4     0.847    0.852
##  5              5     0.852    0.836
##  6              6     0.866    0.771
##  7              7     0.861   NA
##  8              8     0.842   NA
##  9              9     0.863    0.836
## 10             10     0.833    0.743
## # … with 90 more rows
```

Bootstrap + `filter()`

- We are missing an observation for Virginia in the 7th simulated season!
- Why?
  - Just bad luck...didn't get any players in that sample
- Could ignore, or could `filter()` the data prior to bootstrapping

Bootstrap + `filter()`

```
nbaTNVA <- nba %>% filter(org %in% c('Tennessee','Virginia'))
set.seed(123)
bsSeasons <- NULL
for(counter in 1:100) {
  tmpSeason <- nbaTNVA %>%
    sample_n(size = nrow(.),replace = T) %>%
    select(org,pctFT) %>%
    mutate(bsSeasonNumber = counter)

  bsSeasons <- bind_rows(bsSeasons,tmpSeason)
}

nrow(bsSeasons)
```

```
## [1] 600
```

Bootstrap to measure Confidence

- Compare UVA and UT's FT percentages in each season

```
bsSeasons %>%
  group_by(bsSeasonNumber,org) %>%
  summarise(mean_ftp = mean(pctFT),.groups = 'drop') %>%
  spread(org,mean_ftp) %>%
  filter(complete.cases(.)) %>%
  mutate(TNWin = ifelse(Tennessee > Virginia,1,0))
```

```
## # A tibble: 95 × 4
##    bsSeasonNumber Tennessee Virginia TNWin
##             <int>     <dbl>    <dbl> <dbl>
##  1              1     0.866    0.878     0
##  2              2     0.848    0.785     1
##  3              3     0.861    0.830     1
##  4              4     0.830    0.810     1
##  5              5     0.844    0.833     1
##  6              6     0.841    0.833     1
##  7              7     0.830    0.810     1
##  8              8     0.863    0.833     1
##  9              9     0.841    0.805     1
## 10             10     0.863    0.810     1
```

Bootstrap to measure Confidence

- Compare UVA and UT's FT percentages in each season

```
(conf <- bsSeasons %>%
  group_by(bsSeasonNumber,org) %>%
  summarise(mean_ftp = mean(pctFT),.groups = 'drop') %>%
  spread(org,mean_ftp) %>%
  filter(complete.cases(.)) %>%
  mutate(TNWin = ifelse(Tennessee > Virginia,1,0)) %>%
  summarise(TNWin = mean(TNWin)))
```

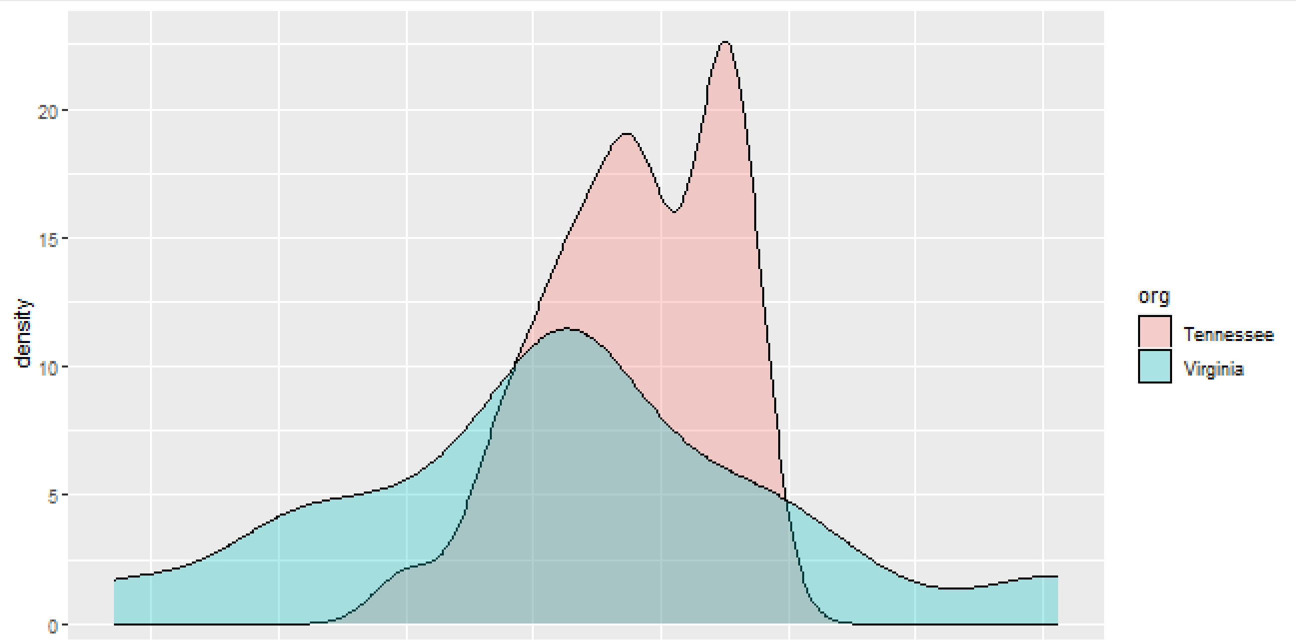
```
## # A tibble: 1 × 1
##   TNWin
##   <dbl>
## 1 0.674
```

- TN beats UVA 67.4% of the time! (How much do you bet on next season?)

Other ways to use bootstraps

- Could plot the **distributions** for each school

