STAT 442: Final Project

Abhay Sharma

Writing to Tie Data and Visualizations Together:

This project provides an in-depth analysis of the 2017–2018 English Premier League season by integrating four diverse visualization techniques to explore team performance, fan engagement, and geography.

The first visualization, a multi-continuous bubble scatter plot, compares each team's total expected goals (xG_For) against their actual goals scored (Goals_For). In this plot, bubble size represents average attendance as a representation for fan support, while bubble color indicates the performance by subtracting goals scored and total expected goals, shedding light on the efficiency of team performance relative to expectations. Labels for each team (as initials) further help identify outliers and top performers.

The second visualization maps the geographic distribution of stadium locations across the United Kingdom (UK) with distances in kilometers from the top performing team of the season, Manchester City, to all other team stadiums. By overlaying the manually curated stadium coordinates on a base-map along with the computed distances from each team's stadium to Manchester City's hub, this map illustrates where top clubs are based with respect to one another and emphasizes spatial dimensions of the league. Using a legend ensures that stadium names are legible without cluttering the plot due to multiple stadiums being in close proximity to one another as observed by the points and distance segments.

The third visualization adopts a categorical approach by examining match attendance distributions through a violin plot (with an embedded box-plot) for each match outcome (Home Win, Away Win, Draw). This visualization highlights how attendance varies with match results, indicating differences in fan engagement depending on game excitement and competitive balance.

Lastly, the performance table created using the grammar of tables package offers a concise summary of team statistics, including wins, draws, losses, goals, and expected goals, with conditional formatting to emphasize key performance indicators such as match outcomes shown in color gradients, average attendance with high attendance (over 40000 people) being in bold, and goal differences where teams with positive goal differences contain an asterisk beside their name.

Together, these visualizations provide a cohesive narrative of tactical efficiency (xG vs goals), logistical context (travel map), spectator behavior (attendance), and comprehensive team statistics, all of which were derived from a single, well-curated dataset.

Data Collection and Cleaning:

```
library(sportyR)
library(worldfootballR)

library(gt)
library(maps)
library(ggrepel)
```

```
## Loading required package: ggplot2
```

```
library(ggplot2)

library(scales)
library(viridis)
```

```
## Loading required package: viridisLite

##
## Attaching package: 'viridis'

## The following object is masked from 'package:scales':
##
##     viridis_pal

## The following object is masked from 'package:maps':
##
##     unemp
```

```
library(RColorBrewer)

library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(stringr)
library(tidyverse)
```

