Group 7 Final Project

Connor Beaudry, Rishi Young, James Jeffs

STAT 27815, University of Chicago

Topic and Data Collection

What are the main questions that this presentation intends to address?

- What is the effect of `Home Team Advantage` on scoring and match outcome?
- Does being the away or home team affect how the game is played?
- Does distance traveled by the away team effect how well they play?

Data Collection

- worldfootballr is an r package made by `Jason Zivkovic`.
  - It contains functions for scraping data from FBRef, Transfermarkt, and Understat.
  - These websites provide info from all major soccer leagues on match summaries, results, shooting data and more as well as player/team data.

Comparing Different League Statistics

- Let us first analyze whether there is significant difference between leagues.
- In order to do so, we will scrape data for seasons 2016-2017 through 2022-2023 for 5 major leagues: the EPL, La Liga, Ligue 1, Bundesliga, and MLS.

```r
options(width = 80)
library(worldfootballR)

pull_league_data = function(countries) {
  output = data.frame()
  for(country in countries){
    new = fb_match_results(
      country = country,
      gender = "M",
      season_end_year = c(2017:2023),
      tier = "1st"
      )
    output = rbind(output, new)
  }
  output = output |>
    select(Country, Season_End_Year, Home, HomeGoals, Away, AwayGoals) |>
    mutate(TotalGoals = HomeGoals + AwayGoals)

  return(output)
}

League_Comparison = pull_league_data(c("ENG", "ESP", "FRA", "GER", "USA"))
```

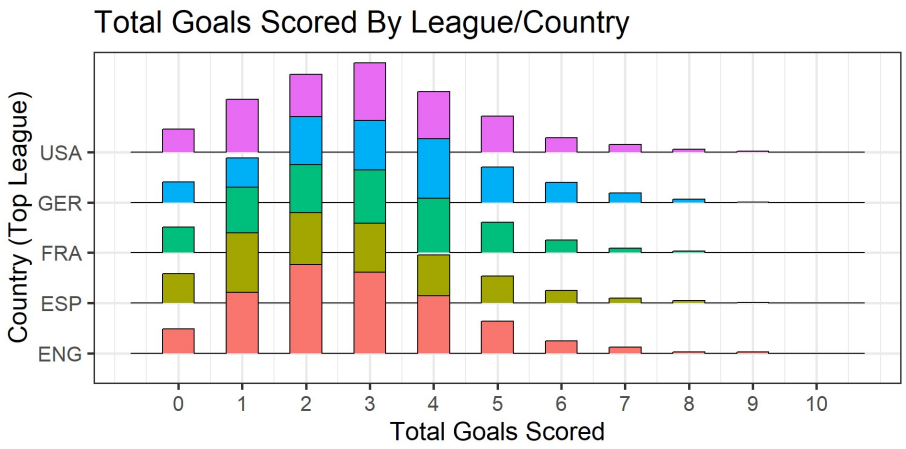
Comparing Different League Statistics

	Country	Season_End_Year	Home	HomeGoals	Away	AwayGoals	TotalGoals
412	ENG	2018	West Brom	0	Chelsea	4	4
2323	ENG	2023	Manchester City	3	Brighton	1	4
10133	GER	2023	Werder Bremen	5	Gladbach	1	6
908	ENG	2019	Everton	2	Bournemouth	0	2
2949	ESP	2017	Celta Vigo	0	Athletic Club	3	3
8154	GER	2017	Hamburger SV	2	Köln	1	3
10847	USA	2018	NYCFC	1	D.C. United	1	2
12663	USA	2023	Inter Miami	2	CF Montréal	0	2
4572	ESP	2022	Valencia	2	Mallorca	2	4
7259	FRA	2022	Paris S-G	2	Lille	1	3

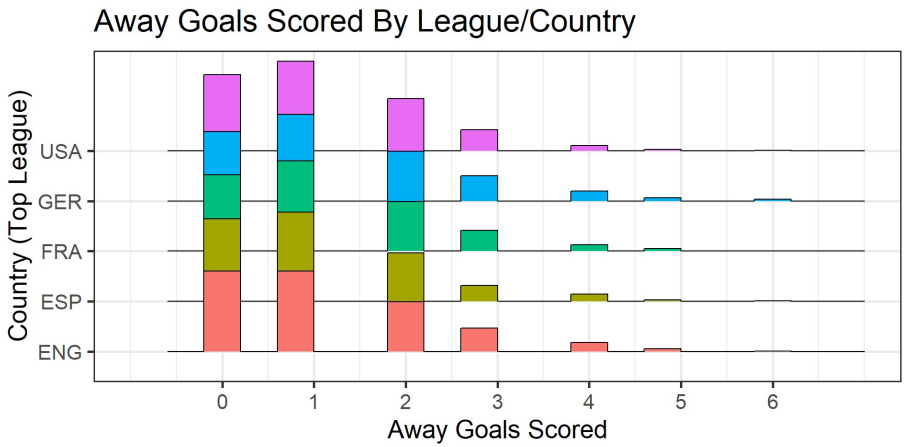
Comparing Different League Statistics

Using this data, we make a couple basic visualizations:

```
League_Comparison |>
ggplot(aes(x = TotalGoals, y = Country, height = stat(dens
  geom_density_ridges(stat = "binline", bins = 21) +
  guides(fill = "none") +
  labs(x = "Total Goals Scored", y = "Country (Top League)"
  expand_limits(y = 7) +
  scale_x_continuous(breaks = 0:10) +
  theme_bw(base_size = 20)
```