```
bsSeasons %>%
  group_by(org,bsSeasonNumber) %>%
  summarise(mean_FT = mean(pctFT)) %>%
  ggplot(aes(x = mean_FT,fill = org)) +
  geom_density(alpha = .3)
```



Other ways to use bootstraps

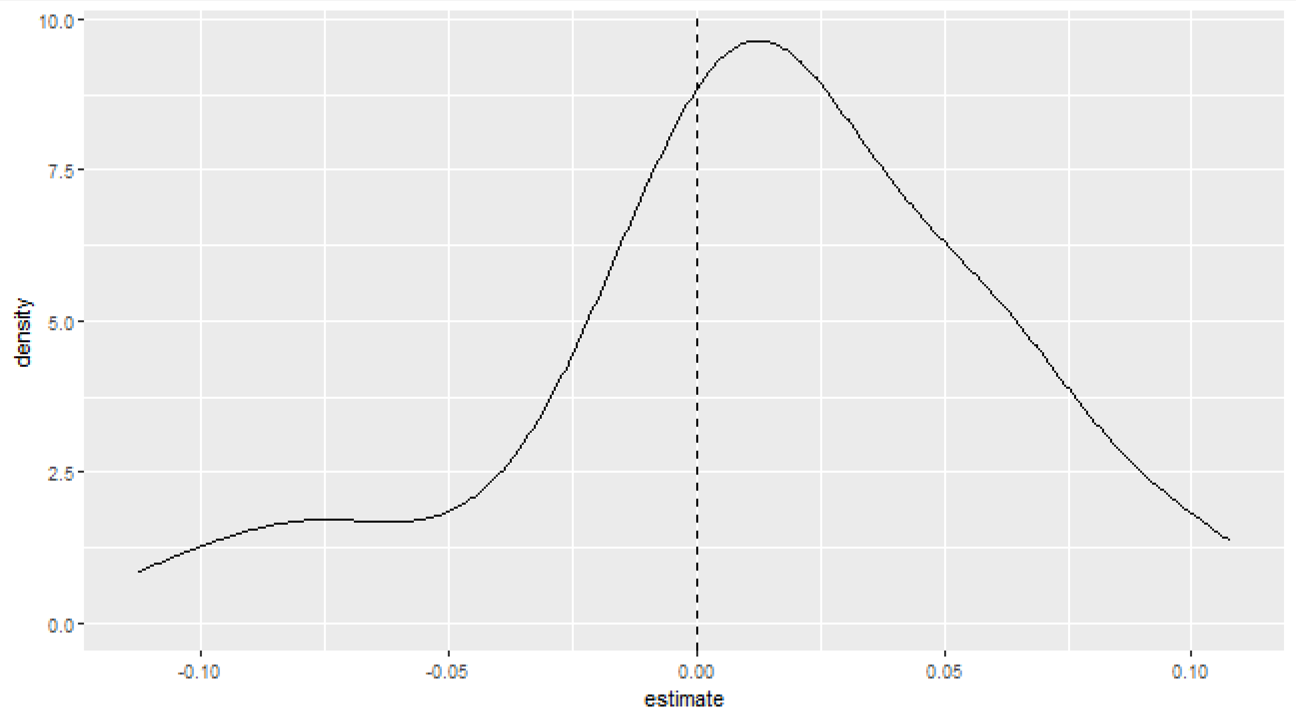
- Could plot the **distributions** of the "estimate"

```r
p <- bsSeasons %>%
  group_by(org,bsSeasonNumber) %>%
  summarise(mean_FT = mean(pctFT)) %>%
  spread(key = org,value = mean_FT) %>%
  mutate(estimate = Tennessee - Virginia) %>%
  ggplot(aes(x = estimate)) +
  geom_density(alpha = .3) +
  geom_vline(xintercept = 0,linetype = 'dashed')
```

Other ways to use bootstraps

- Could plot the **distributions** of the "estimate"

```
p
```



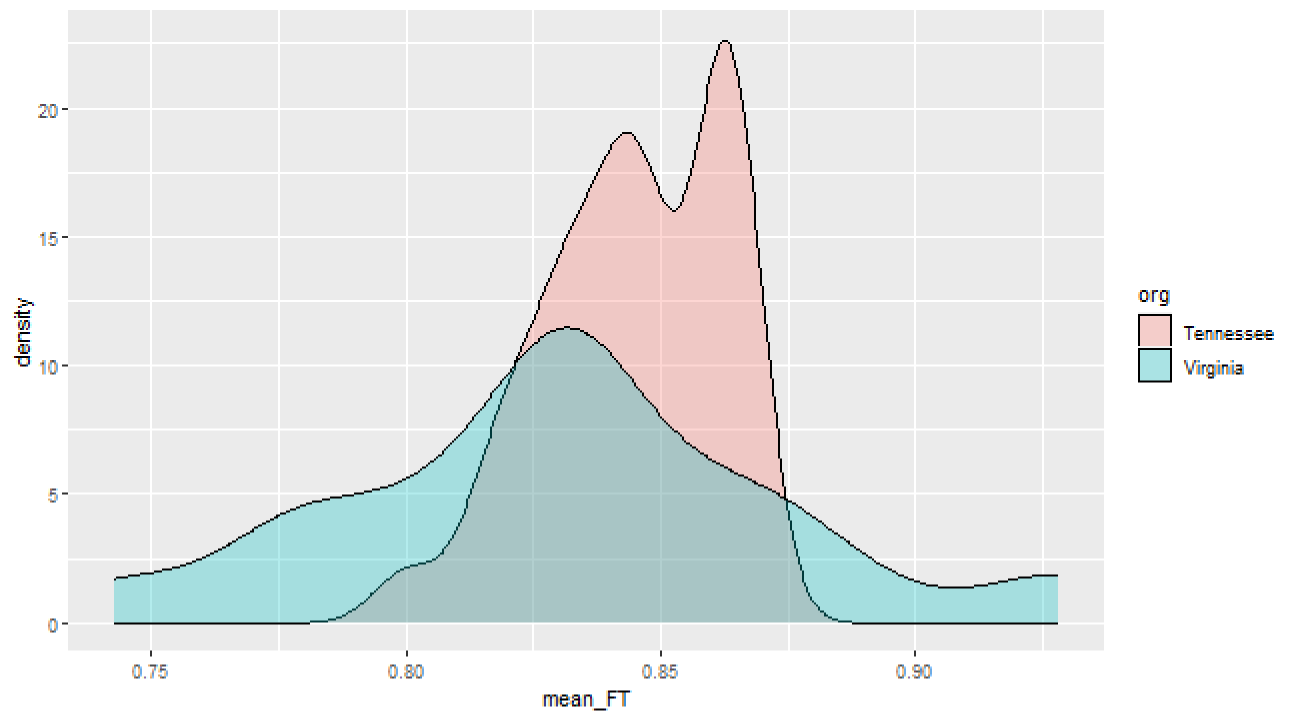
Where to calculate the "estimate"

- **First** we created a new dataset of 100 simulated seasons
- **Then** we calculate average FT % for TN and UVA for each simulation
- **Finally** we calculate proportion of times average is higher for TN
- **BUT!** It is equally valid to calculate the "estimate" *within* the `for()` loop