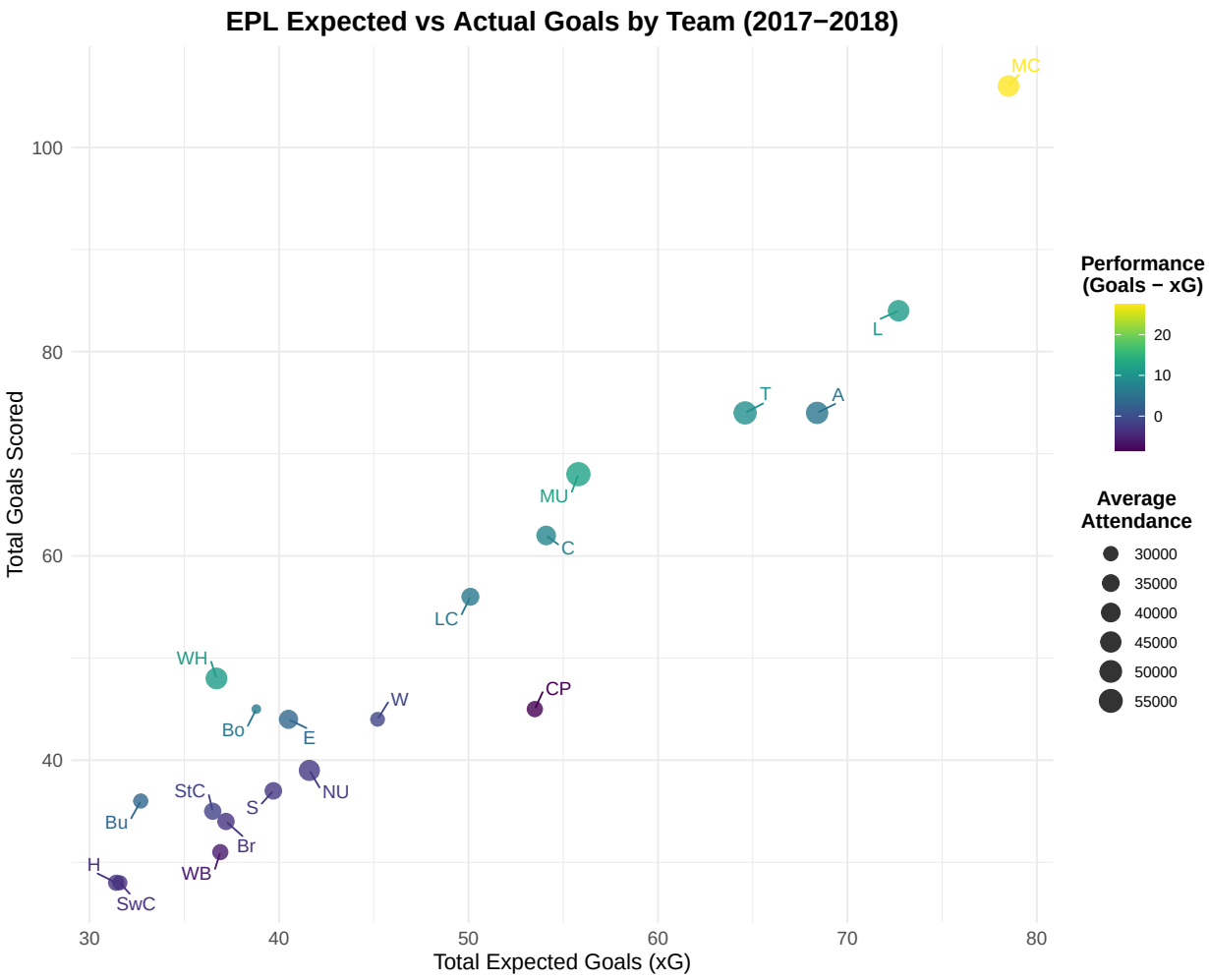
```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.5
## v lubridate 1.9.4     v tibble    3.2.1
## v purrr     1.0.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x readr::col_factor() masks scales::col_factor()
## x purrr::discard()    masks scales::discard()
## x dplyr::filter()     masks stats::filter()
## x dplyr::lag()        masks stats::lag()
## x purrr::map()        masks maps::map()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(geosphere)

## Initialize dataset and perform data cleaning as well as modifications
# Retrieve English Premier League match results for 2017-2018
season <- "2017-2018"
epl_results <- fb_match_results(country="ENG", gender="M", season_end_year=2018,
                                tier="1st")
# Add match results column
epl_results <- epl_results %>%
  mutate(
    Match_Result=case_when(
      HomeGoals > AwayGoals ~ "Home Win",
      HomeGoals < AwayGoals ~ "Away Win",
      TRUE ~ "Draw"
    )
  )

# Create data frame for home matches
home_stats <- epl_results %>%
  transmute(
    team=Home,
    goals_for=HomeGoals,
    goals_against=AwayGoals,
    xG_for=Home_xG,
    xG_against=Away_xG,
    result=case_when(
      HomeGoals > AwayGoals ~ "Win",
      HomeGoals < AwayGoals ~ "Loss",
      TRUE ~ "Draw"
    ),
    attendance=Attendance
  )
# Create data frame for away matches
away_stats <- epl_results %>%
  transmute(
    team=Away,
    goals_for=AwayGoals,
    goals_against=HomeGoals,
    xG_for=Away_xG,
    xG_against=Home_xG,
    result=case_when(
```

```
      AwayGoals > HomeGoals ~ "Win",
      AwayGoals < HomeGoals ~ "Loss",
      TRUE ~ "Draw"
    ),
    attendance=Attendance
  )
# Combine home and away statistics then summarize per team
team_summary_stats <- bind_rows(home_stats, away_stats) %>%
  mutate(Team=team) %>%
  group_by(Team) %>%
  summarize(
    Matches_Played=n(),
    Wins=sum(result == "Win"),
    Draws=sum(result == "Draw"),
    Losses=sum(result == "Loss"),
    Goals_For=sum(goals_for, na.rm=TRUE),
    Goals_Against=sum(goals_against, na.rm=TRUE),
    Goal_Difference=Goals_For-Goals_Against,
    xG_For=sum(xG_for, na.rm=TRUE),
    xG_Against=sum(xG_against, na.rm=TRUE),
    xG_Difference=xG_For-xG_Against,
    Average_Attendance=mean(attendance)
  ) %>%
  select(Team, Matches_Played, Wins, Draws, Losses, Goals_For, Goals_Against,
         Goal_Difference, xG_For, xG_Against, xG_Difference,
         Average_Attendance) %>%
  arrange(desc(Wins))
# Add season column for reference
team_summary_stats$Season <- season
```