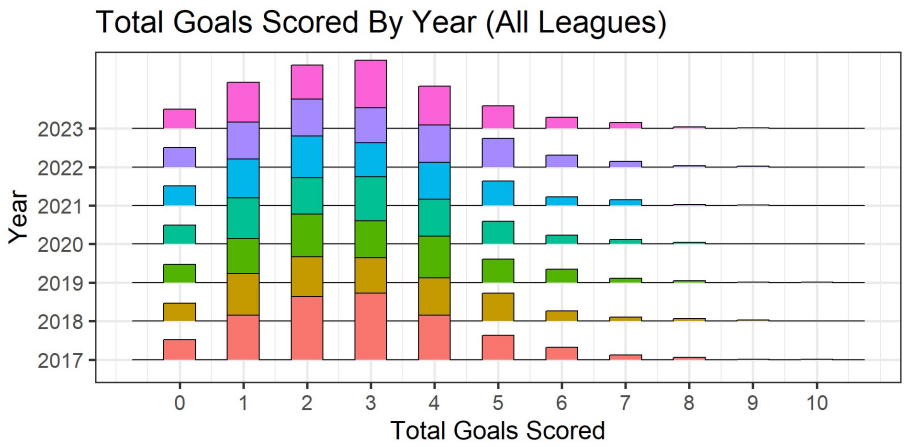


```r
League_Comparison |>
ggplot(aes(x = AwayGoals, y = Country, height = stat(densi
  geom_density_ridges(stat = "binline", bins = 21) +
  guides(fill = "none") +
  labs(x = "Away Goals Scored", y = "Country (Top League)",
  expand_limits(y = 7) +
  scale_x_continuous(breaks = 0:6, limits = c(-1, 7)) +
  theme_bw(base_size = 20)
```



Comparing Different League Statistics

```
1  library(GGally)
2  League_Comparison |>
3    ggplot(mapping = aes(x = TotalGoals, y = factor(Season_E
4    geom_density_ridges(stat = "binline", bins = 21) +
5    guides(fill = "none") +
6    labs(y = "Year", x = "Total Goals Scored", title = "Tota
7    expand_limits(y = 9) +
8    scale_x_continuous(breaks = 0:10) +
9    theme_bw(base_size = 20)
```



The English Premier League

The English Premier League

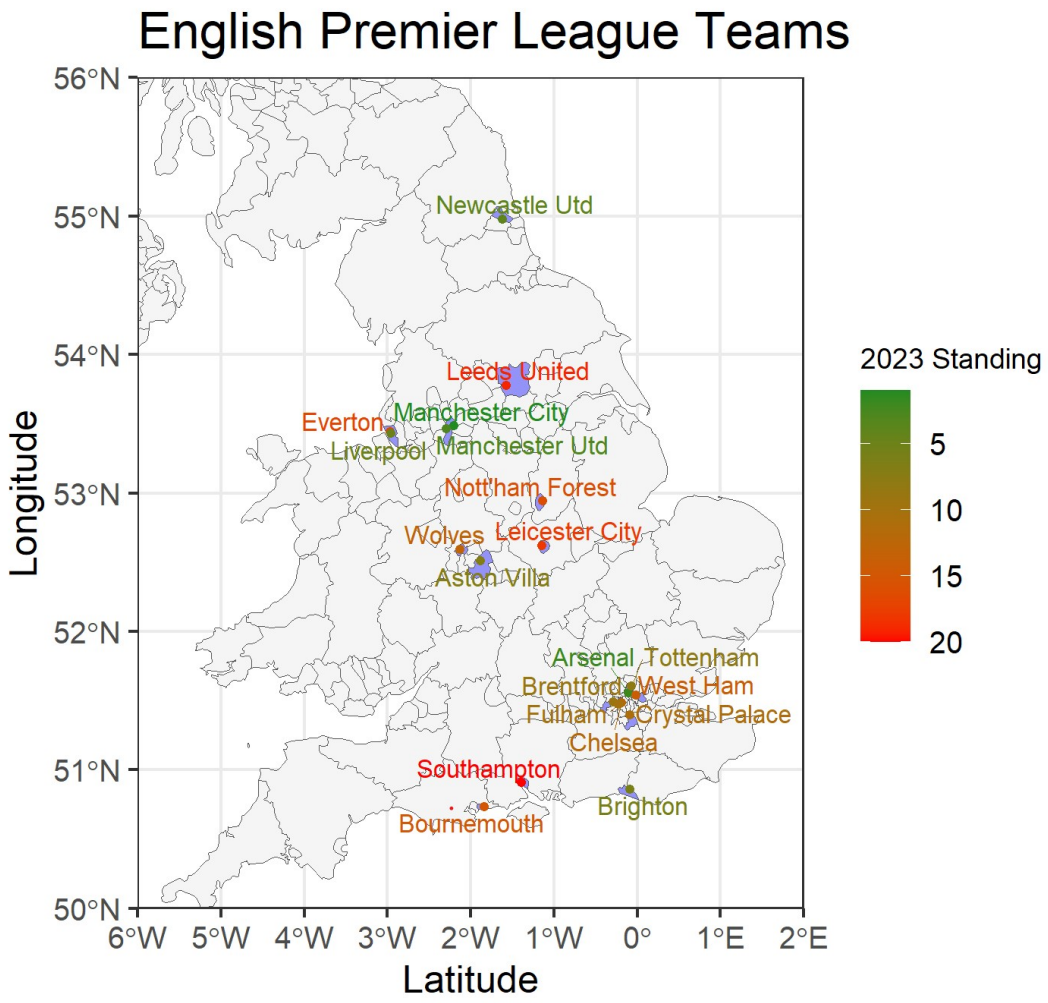
- Making a map of **The English Premier League** (EPL):

```
library(rnaturalearth)
library(rnaturalearthdata)
library(rnaturalearthhires)
library(ggrepel)

locations_2023 = read_csv(file = "data/Team Stadium Locations EPL 2023.csv", show_col_types = FALSE)
UK_map = ne_states(country = "United Kingdom", returnclass = "sf")
counties = c("Newcastle upon Tyne", "Leeds", "Manchester", "Liverpool",
             "Wolverhampton", "Nottingham", "Leicester", "Birmingham",
             "Southampton", "Bournemouth", "Brighton and Hove",
             "Islington", "Hounslow", "Haringey", "Hammersmith and Fulham",
             "Newham", "Croydon")
UK_map_teams = filter(UK_map, name %in% counties)

ggplot(data = UK_map) +
  geom_sf(fill = "whitesmoke") +
  geom_sf(data = UK_map_teams, fill = "blue", alpha = 0.4) +
  theme_bw(base_size = 30) +
  theme(axis.title = element_text(size = 30),
        legend.title = element_text(size = 22),
        legend.key.size = unit(40, "pt"),
        plot.title = element_text(size = 40)) +
  geom_point(data = locations_2023, aes(x = Latitude,
                                        y = Longitude,
                                        color = Standing),
             size = 3) +
  geom_text_repel(data = locations_2023,
                  aes(x = Latitude,
                      y = Longitude,
                      color = Standing,
                      label = `Team Name`),
                  size = 7, max.overlaps = 15, force = 2) +
  scale_colour_gradient(low = "forestgreen", high = "red",
```

```
                      guide = guide_colorbar(reverse = TRUE)) +
  coord_sf(xlim = c(-6, 2), ylim = c(50, 56), expand = FALSE) +
  labs(title = "English Premier League Teams",
```