```
set.seed(123)
bsRes <- NULL
for(counter in 1:100) {
  tmpEst <- nbaTNVA %>%
    sample_n(size = nrow(.),replace = T) %>%
    group_by(org) %>%
    summarise(mean_FT = mean(pctFT,na.rm=T)) %>%
    mutate(bsSeason = counter)

  bsRes <- bind_rows(bsRes,tmpEst)
}
```

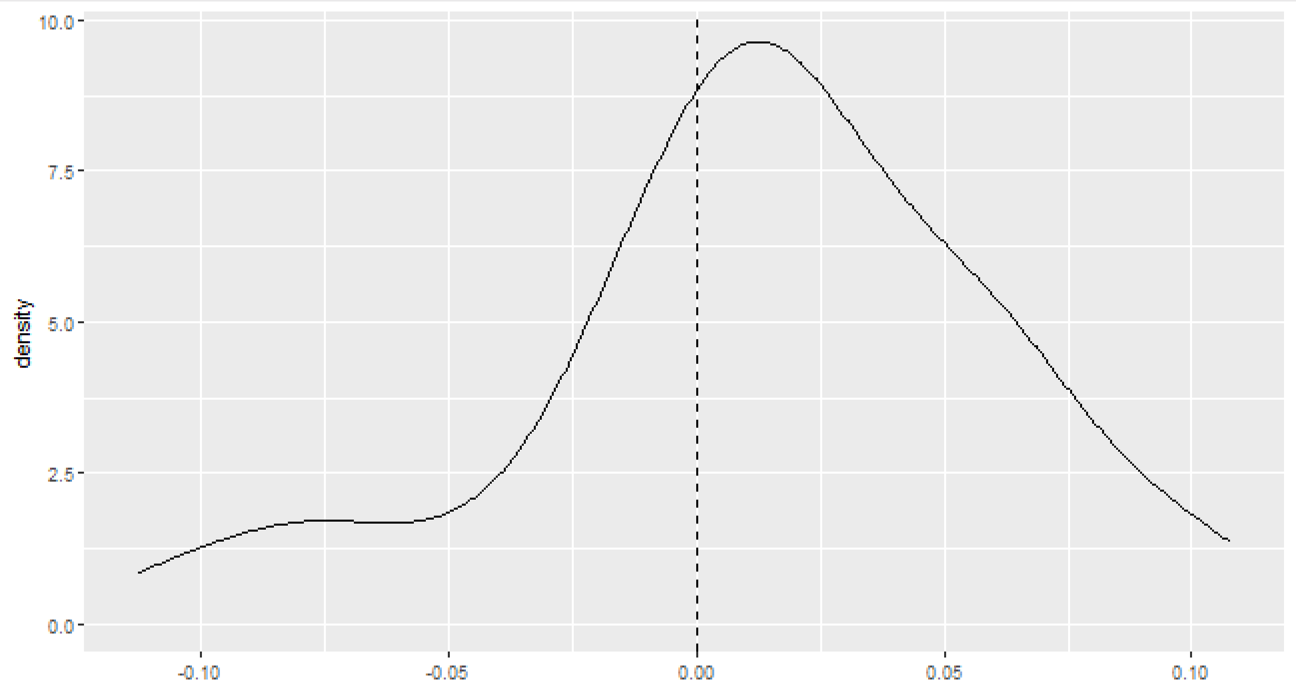
Where to calculate the "estimate"

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



Where to calculate the "estimate"

```
bsRes %>%
  spread(org,mean_FT) %>%
  mutate(TNWin = Tennessee - Virginia) %>%
  ggplot(aes(x = TNWin)) +
  geom_density(alpha = .3) +
  geom_vline(xintercept = 0,linetype = 'dashed')
```



Where to calculate the "estimate"

- Same confidence measure

```
bsRes %>%
  spread(key = org,value = mean_FT) %>%
  mutate(TNWin = ifelse(Tennessee > Virginia,1,0)) %>%
  summarise(confidence = mean(TNWin,na.rm=T))
```

```
## # A tibble: 1 × 1
##   confidence
##        <dbl>
## 1      0.674
```

Interpreting Confidence

- **Is this high?**
  - What value reflects the minimum confidence?
  - A coin flip → 50%
- What does a confidence level of 0.1 (or 10%) mean?
  - We are 90% confident that Virginia is better!

Other Applications

- Could do the same to express **confidence** in conclusions about:
  - The relationship between SAT scores and selective admissions
  - The relationship between MSM polls and anti-Trump bias
  - Whether state polls are good at predicting the 2020 president

Other NBA Data

- Download and load the `game_summary.Rds` data

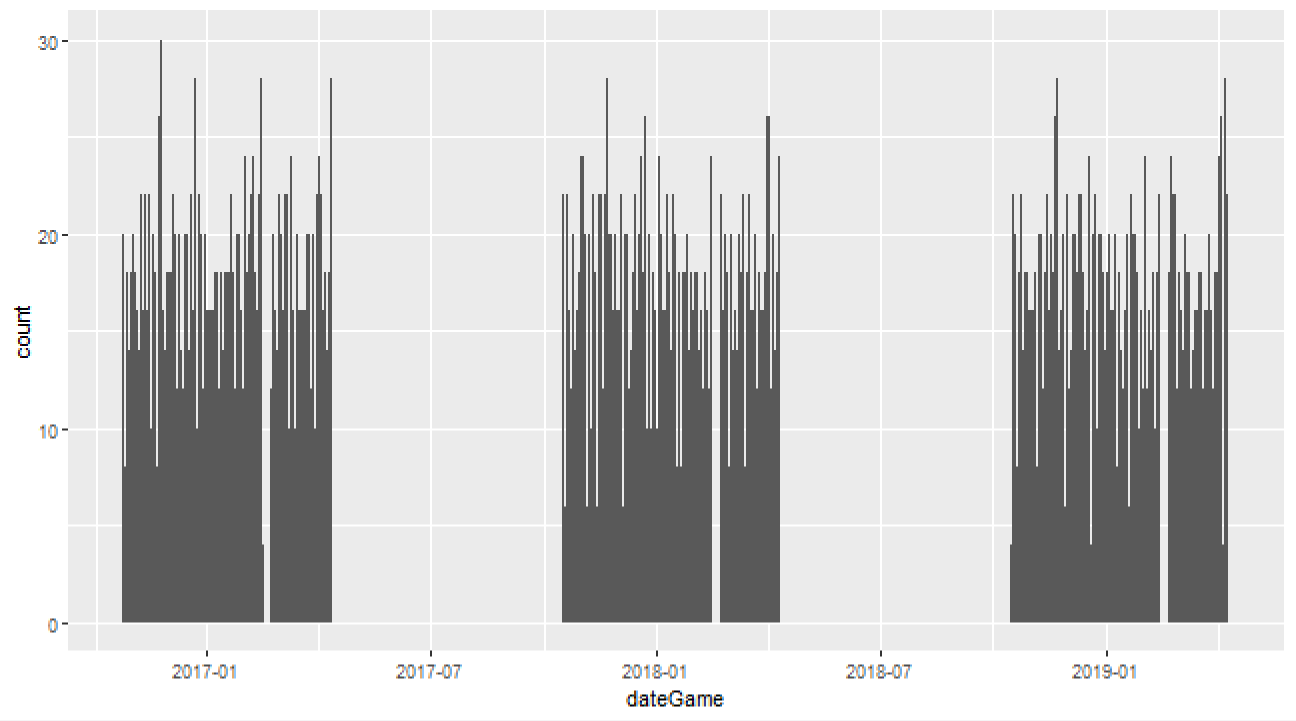
```
gms <- readRDS('../data/game_summary.Rds')
gms
```

