ESPN_VAR_data

Oliver Hagger

2023-08-29

R VAR Table Data

```
library(rvest)
library(dplyr)
library(stringr)
library(tidyr)
library(readxl)
library(ggplot2)
```

Set Working Directory

```
setwd("C:~~/VAR_scraping")
```

Define the URL

```
VAR_PAGE_2020_2021 <- "https://www.espn.com/soccer/english-premier-league/story/3929823/how-var-decisio
```

Read the webpage

```
page <- read_html(VAR_PAGE_2020_2021)

VAR_team <- page %>%
  html_nodes("aside.inline.editorial.float-r") %>%
  html_text()
VAR_team <- VAR_team[2:3]

split_lines <- strsplit(VAR_team, "\n")
split_lines <- lapply(split_lines, function(x) x[-1])
print(split_lines)
```

```
## [[1]]
##  [1] "Brighton & Hove Albion 10" "Manchester United 10"
##  [3] "Crystal Palace 8"          "Leicester City 8"
```

```
##  [5] "Manchester City 8"         "Southampton 8"
##  [7] "Tottenham Hotspur 8"       "Burnley 7"
##  [9] "West Ham 6"                "AFC Bournemouth 5"
## [11] "Chelsea 5"                 "Liverpool 5"
## [13] "Arsenal 4"                 "Everton 3"
## [15] "Newcastle 3"               "Sheffield United 3"
## [17] "Aston Villa 2"             "Norwich City 2"
## [19] "Watford 2"                 "Wolves 2"
##
## [[2]]
##  [1] "West Ham 10"               "Norwich City 9"
##  [3] "Manchester City 8"         "Sheffield United 8"
##  [5] "AFC Bournemouth 7"         "Chelsea 7"
##  [7] "Leicester City 7"          "Tottenham Hotspur 7"
##  [9] "Wolves 7"                  "Arsenal 5"
## [11] "Aston Villa 5"             "Southampton 5"
## [13] "Burnley 4"                 "Crystal Palace 4"
## [15] "Everton 4"                 "Watford 4"
## [17] "Liverpool 3"               "Manchester United 3"
## [19] "Brighton & Hove Albion 2" "Newcastle 0"
```

```
decisions_for <- list(split_lines[[1]])
decisions_against <- list(split_lines[[2]])
```

Initialise empty data frame

```
data_df_for <- data.frame(team = character(0), count = character(0))
data_df_against <- data.frame(team = character(0), count = character(0))
```

Iterate through each list and extract data

```
for (lines in decisions_for) {
  team <- str_extract(lines, "([\\w\\s&]+)")
  team <- gsub("\\d+", "", team)
  count <- str_extract(lines, "\\d+")

  data_df_for <- data_df_for %>%
    add_row(team = team, count = count)
}

for (lines in decisions_against) {
  team <- str_extract(lines, "([\\w\\s&]+)")
  team <- gsub("\\d+", "", team)
  count <- str_extract(lines, "\\d+")

  data_df_against <- data_df_against %>%
    add_row(team = team, count = count)
}
```

Join the data

```
combined_data <- full_join(data_df_for, data_df_against, by = "team")


colnames(combined_data) <- c("team", "count_for", "count_against")
```

```
print(combined_data)
```

```
##                       team count_for count_against
## 1  Brighton & Hove Albion         10             2
## 2       Manchester United         10             3
## 3          Crystal Palace          8             4
## 4          Leicester City          8             7
## 5         Manchester City          8             8
## 6             Southampton          8             5
## 7       Tottenham Hotspur          8             7
## 8                 Burnley          7             4
## 9               West Ham           6            10
## 10        AFC Bournemouth          5             7
## 11                Chelsea          5             7
## 12              Liverpool          5             3
## 13                Arsenal          4             5
## 14                Everton          3             4
## 15              Newcastle          3             0
## 16       Sheffield United          3             8
## 17            Aston Villa          2             5
## 18           Norwich City          2             9
## 19                Watford          2             4
## 20                 Wolves          2             7
```

Export as a CSV file

```
combined_data$year <- '2019/2020'


year <- '2019_2020'
csv_file_name <- paste0("VAR_decisions", year, ".csv")

write.csv(combined_data, csv_file_name, row.names = T)
```