Stadium coordinate data was acquired manually from the EPL website.

The English Premier League



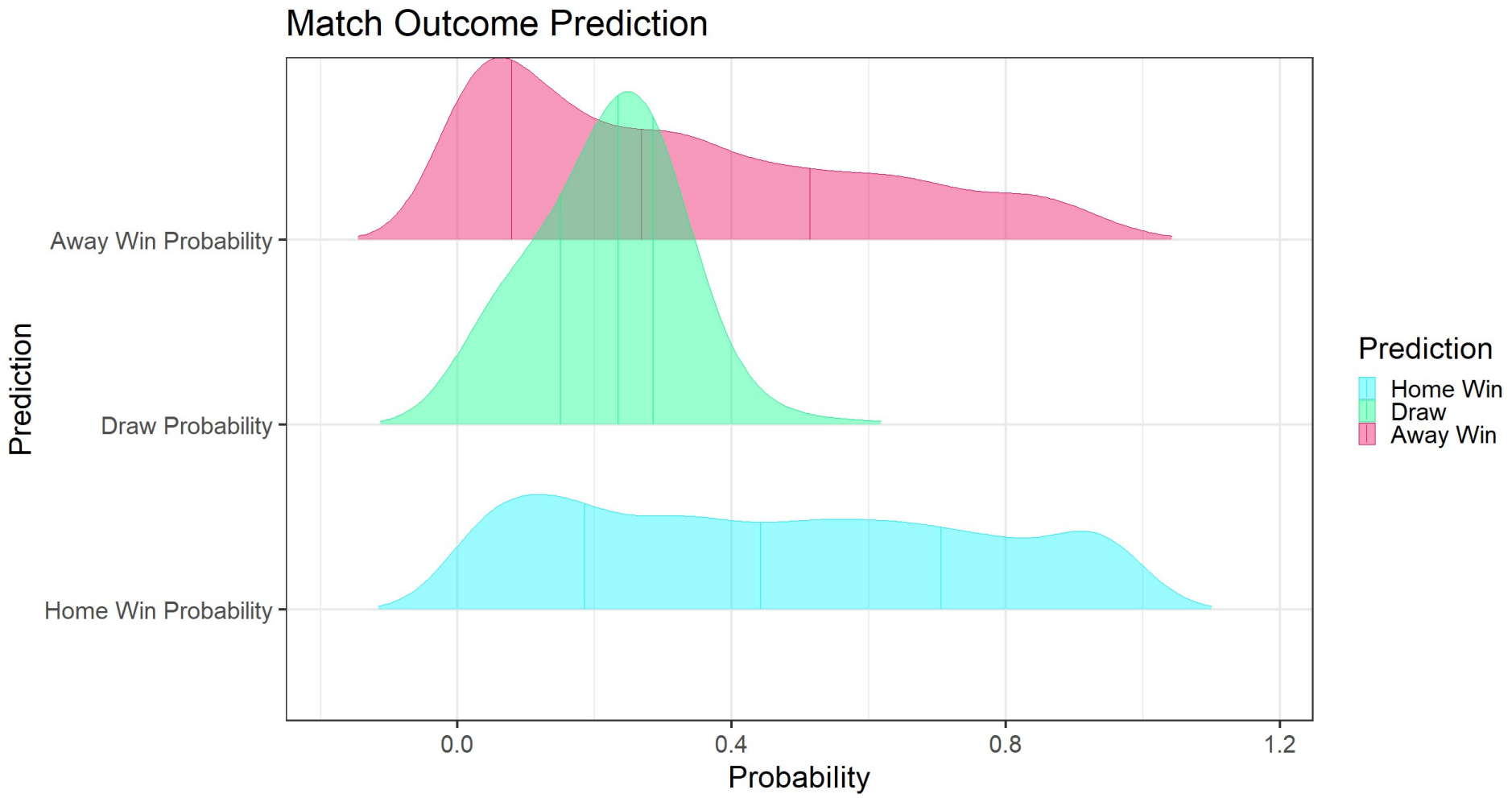
Home Team Advantage

- We will begin our analysis of the EPL 2022-23 season by determining how much of an advantage the home team gets.

```r
library(ggridges)
options(width = 100)
results <- understat_league_match_results(league = "EPL",
                                          season_start_yea
results <- results |>
  rename(`Home Win Probability` = forecast_win,
         `Draw Probability` = forecast_draw,
         `Away Win Probability` = forecast_loss)

long_data <- pivot_longer(
  results,
  cols = c(`Home Win Probability`,
           `Draw Probability`,
           `Away Win Probability`),
  names_to = "Prediction",
  values_to = "probability"
)

long_data$Prediction <- factor(
  long_data$Prediction,
  levels = c("Home Win Probability",
             "Draw Probability",
             "Away Win Probability")
)
```

```r
ggplot(long_data, aes(x = probability, y = Prediction,
                      fill = Prediction, color = Predictio
  geom_density_ridges(
    alpha = 0.4,
    rel_min_height = 0.01,
    quantile_lines = TRUE,
    quantiles = c(0.25, 0.5, 0.75)
  ) +
  labs(
    title = "Match Outcome Prediction",
    x = "Probability",
    y = "Prediction"
  ) +
  scale_fill_manual(
    values = c(`Home Win Probability` = "#04f5ff",
               `Draw Probability` = "#00ff85",
               `Away Win Probability` = "#e90052"),
    labels = c("Home Win", "Draw", "Away Win")
  ) +
  scale_color_manual(
    values = c(`Home Win Probability` = "#04f5ff",
               `Draw Probability` = "#00ff85",
               `Away Win Probability` = "#e90052"),
    labels = c("Home Win", "Draw", "Away Win")
  ) +
  theme_ridges() +
  theme(
    plot.title = element_text(hjust = 0.5, margin = margin