```
## # A tibble: 7,380 × 16
##       idGame yearSe…¹ dateGame   idTeam nameT…² locat…³   tov
##        <dbl>    <int> <date>      <dbl> <chr>   <chr>   <dbl>
##  1 21600001     2017 2016-10-25 1.61e9 Clevel… H          14
##  2 21600001     2017 2016-10-25 1.61e9 New Yo… A          18
##  3 21600002     2017 2016-10-25 1.61e9 Portla… H          12
##  4 21600002     2017 2016-10-25 1.61e9 Utah J… A          11
##  5 21600003     2017 2016-10-25 1.61e9 Golden… H          16
##  6 21600003     2017 2016-10-25 1.61e9 San An… A          13
##  7 21600004     2017 2016-10-26 1.61e9 Miami … A          10
##  8 21600004     2017 2016-10-26 1.61e9 Orland… H          11
##  9 21600005     2017 2016-10-26 1.61e9 Dallas… A          15
## 10 21600005     2017 2016-10-26 1.61e9 Indian… H          16
## # … with 7,370 more rows, 9 more variables: pts <dbl>,
## #   treb <dbl>, oreb <dbl>, pctFG <dbl>, pctFT <dbl>,
## #   teamrest <dbl>, second_game <lgl>, isWin <lgl>,
## #   ft_80 <dbl>, and abbreviated variable names
```

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   locatio…¹ 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 Blazers 2016-10-25 H         TRUE
## 4 21600002 Utah Jazz              2016-10-25 A         FALSE
## 5 21600003 Golden State Warriors  2016-10-25 H         FALSE
## 6 21600003 San Antonio Spurs      2016-10-25 A         TRUE
## # … with abbreviated variable name ¹locationGame
```

- 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(20220921)
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  14.1
## 2 TRUE   13.4
## 3 FALSE  15.0
## 4 TRUE   12.6
## 5 FALSE  14.0
## 6 TRUE   12.8
```

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))
```

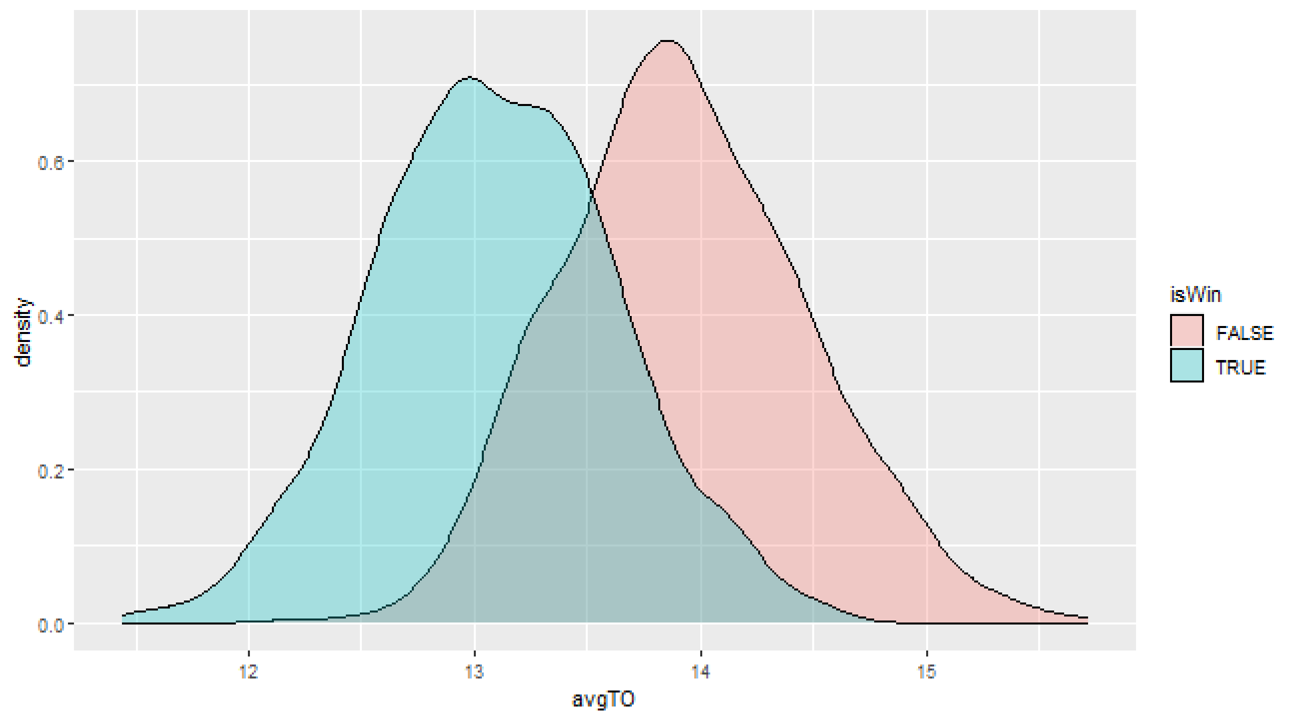
```
## # A tibble: 2 × 2
##   isWin data_est
##   <lgl>    <dbl>
## 1 FALSE     13.9
## 2 TRUE      13.1
```