Import CSV data and bind

```
data_sheet_1 <- read_excel("C:/~~/Team2019_2020.xlsx", sheet = "Team2019_2020")
data_sheet_2 <- read_excel("C:/~~/Team2019_2020.xlsx", sheet = "Team2020_2021")
data_sheet_3 <- read_excel("C:/~~/Team2019_2020.xlsx", sheet = "Team2021_2022")
data_sheet_4 <- read_excel("C:/~~/Team2019_2020.xlsx", sheet = "Team2022_2023")
# Generate graphs
```

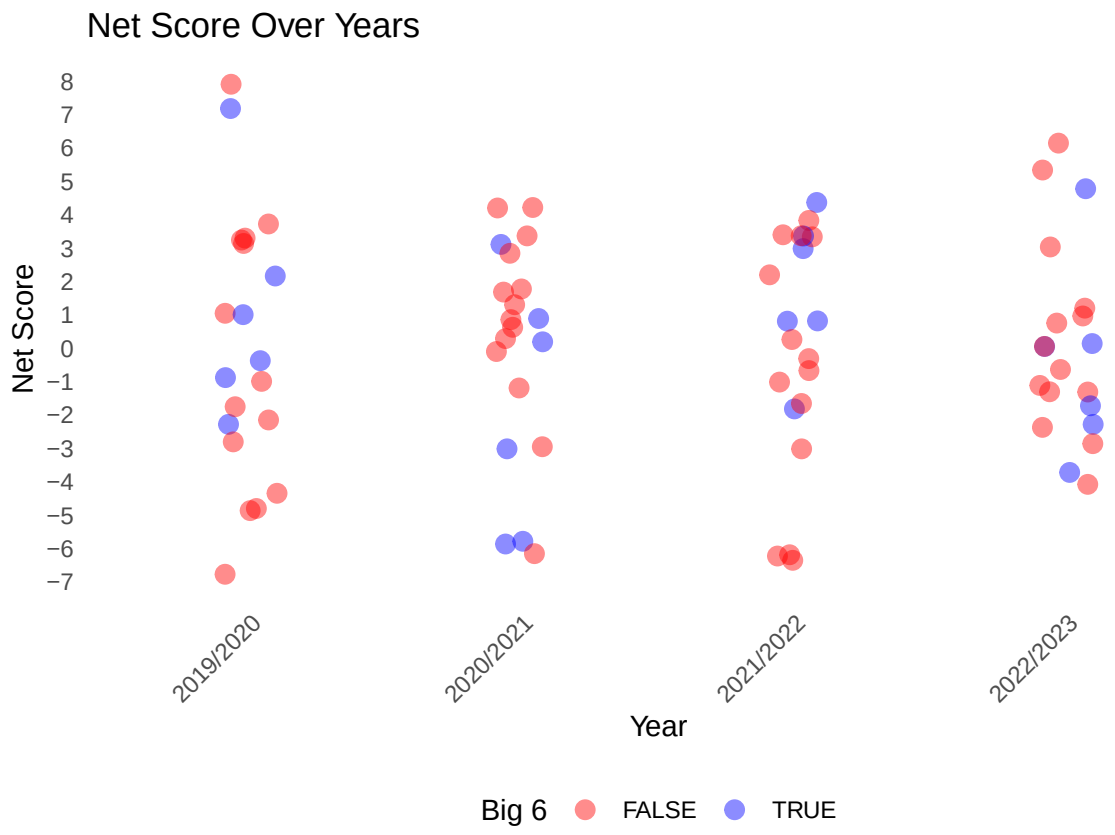
```
all_data <- bind_rows(data_sheet_1, data_sheet_2, data_sheet_3, data_sheet_4)
```

Set Big 6 Teams

```
big6_teams <- c("Arsenal", "Chelsea", "Liverpool", "Manchester City", "Manchester United", "Tottenham H
big6_data <- all_data %>% filter(team_name %in% big6_teams)
rest_of_teams_data <- all_data %>% filter(!team_name %in% big6_teams)
```