Option 1 Visualization:

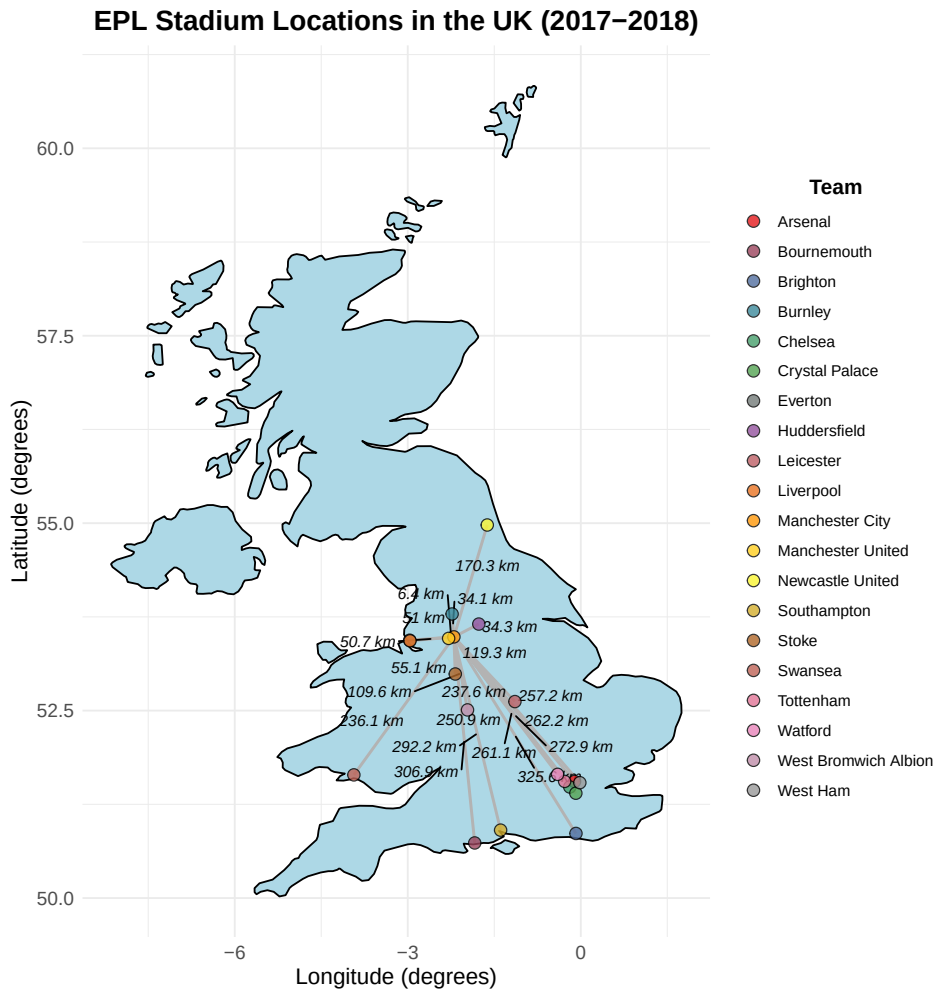
```r
## Option 1 (Visualization 1): Multi-Continuous Variables (Bubble Scatter Plot)
# Create function to generate initials from team name
get_initials <- function(name) {
  # Split team name into words
  words <- unlist(strsplit(name, " "))
  # Combine first letters of each word into single string
  paste(substr(words, 1, 1), collapse="")
}
# Add column of team initials to summary data
team_summary_stats <- team_summary_stats %>%
  mutate(Initials=sapply(Team, get_initials))
# Find teams with duplicate initials
dup_init_counts <- team_summary_stats %>%
  count(Initials) %>%
  filter(n > 1)
# Remove duplicates by adding 2nd character of team name
team_summary_stats <- team_summary_stats %>%
  mutate(
    Initials=ifelse(
      Initials %in% dup_init_counts$Initials,
      paste0(substr(Team, 1, 2), substr(Initials, 2, 2)),
      Initials
    )
  )

# Create scatter plot of total expected goals against total goals with bubble
# size representing average attendance per match and bubble color representing
# difference in actual vs expected goals
create_bubble_scatter_plot <- function(data) {
  ggplot(data=data, aes(x=xG_For, y=Goals_For)) +
  geom_point(aes(size=Average_Attendance, color=Goals_For - xG_For),
             alpha=0.8) +
  geom_text_repel(aes(label=Initials, color=Goals_For - xG_For), size=4,
                  box.padding=0.5, max.overlaps=10, force=5) +
  scale_color_viridis(option="viridis", name="Performance\n(Goals - xG)") +
  scale_size(range=c(2, 6), name="Average\nAttendance") +
  labs(title="EPL Expected vs Actual Goals by Team (2017-2018)",
       x="Total Expected Goals (xG)", y="Total Goals Scored") +
  theme_minimal() +
  theme(plot.title=element_text(size=16, face="bold", hjust=0.5),
        legend.title=element_text(size=12, face="bold", hjust=0.5),
        axis.text.x=element_text(size=11), axis.text.y=element_text(size=11),
        axis.title.x=element_text(size=13), axis.title.y=element_text(size=13))
}
create_bubble_scatter_plot(team_summary_stats)
```