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

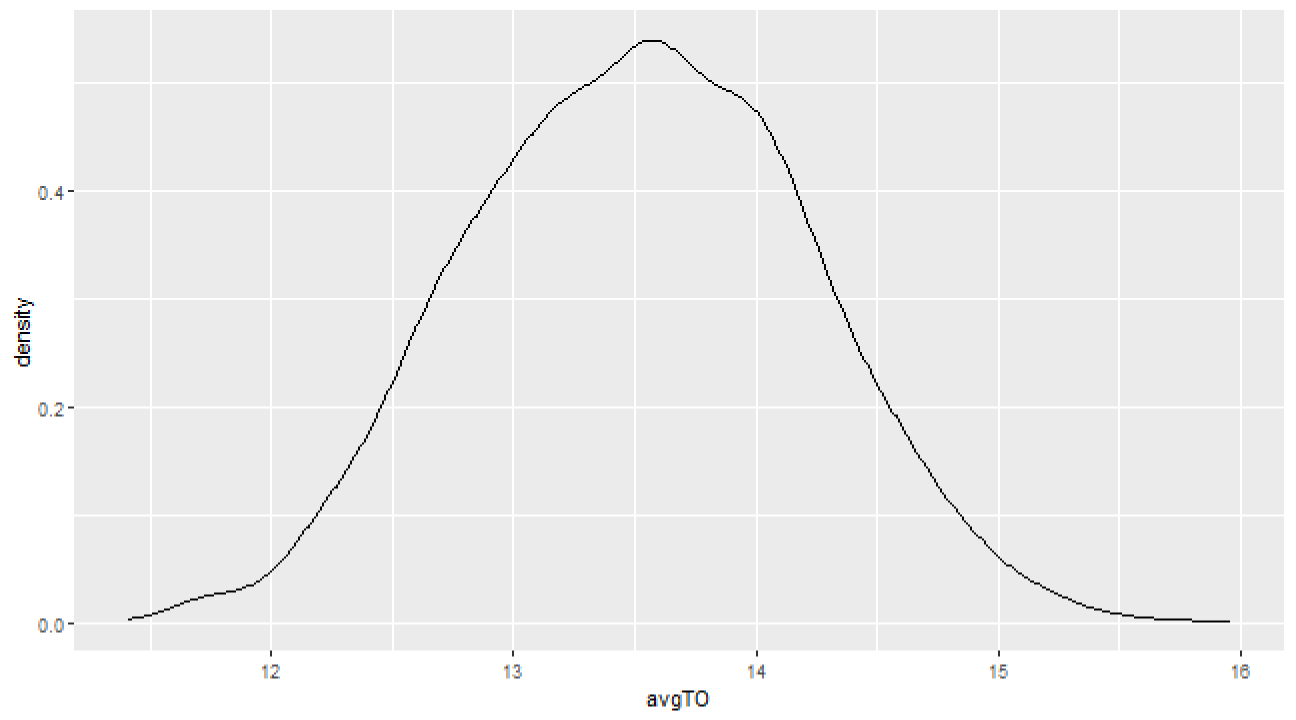
Generalizability

- 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

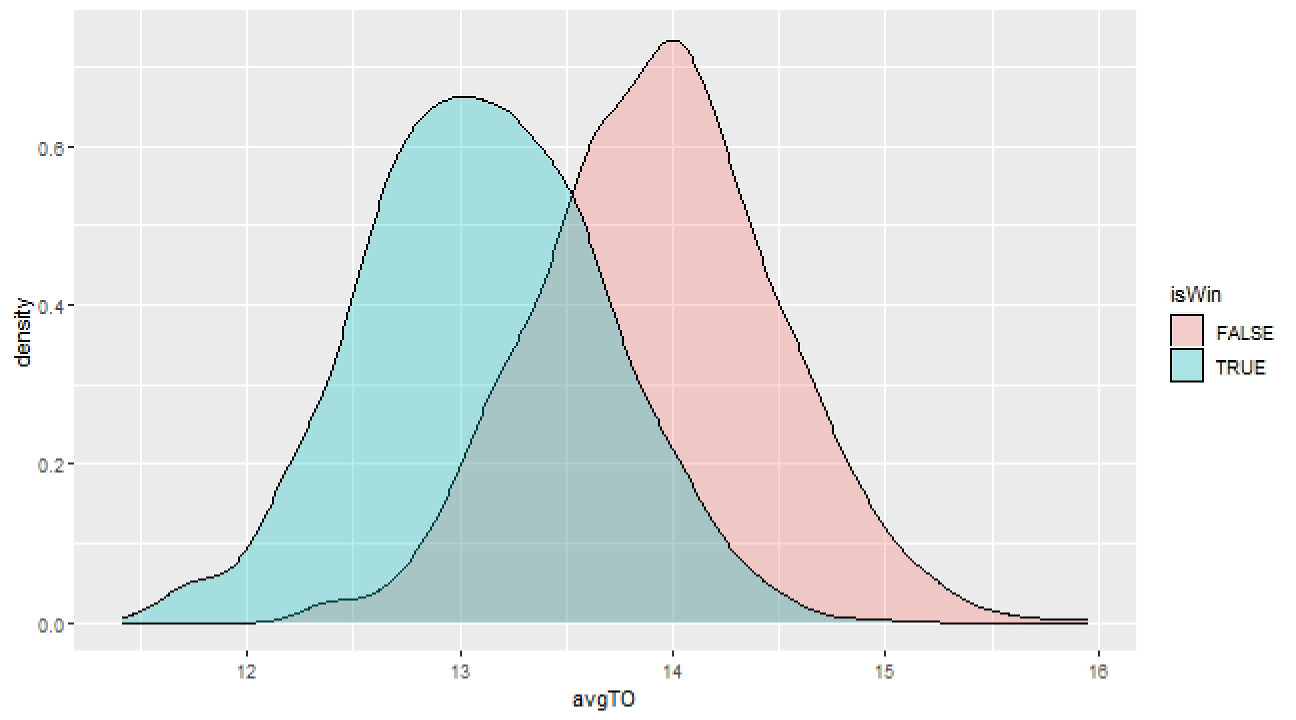
```
bsRes %>%
  ggplot(aes(x = avgTO)) +
  geom_density(alpha = .3)
```



- Is this answering our question?