Scatter plot for net score

```
# Generate scatter plot
ggplot(all_data, aes(x = year, y = net_score, color = team_name %in% big6_teams)) +
  geom_jitter(alpha = 0.45, width = 0.1, size = 3) +
  scale_color_manual(values = c("red", "blue")) +
  labs(title = "Net Score Over Years",
       x = "Year", y = "Net Score",
       color = "Big 6") +
  theme_minimal() +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 45, hjust = 1),
        panel.grid.major = element_blank(),  # Remove major grid lines
        panel.grid.minor = element_blank(),  # Remove minor grid lines
        axis.ticks.y = element_blank()) + # Remove y-axis tick marks
        scale_y_continuous(breaks = seq(-8, 8, by = 1))  # Set y-axis tick positions and labels
```

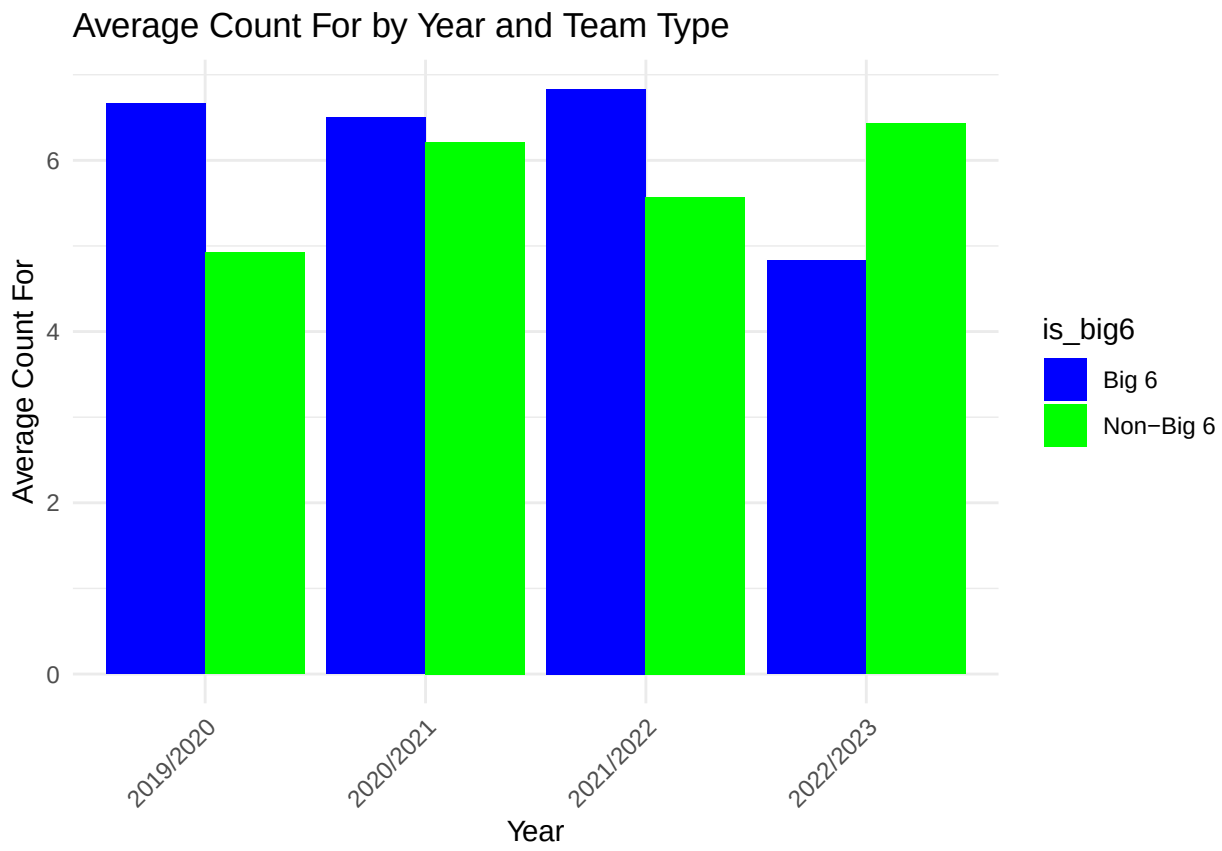


Bar plot for count for and against

```
avg_counts <- all_data %>%
  mutate(is_big6 = ifelse(team_name %in% c("Arsenal", "Chelsea", "Liverpool", "Manchester City", "Manch
  group_by(year, is_big6) %>%
  summarize(avg_count_for = mean(`Count for`))
```