EPL Expected vs Actual Goals by Team (2017−2018)
Total Goals Scored
Total Expected Goals (xG)
100
80
60
40
30
40
50
60
70
80
MC
L
T
A
MU
C
LC
WH
Bo
E
W
CP
NU
StC
S
Bu
Br
WB
H
SwC
Performance (Goals − xG)
20
10
0
Average Attendance
30000
35000
40000
45000
50000
55000

Option 2 Visualization:

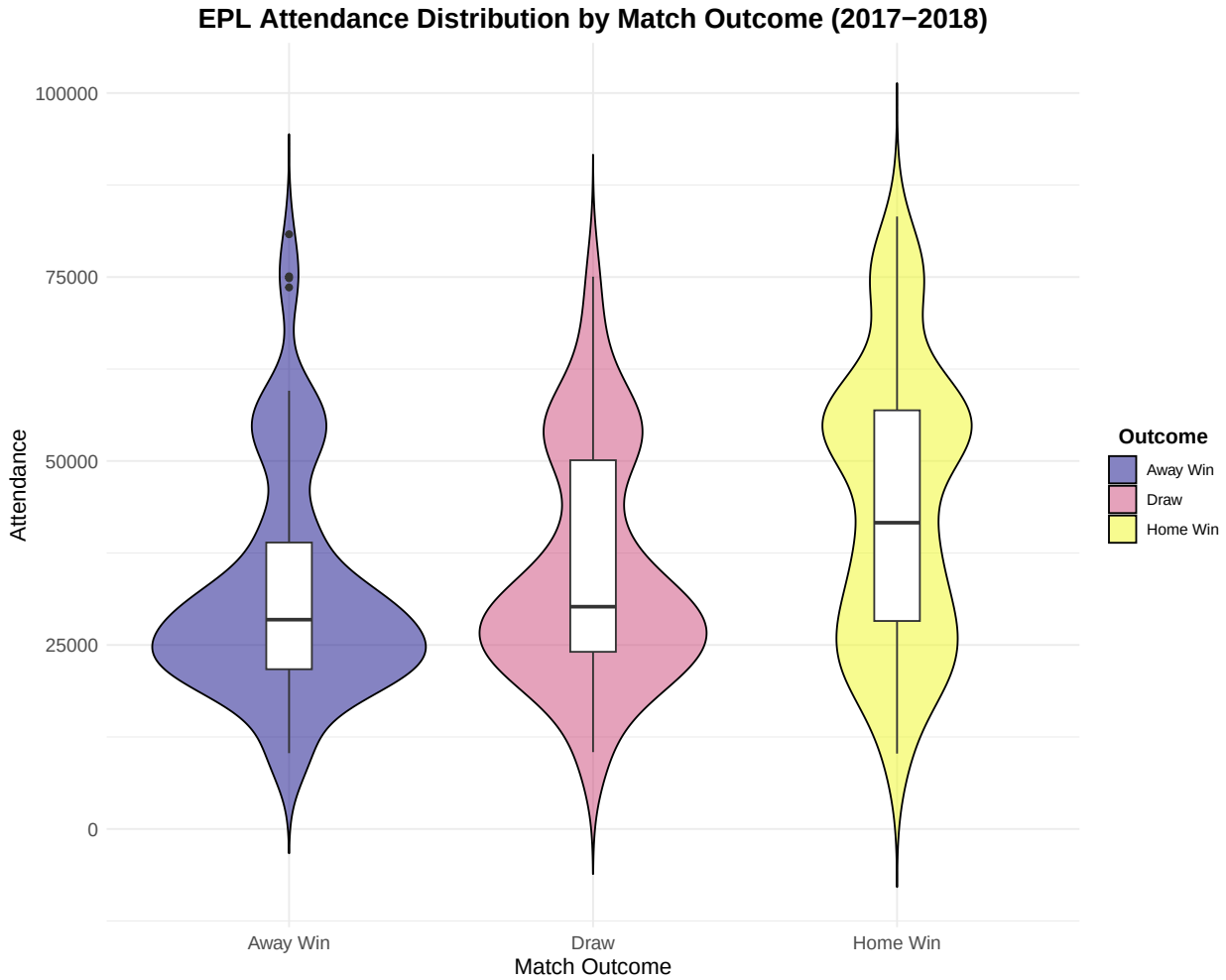
```
## Option 2 (Visualization 2): Geographical Visualization (Map of Team Stadiums)
# I would like this visualization to be worth more due to higher effort
# Create manual data frame of stadium names and coordinates per team
stadiums.dat <- data.frame(
  Team=c("Arsenal", "Bournemouth", "Brighton", "Burnley", "Chelsea",
         "Crystal Palace", "Everton", "Huddersfield", "Leicester", "Liverpool",
         "Manchester City", "Manchester United", "Newcastle United",
         "Southampton", "Stoke", "Swansea", "Tottenham", "Watford",
         "West Bromwich Albion", "West Ham"),
  Stadium=c("Emirates Stadium", "Vitality Stadium", "American Express Stadium",
            "Turf Moor", "Stamford Bridge", "Selhurst Park", "Goodison Park",
            "John Smith's Stadium", "King Power Stadium", "Anfield",
            "Etihad Stadium", "Old Trafford", "St. James' Park",
            "St. Mary's Stadium", "Bet365 Stadium", "Swansea.com Stadium",
            "Wembley Stadium", "Vicarage Road", "The Hawthorns",
            "London Stadium"),
  Lat=c(51.5551, 50.7348, 50.8616, 53.7887, 51.4817, 51.3981, 53.4388, 53.6543,
        52.6204, 53.4308, 53.4832, 53.4631, 54.9756, 50.9058, 52.9883, 51.6427,