Plotting the results

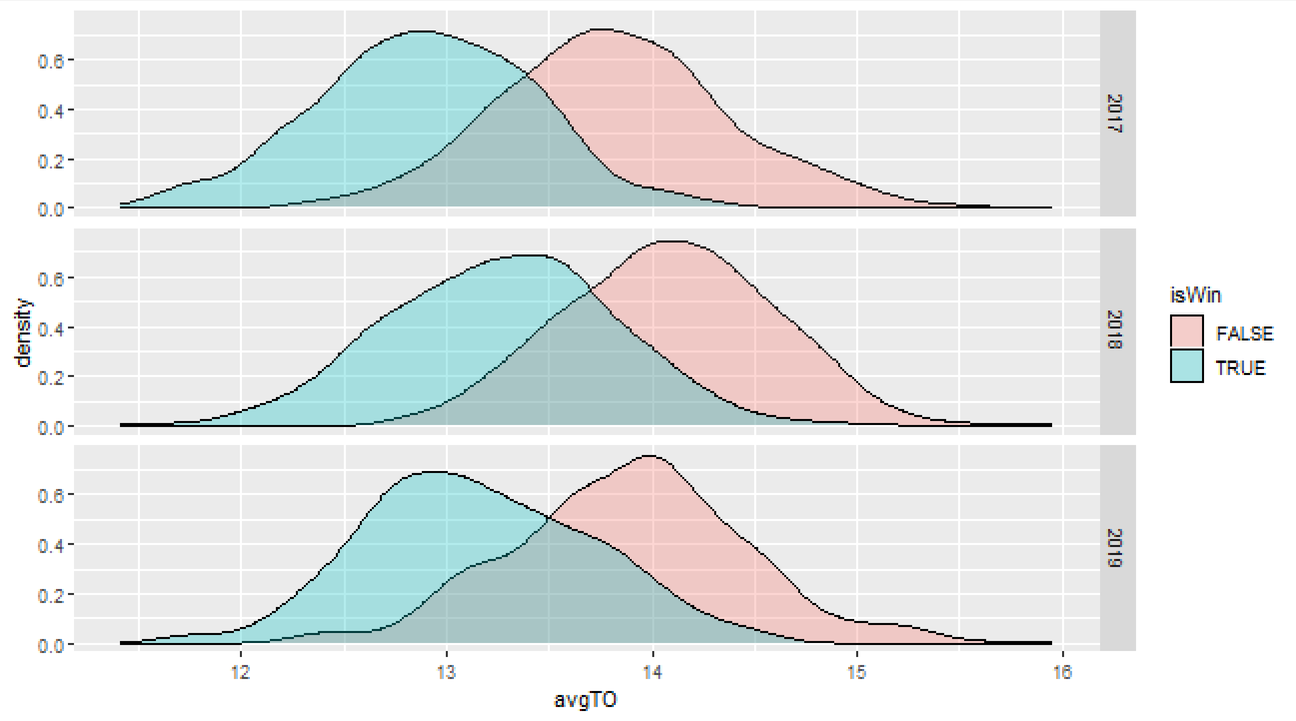
```
bsRes %>%
  ggplot(aes(x = avgTO,fill = isWin)) +
  geom_density(alpha = .3)
```



- Is this answering our question?