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

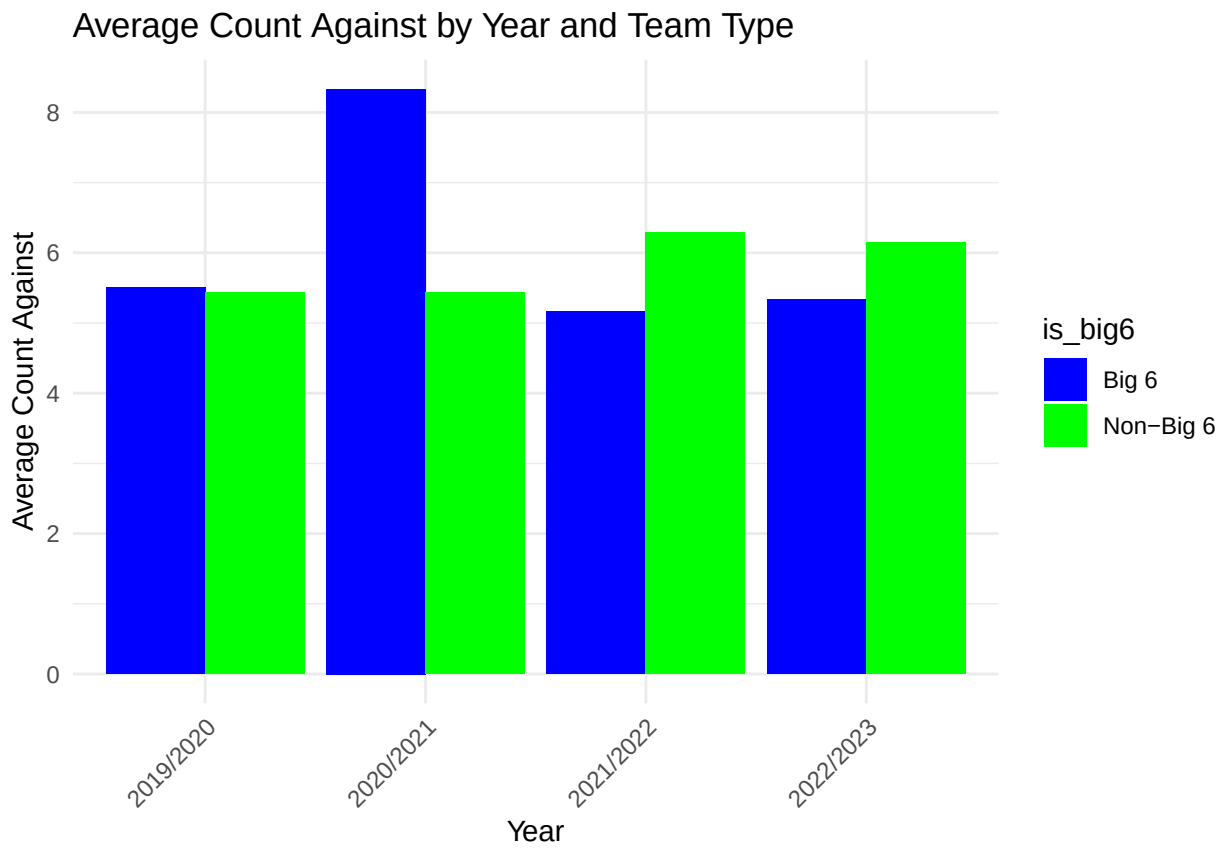
```
ggplot(avg_counts, aes(x = year, y = avg_count_for, fill = is_big6)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Average Count For by Year and Team Type",
       x = "Year",
       y = "Average Count For") +
  scale_fill_manual(values = c("Big 6" = "blue", "Non-Big 6" = "green")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
avg_counts_against <- all_data %>%
  mutate(is_big6 = ifelse(team_name %in% c("Arsenal", "Chelsea", "Liverpool", "Manchester City", "Manch
  group_by(year, is_big6) %>%
  summarize(avg_count_against = mean(`Count against`))
```

