

Review notes

Prof. Bisbee, Vanderbilt University

2023-02-15

```
require(tidyverse)
```

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

```
## — Attaching packages — tidyverse 1.3.2 —
## ✓ ggplot2 3.3.6    ✓ purrr   0.3.4
## ✓ tibble  3.1.7    ✓ dplyr  1.0.9
## ✓ tidyr   1.2.0    ✓ stringr 1.4.0
## ✓ readr   2.1.2    ✓ forcats 0.5.1
## — Conflicts — tidyverse_conflicts() —
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()    masks stats::lag()
```

```
games <- read_rds('https://github.com/jbisbee1/DS1000_S2023/blob/main/Lectures/4_Uni_Multivariate/data/game_summary.Rds?raw=true')
```

Calculate turnovers by winning / losing by season

```
games %>%
  group_by(isWin, yearSeason) %>% # Calculating by both game outcome and season
  summarise(avgT0 = mean(tov),
            avgPts = mean(pts)) %>% # calculate average turnovers
  # spread(key = isWin, value = avgT0)
  pivot_wider(names_from = isWin, values_from = c(avgT0, avgPts))
```

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

```
## # A tibble: 3 × 5
##   yearSeason avgT0_FALSE avgT0_TRUE avgPts_FALSE avgPts_TRUE
##   <int>      <dbl>      <dbl>      <dbl>      <dbl>
## 1    2017      13.8      12.9      99.9      111.
## 2    2018      14.1      13.3     101.      112.
## 3    2019      13.9      13.1     105.      117.
```

Putting this into a bootstrap

```
# Proof of how replace works
bootstrap_result <- NULL
for(bsNumber in 1:100) {
  bootstrap_result <- games %>%
    sample_n(size = nrow(.),replace = T) %>%
    group_by(isWin) %>%
    summarise(avgTO = mean(tov)) %>%
    ungroup() %>%
    mutate(bsSimulation = bsNumber) %>%
    bind_rows(bootstrap_result)
  # print('.')
}

# Method 1: with backticks
bootstrap_result %>%
  spread(key = isWin,value = avgTO) %>%
  mutate(TO_diff = `TRUE` - `FALSE`)
```

```
## # A tibble: 100 × 4
##   bsSimulation `FALSE` `TRUE` TO_diff
##         <int>   <dbl> <dbl>   <dbl>
## 1             1    13.8  13.2 -0.671
## 2             2    13.9  13.1 -0.798
## 3             3    14.0  12.9 -1.04
## 4             4    13.8  13.1 -0.726
## 5             5    13.9  13.2 -0.719
## 6             6    13.9  13.2 -0.764
## 7             7    13.9  13.1 -0.862
## 8             8    14.0  13.3 -0.668
## 9             9    13.9  13.2 -0.746
## 10           10    14.0  13.2 -0.845
## # ... with 90 more rows
```

```
# Method 2: adjustent to spread function
bootstrap_result %>%
  spread(key = isWin,value = avgTO,sep = 'ZZZ') %>%
  mutate(TO_diff = isWinZZZTRUE - isWinZZZFALSE) %>%
  summarise(conf = mean(TO_diff < 0))
```

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

Do this for multiple teams

```

set.seed(123)
bootstrap_result <- NULL
for(bsNumber in 1:100) {
  bootstrap_result <- games %>%
    group_by(nameTeam) %>%
    # summarise(whatIsNparents = n())
    sample_n(size = n(),replace = T) %>%
    group_by(isWin,nameTeam) %>%
    summarise(avgT0 = mean(tov),
              avgPts = mean(pts),.groups = 'drop') %>%
    ungroup() %>%
    mutate(bsSimulation = bsNumber) %>%
    bind_rows(bootstrap_result)
  cat('.')
}