Plotting the results

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



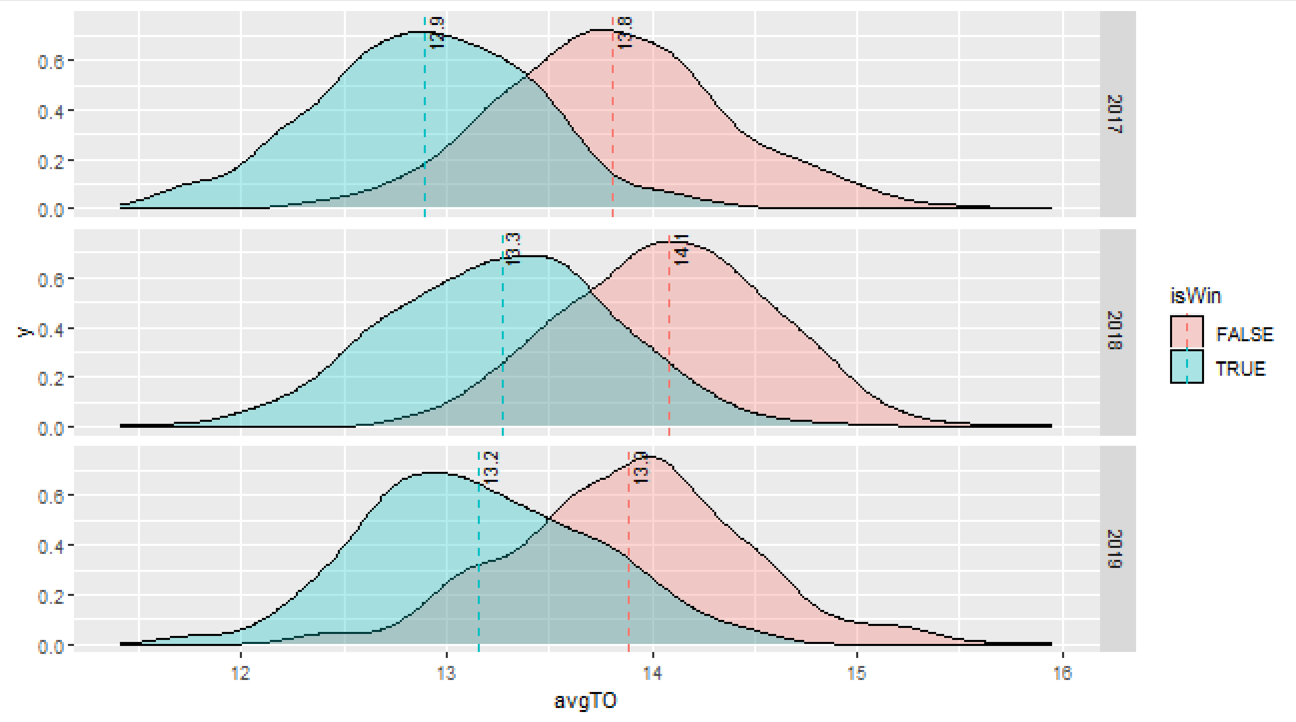
Plotting the results

```
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

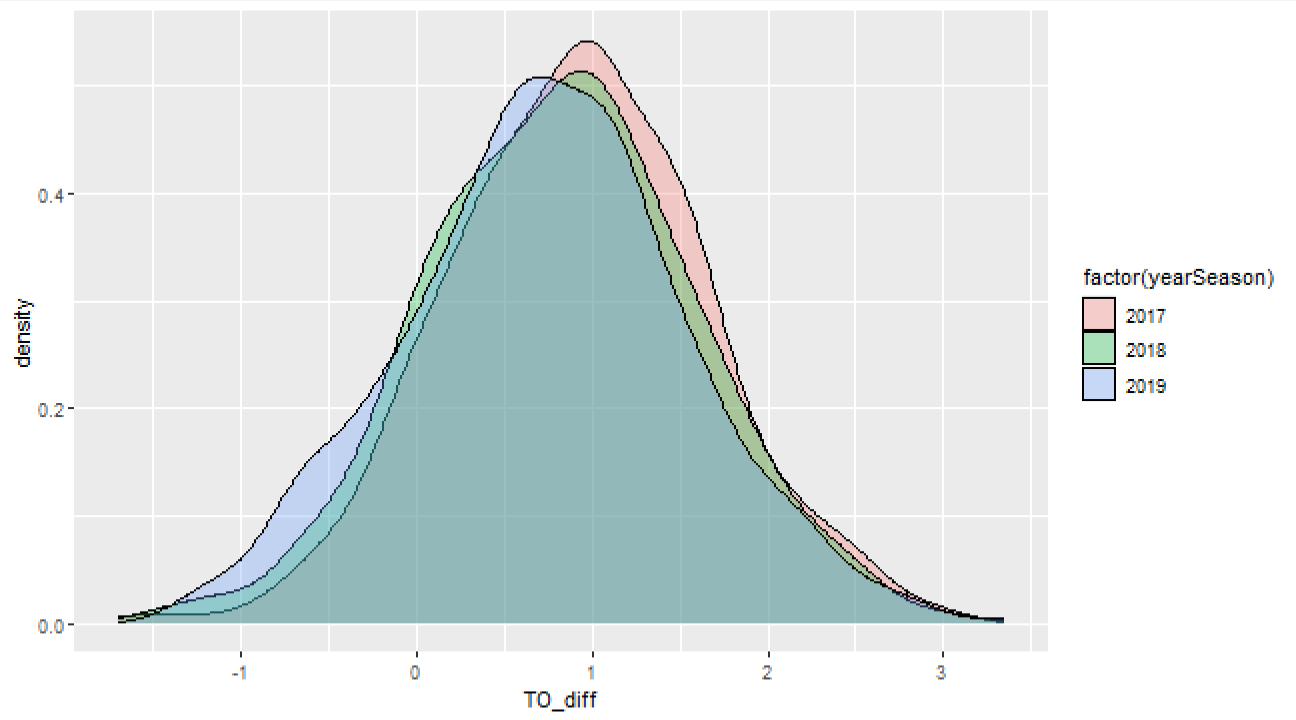
- We are *actually* interested in whether winning teams turnover the ball less
    - Science: never forget your theory / hypothesis!
- So let's actually calculate this!
- The `spread` command to create two columns

```
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        13.7       13.3   0.34
##  2       2017     2        13.7       13.0   0.641
##  3       2017     3        14.1       13.6   0.546
##  4       2017     4        13.7       12.2   1.46
##  5       2017     5        13.3       13.1   0.212
##  6       2017     6        14.8       13.2   1.58
##  7       2017     7        13.9       12.2   1.77
```

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)
```