    axis.title.x = element_text(hjust = 0.5, margin = marg
    axis.title.y = element_text(hjust = 0.5, margin = marg
    axis.text.y = element_blank(),
    axis.ticks.y = element_blank()
```

Home Team Advantage



Away Team Travel Effect

- Data on the travel distance of Premier League clubs in the 2022-23 season was acquired from footballteamnews.

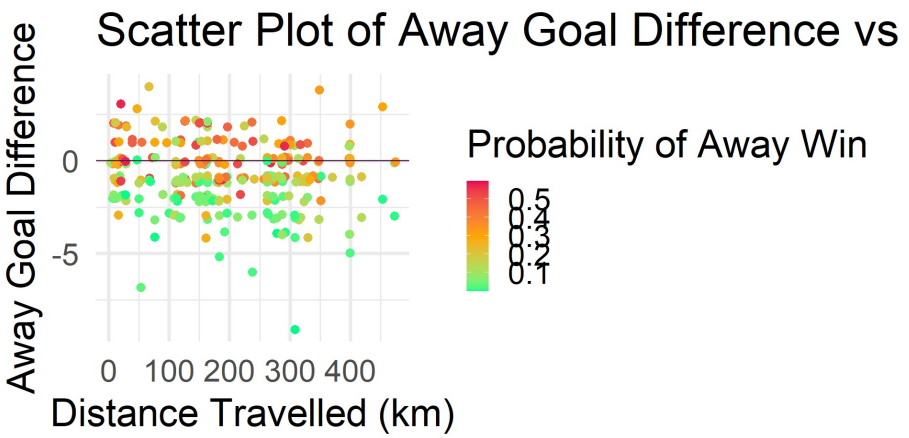
```
distances <- read_csv("data/England_Distances - 2023.csv")
```

```
find_distance <- function(home_team, away_team, distance_ma
  teams <- colnames(distance_matrix)[2:21]
  home_index <- which(teams == home_team)
  away_index <- which(teams == away_team)
  if(length(home_index) > 0 && length(away_index) > 0) {
    return(pull(distance_matrix[home_index, away_index + 1]
  } else {
    return(NA)
  }
}
modified_results <- results |>
  rowwise() |>
  mutate(
    Distance_Travelled = find_distance(home_team,
                                       away_team, distances
    Away_Goal_Diff = away_goals - home_goals
  ) |>
  ungroup()
```

Away Team Travel Effect

- Data was filtered to include games that did not have a high probability of the away team winning.

```r
filtered_results <- modified_results |>
  filter(`Away Win Probability` < 0.6)

ggplot(filtered_results, aes(x = Distance_Travelled,
                             y = Away_Goal_Diff)) +
  geom_jitter(aes(color = `Away Win Probability`), size = 
                  width = 0.2, height = 0.2) +
    scale_color_gradient2(low = "#00ff85",
                          mid = "orange",
                          high = "#e90052",
                          midpoint = 0.3,
                          name = "Probability of Away Win"
  geom_hline(yintercept = 0, color = "#38003c") +
  labs(title = "Scatter Plot of Away Goal Difference vs Dis
       x = "Distance Travelled (km)",
       y = "Away Goal Difference") +
  theme_minimal(base_size = 30)
```