```

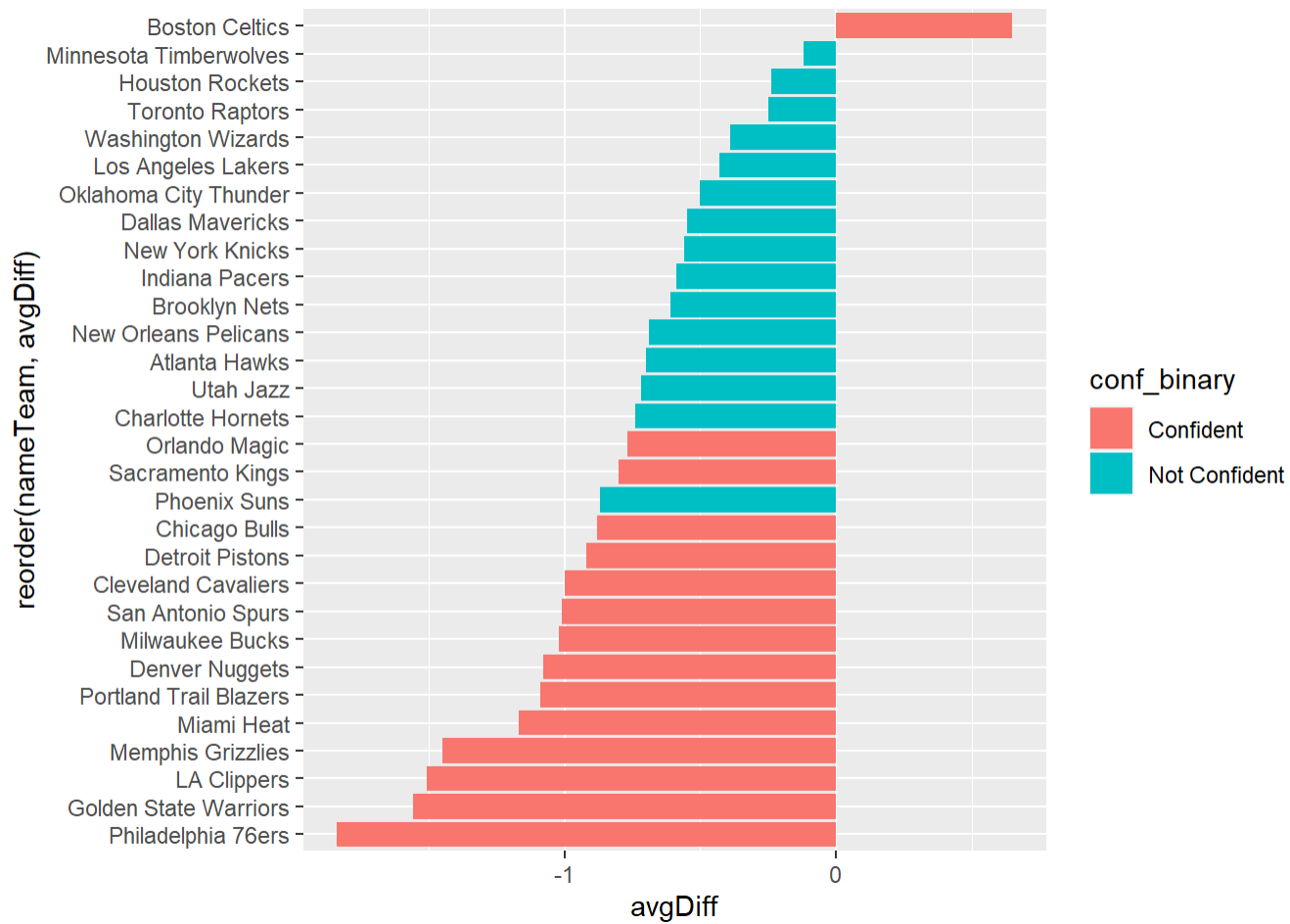
```
## .....
```

```

toplot <- bootstrap_result %>%
  pivot_wider(names_from = isWin,values_from = c(avgT0,avgPts)) %>%
  mutate(diff_T0 = avgT0_TRUE - avgT0_FALSE,
         diff_pts = avgPts_TRUE - avgPts_FALSE) %>%
  group_by(nameTeam) %>%
  summarise(conf_T0 = mean(diff_T0 < 0),
            conf_pts = mean(diff_pts > 0),
            avgDiff = mean(diff_T0),
            avgPts = mean(diff_pts))

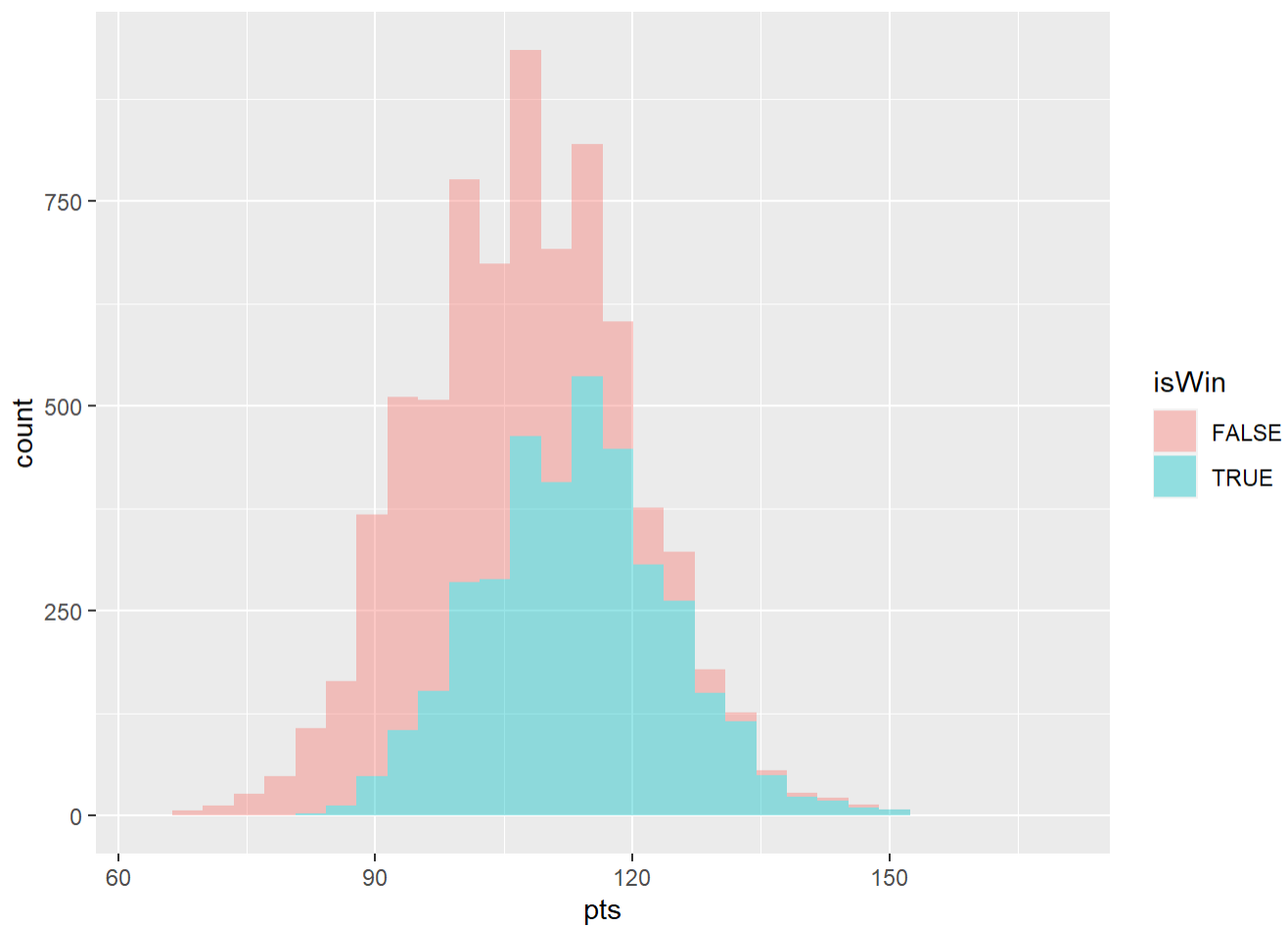
# Let's visualize
toplot %>%
  mutate(conf_binary = ifelse(conf_T0 > .95 | conf_T0 < .05,'Confident','Not Confident'),
         avgDiff = round(avgDiff,digits = 2)) %>%
  ggplot(aes(x = avgDiff,y = reorder(nameTeam,avgDiff),fill = conf_binary)) +
  geom_bar(stat = 'identity')

```



```
games %>%
  ggplot(aes(x = pts, fill = isWin)) +
  geom_histogram(alpha = .4)
```

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



```
games %>%  
  ggplot(aes(x = pts, fill = isWin)) +  
  geom_density(alpha = .4)
```