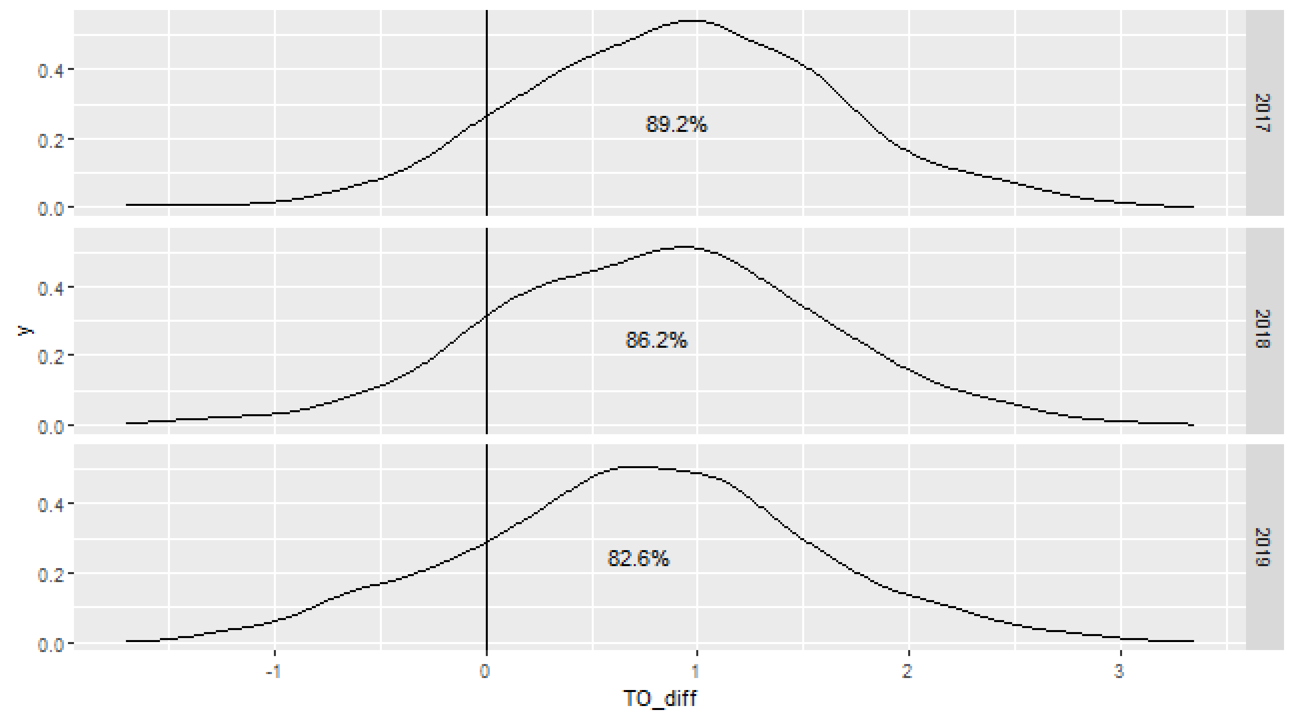


Comparing across seasons

```
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

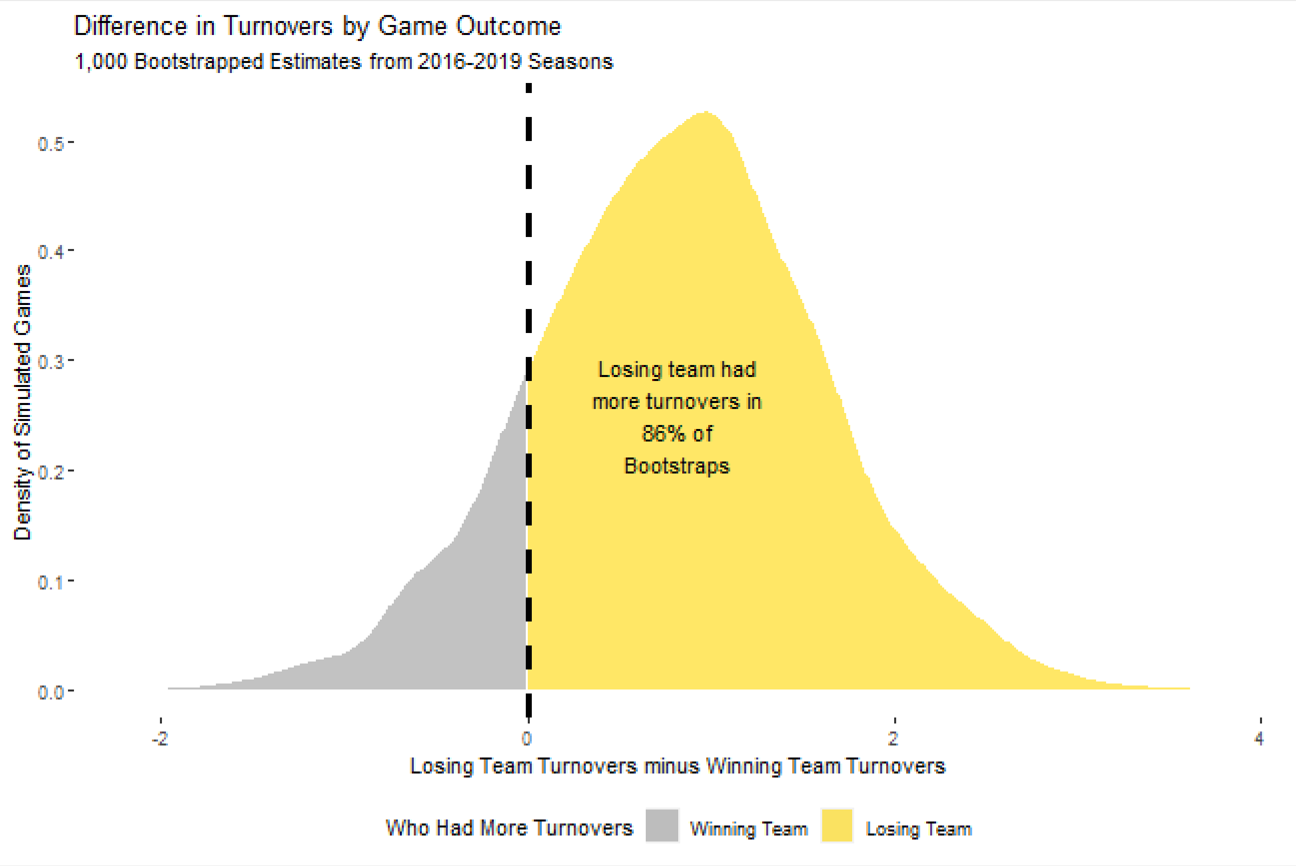
```
p
```



Visualization is **DEEP**

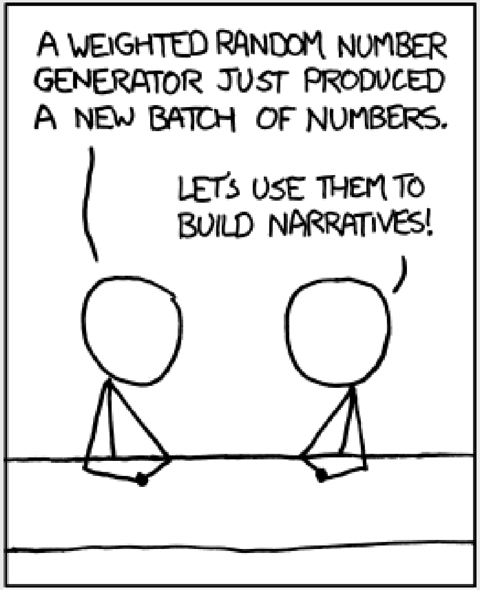
```
toplot <- bsRes %>%
  spread(isWin,avgTO,sep = '_') %>%
  mutate(TO_diff = isWin_FALSE - isWin_TRUE)

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')) +
  theme(panel.background = element_blank()
```

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 **9th** quiz
  - The password to take the quiz is ####
- **Homework:**
  1. Work through Multivariate_Analysis_part3_hw.Rmd (regression!)
  2. Finish Problem Set 4 (on Brightspace)