Away Team Effect on Shotstyle

Away Team Effect on Shotstyle

- Analyzing the effects of being the away team on shooting:

```
1 EPL_Shots <- understat_league_season_shots(league = "EPL", season_start_year = 2022)
```

	X	Y	xG	h_a	home_team	away_team	league	id	minute	result
2078	0.981	0.629	0.3262840	h	Brighton	Tottenham	EPL	491814	68	SavedShot
7361	0.978	0.428	0.0756614	a	Southampton	Manchester City	EPL	518223	40	MissedShots
6785	0.863	0.507	0.1013050	a	Arsenal	Leeds	EPL	516429	64	MissedShots
664	0.899	0.515	0.0207121	h	Leeds	Chelsea	EPL	482403	28	MissedShots
171	0.926	0.328	0.0705736	h	Manchester United	Brighton	EPL	479580	6	BlockedShot
1778	0.858	0.478	0.0583403	a	Fulham	Newcastle United	EPL	490163	4	SavedShot
4946	0.916	0.486	0.1384490	h	Arsenal	Manchester	EPL	505089	63	BlockedShot

	X	Y	xG	h_a	home_team	away_team	league	id	minute	result
						United				
3440	0.842	0.752	0.0328334	a	Tottenham	Liverpool	EPL	498841	35	BlockedShot
3549	0.866	0.626	0.0564314	h	Nottingham Forest	Crystal Palace	EPL	500158	73	MissedShots
2975	0.945	0.533	0.1269200	a	Bournemouth	Tottenham	EPL	496616	91	BlockedShot

Away Team Effect on Shotstyle

- To create a proper heat-map, we need to wrangle our data into the proper shape:

```
create_heatmap_data <- function(data) {
  x_breaks <- seq(0.65, 1, by = 0.025)
  y_breaks <- seq(0.35, 0.65, by = 0.025)
  grid <- expand.grid(xmin = head(x_breaks, -1),
                      xmax = tail(x_breaks, -1),
                      ymin = head(y_breaks, -1),
                      ymax = tail(y_breaks, -1))
  heatmap_data <- grid |>
    rowwise() |>
    mutate(shot_count = sum(data$X >= xmin & data$X < xmax
                            data$Y < ymax),
           goal_count = sum(data$X >= xmin & data$X < xmax
                            data$Y < ymax & data$result =
           goal_percentage = ifelse(shot_count > 0,
                                    (goal_count / shot_cou
      ungroup()
  return(heatmap_data)
}
```

```
home_shots <- EPL_Shots |> filter(h_a == "h")
away_shots <- EPL_Shots |> filter(h_a == "a")
home_shots_filtered <- home_shots |>
  filter(X >= 0.65 & X <= 1, Y >= 0.35 & Y <= 0.65)

away_shots_filtered <- away_shots |>
  filter(X >= 0.65 & X <= 1, Y >= 0.35 & Y <= 0.65)

home_heatmap_data <- create_heatmap_data(home_shots_filtere
away_heatmap_data <- create_heatmap_data(away_shots_filtere
home_heatmap_data <- home_heatmap_data |>
  rename(home_goal_percentage = goal_percentage)
away_heatmap_data <- away_heatmap_data |>
  rename(away_goal_percentage = goal_percentage)
merged_data <- full_join(
  home_heatmap_data,
  away_heatmap_data,
  by = c("xmin", "xmax", "ymin", "ymax")
)
merged_data <- merged_data |>
  mutate(
    goal_percentage_diff = coalesce(home_goal_percentage, (
                           coalesce(away_goal_percentage, (
  )
goal_percentage_diff_data <- merged_data |>
  select(xmin, xmax, ymin, ymax, home_goal_percentage,
         away_goal_percentage, goal_percentage_diff)
```