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

```
# Generate the bar chart
ggplot(avg_counts_against, aes(x = year, y = avg_count_against, fill = is_big6)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Average Count Against by Year and Team Type",
       x = "Year",
       y = "Average Count Against") +
  scale_fill_manual(values = c("Big 6" = "blue", "Non-Big 6" = "green")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

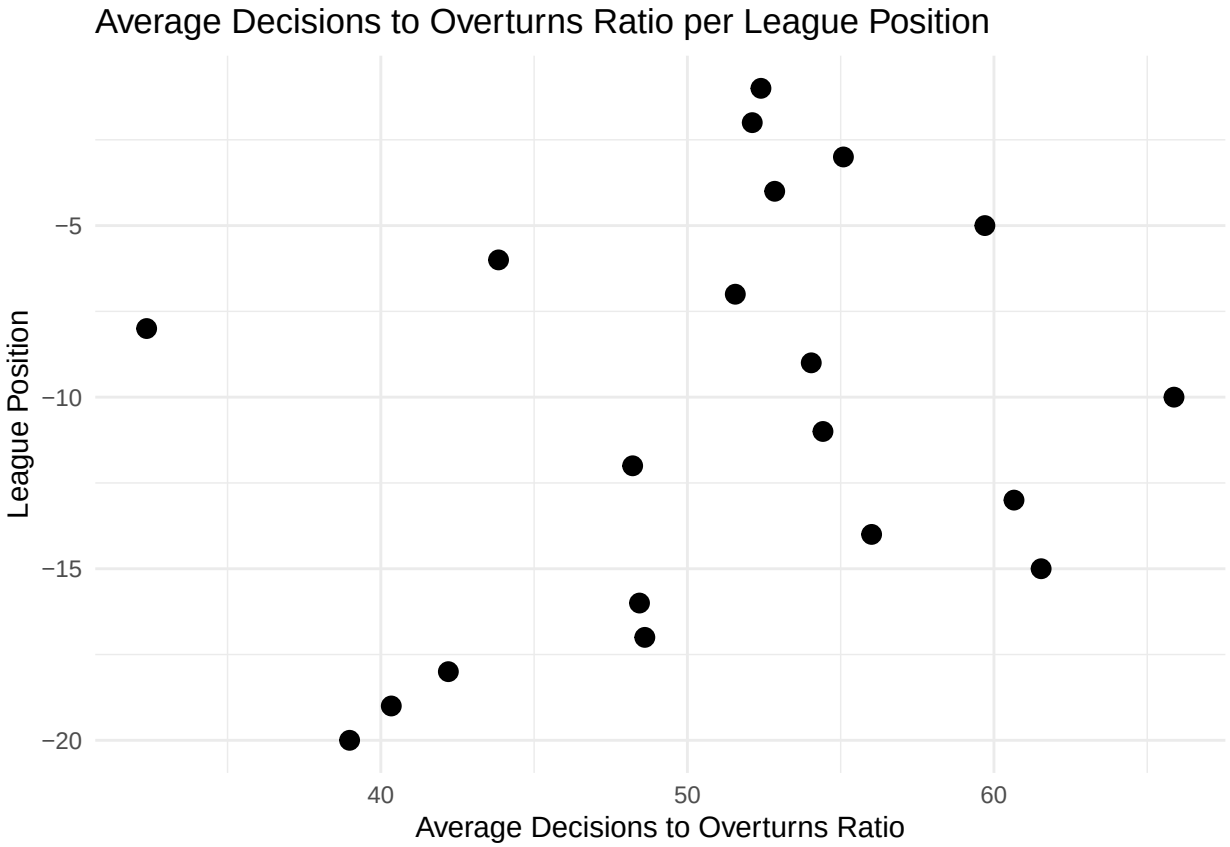


Average Decisions to Overturn Ratio per League Position

```
grouped_data <- all_data %>%
  group_by(Position) %>%
  mutate(decisions_to_overturns = (`Count for` / overturns_total) * 100)

avg_ratios <- grouped_data %>%
  group_by(Position) %>%
  summarize(avg_ratio = mean(decisions_to_overturns, na.rm = TRUE))

ggplot(avg_ratios, aes(x = avg_ratio, y = desc(Position))) +
  geom_point(size = 3) +
  labs(title = "Average Decisions to Overturns Ratio per League Position",
       x = "Average Decisions to Overturns Ratio",
       y = "League Position") +
  theme_minimal()
```


Average Decisions to Overturns Ratio per League Position
League Position
−5
−10
−15
−20
40
50
60
Average Decisions to Overturns Ratio