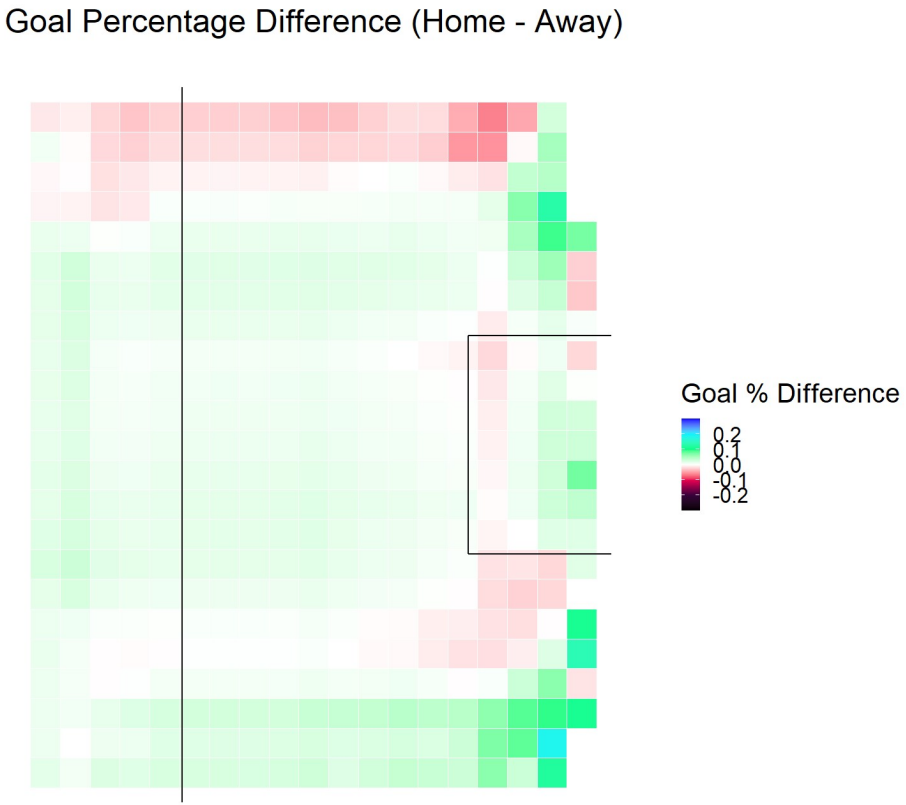
Away Team Effect on Shotstyle

- Then we make the heat-map function for the percentage of shots that are made:

```r
create_heatmap_plot <- function(data, title, colors, values
  data <- data |>
    mutate(
      x = (xmin + xmax) / 2,
      y = (ymin + ymax) / 2
    )

  ggplot(data, aes(x = x, y = y)) +
    geom_tile(aes(fill = goal_percentage_diff), color = "wh
    scale_fill_gradientn(
      colors = colors,
      limits = c(-0.3, 0.3),
      name = "Goal % Difference"
    ) +
    geom_segment(
      aes(x = 0.82, xend = 0.82, y = 0.35, yend = 0.65),
      color = "black", linewidth = 0.75
    ) +
    geom_segment(
      aes(x = 0.94, xend = 1, y = 0.5458, yend = 0.5458),
      color = "black", linewidth = 0.75
    ) +
    geom_segment(
      aes(x = 0.94, xend = 1, y = 0.4542, yend = 0.4542),
      color = "black", linewidth = 0.75
    ) +
    geom_segment(
      aes(x = 0.94, xend = 0.94, y = 0.5458, yend = 0.4542
      color = "black", linewidth = 0.75
    ) +
    labs(title = title) +
    theme_minimal(base_size = 25) +
    theme(
      panel.background = element_blank(),
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      axis.text.x = element_blank()
```

```r
goal_percentage_diff_range <- range(goal_percentage_diff_da

p_colors <- c("black", EPL_Colors[5], EPL_Colors[2], EPL_Co
              EPL_Colors[4], EPL_Colors[1], "blue")
p_values <- c(-0.3, -0.1, -0.05, 0, 0.05, 0.1, 0.3)
```

Away Team Effect on Shotstyle



Away Team Effect on Shotstyle

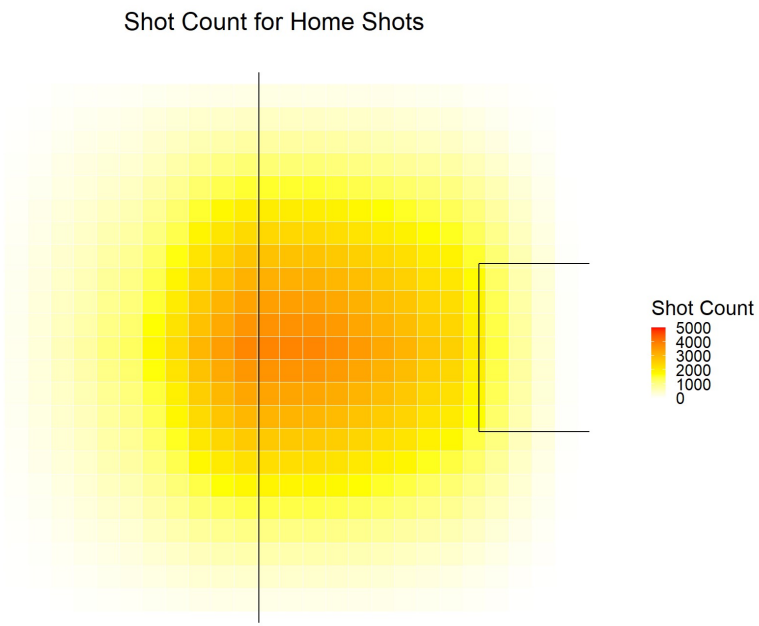
- Using a similar function, we can produce the heat-maps for the number of shots taken:

```r
create_shot_heatmap_data <- function(data) {
  x_breaks <- seq(0.65, 1, by = 0.025)
  y_breaks <- seq(0.35, 0.65, by = 0.025)
  grid <- expand.grid(xmin = head(x_breaks, -1),
                      xmax = tail(x_breaks, -1),
                      ymin = head(y_breaks, -1),
                      ymax = tail(y_breaks, -1))
  heatmap_data <- grid |>
    rowwise() |>
    mutate(shot_count = sum(data$X >= xmin & data$X < xmax
    ungroup()

  return(heatmap_data)
}

home_shots_filtered <- home_shots |> filter(X >= 0.65 & X <
away_shots_filtered <- away_shots |> filter(X >= 0.65 & X <
home_shot_heatmap_data <- create_shot_heatmap_data(home_sh
away_shot_heatmap_data <- create_shot_heatmap_data(away_sh
mean_shot_count <- mean(home_shot_heatmap_data$shot_count)
sd_shot_count <- sd(home_shot_heatmap_data$shot_count)
shot_count_colors <- c("white", "yellow", "orange","red")
```

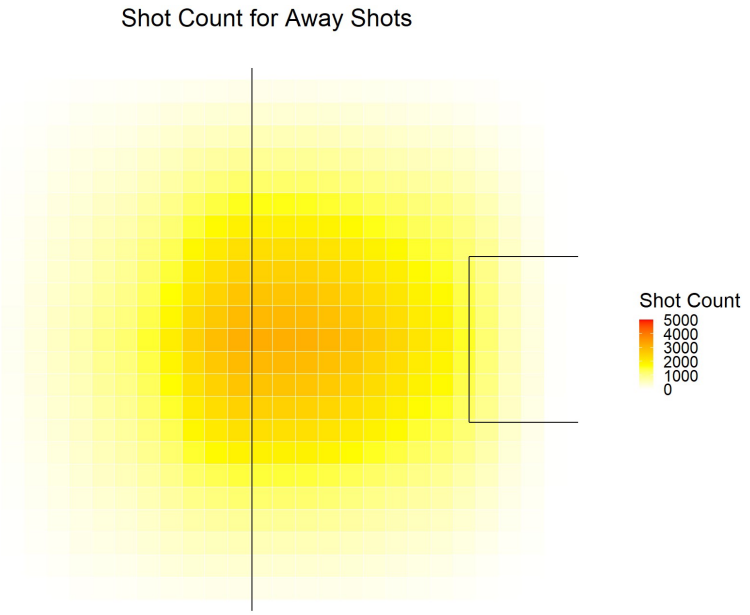
```r
create_shot_heatmap_plot <- function(data, title, colors)
  data <- data |>
    mutate(x = (xmin + xmax) / 2,
           y = (ymin + ymax) / 2)

  ggplot(data, aes(x = x, y = y)) +
    geom_tile(aes(fill = shot_count), color = "white") +
    scale_fill_gradientn(
      colors = colors,
      values = scales::rescale(c(0, 1, 2, 3)),
      limits = c(0, 5000),
      breaks = seq(0, 5000, by = 1000),
      name = "Shot Count"
    ) +
    labs(title = title) +
    geom_segment(aes(x = 0.82, xend = 0.82, y = 0.35, yend
                 color = "black", size = 0.75) +
    geom_segment(aes(x = 0.94, xend = 1, y = 0.5458, yend =
                 color = "black", size = 0.75) +
    geom_segment(aes(x = 0.94, xend = 1, y = 0.4542, yend =
                 color = "black", size = 0.75) +
    geom_segment(aes(x = 0.94, xend = 0.94, y = 0.5458, ye
                 color = "black", size = 0.75) +
    coord_fixed(ratio = 1) +
    theme_minimal(base_size = 25) +
    theme(
      panel.background = element_blank(),
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      axis.text.x = element_blank(),
      axis.text.y = element_blank(),
      axis.ticks = element_blank(),
      axis.title.x = element_blank(),
      axis.title.y = element_blank(),
      plot.title = element_text(hjust = 0.5)
    )
}
```

Away Team Effect on Shotstyle

- For home teams:



- For away teams:



Home-Away Points Breakdown

Home-Away Points Breakdown

- In soccer, a team's points over the season are calculated as `three times their number of wins plus their number of draws`.
- We will now compare the number of points earned at home games vs away games:

```
games <- results |>
  select(home_team, away_team, home_goals, away_goals) |>
  distinct()
unique_games <- results |>
  select(home_team, away_team, home_goals, away_goals) |>
  distinct()
unique_games <- unique_games |>
  mutate(
    home_points = if_else(home_goals > away_goals, 3, if_else(home_goals == away_goals, 1, 0)),
    away_points = if_else(away_goals > home_goals, 3, if_else(away_goals == home_goals, 1, 0))
  )
```

Home-Away Points Breakdown

Some more data manipulation:

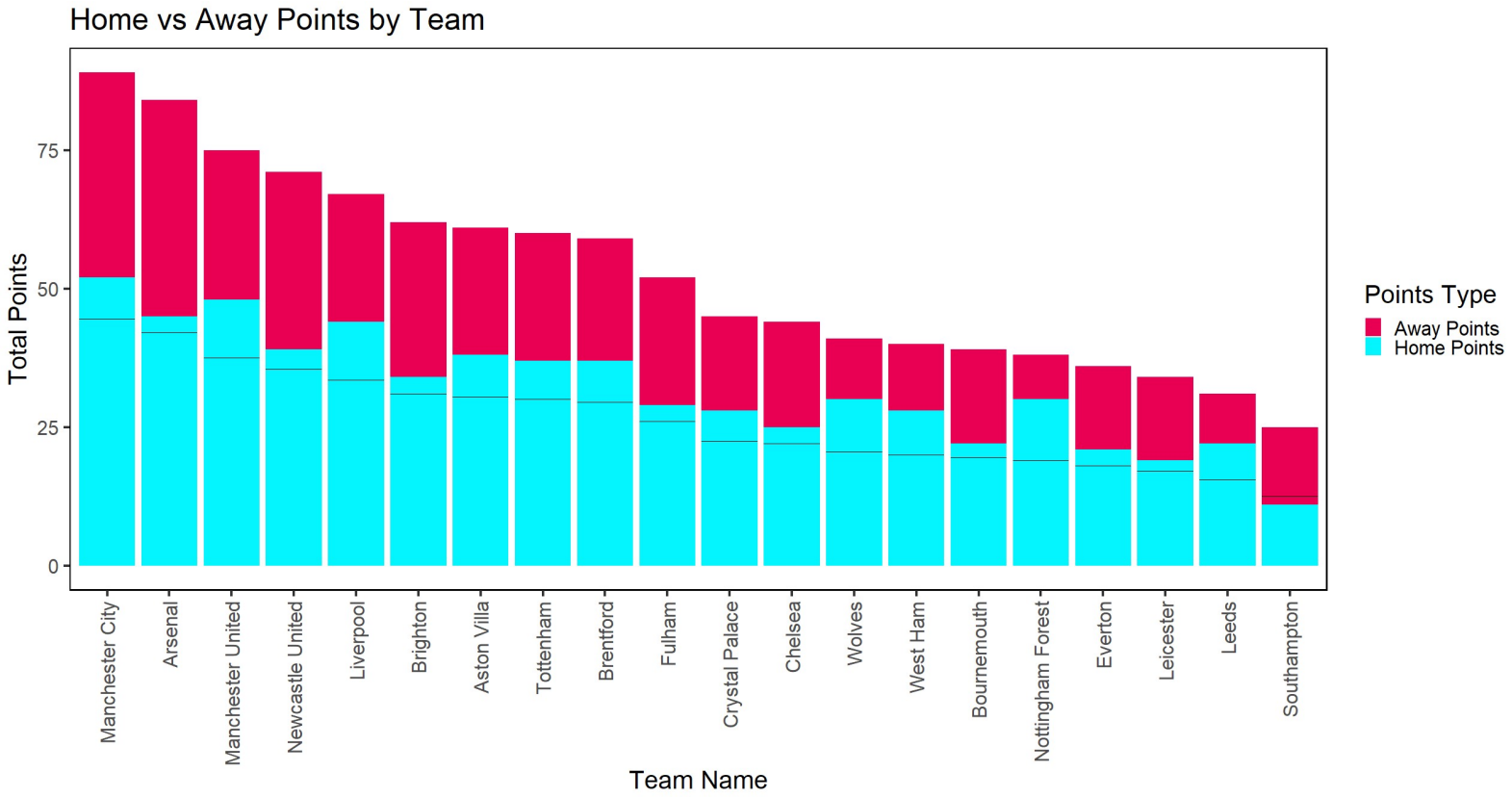
```r
home_points_df <- unique_games |>
  group_by(home_team) |>
  summarize(home_points = sum(home_points, na.rm = TRUE)) |>
  rename(team = home_team)
away_points_df <- unique_games |>
  group_by(away_team) |>
  summarize(away_points = sum(away_points, na.rm = TRUE)) |>
  rename(team = away_team)
team_points_df <- full_join(home_points_df, away_points_df, by = "team")
team_points_df <- team_points_df |>
  mutate(
    home_points = replace_na(home_points, 0),
    away_points = replace_na(away_points, 0),
    total_points = home_points + away_points,
    proportion = home_points / away_points,
    half_points = total_points / 2
  )
team_points_df <- team_points_df |>
  mutate(proportion = ifelse(is.infinite(proportion), NA, proportion))
team_points_df <- team_points_df |>
  arrange(desc(proportion))
```

Home-Away Points Breakdown

- The proportion of points won at home games to those won at away games:

```r
team_points_df <- team_points_df |>
  arrange(desc(total_points))  |>
mutate(team = ifelse(team == "Wolverhampton Wanderers", "Wolves", team))

team_points_df <- team_points_df |>
  arrange(desc(total_points))
long_team_points_df <- team_points_df |>
  pivot_longer(cols = c(home_points, away_points),
               names_to = "type", values_to = "points")
long_team_points_df$team <- factor(long_team_points_df$team,
                                   levels = team_points_df$team)

ggplot(long_team_points_df, aes(x = team, y = points, fill = type)) +
  geom_bar(stat = "identity") +
  geom_errorbar(aes(ymin = half_points, ymax = half_points),
                width = 0.9, color = "black") +
  scale_fill_manual(values = c("home_points" = "#04f5ff",
                               "away_points" = "#e90052"),
                    labels = c("home_points" = "Home Points",
                               "away_points" = "Away Points")) +
  labs(
    title = "Home vs Away Points by Team",
    x = "Team Name",
    y = "Total Points",
    fill = "Points Type"
  ) +
  theme_bw(base_size = 25) +
  theme(axis.text.x = element_text(angle = 90, vjust = .5, hjust = 1),
        panel.background = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_rect(color = "black", fill = NA))
```

Home-Away Points Breakdown



Salary Analysis

Salary Data

Salary Data

```r
premier_league_salaries  = premier_league_salaries |>
  mutate(
    Pos = str_sub(Pos, 1, 2),
    Pos = factor(Pos, levels = c("GK", "DF", "MF", "FW"))
  ) |>
  filter(Pos != "")
```

```r
highest_earners = premier_league_salaries|>
  group_by(Pos) |>
  slice(which.max(AnnualWageUSD))
```

Salary Data

```
ggplot(premier_league_salaries, aes(x = AnnualWageUSD, y = Pos, color = Pos)) +
  geom_point(size = 3) +
  labs(
  title = "Premier League '23 Salaries by Position",
  x = "Salary (USD)",
  y = "Position"
  ) +
  theme_minimal() +
  geom_text(data = highest_earners,
            aes(label = paste(Player, Team, sep = "\n")),
            vjust = 1.5,
            color = "black",
            size = 4.5
            ) +
  guides(color = "none") +
  scale_x_continuous(labels = unit_format(unit = "M", scale = 1e-6)) +
  expand_limits(x = 38000000, y = 0) +
  theme_bw(base_size = 20)
```