        51.5560, 51.6499, 52.5093, 51.5387),
  Long=c(-0.1084, -1.8391, -0.0837, -2.2311, -0.1909, -0.0857, -2.9663, -1.7683,
         -1.1422, -2.9608, -2.2004, -2.2913, -1.6216, -1.3901, -2.1761, -3.9346,
         -0.2796, -0.4015, -1.9639, -0.0165)
)

# Compute distances from Manchester City (most wins in 2017-2018) to all others
hub_name <- "Manchester City"
hub_row <- filter(stadiums.dat, Team == hub_name)
others <- filter(stadiums.dat, Team != hub_name)
dist_km <- distHaversine(
  c(hub_row$Long, hub_row$Lat),
  cbind(others$Long, others$Lat)
) / 1000
# Create data frame for segments
segments.dat <- data.frame(
  x=hub_row$Long,
  y=hub_row$Lat,
  xend=others$Long,
  yend=others$Lat,
  dist=round(dist_km, 1),
  lab=paste0(dist_km %>% round(1), " km")
)
# Calculate midpoint coordinates for labeling
segments.dat <- segments.dat %>%
  mutate(mid_x=(x + xend) / 2, mid_y=(y + yend) / 2)
# Set distinct colors for each team
base_cols <- brewer.pal(9, "Set1")
team_pal <- colorRampPalette(base_cols)(length(stadiums.dat$Team))
# Get UK map data
uk.dat <- map_data("world") %>% filter(region == "UK")

# Create map plot of stadium locations with distance from Manchester City
```

```r
create_map_plot <- function(data1, data2) {
  ggplot() +
  geom_polygon(data=uk.dat, aes(x=long, y=lat, group=group), fill="lightblue",
               color="black") +
  geom_segment(data=data1, aes(x=x, y=y, xend=xend, yend=yend), color="gray70",
               linewidth=0.8) +
  geom_text_repel(data=data1, aes(x=mid_x, y=mid_y, label=lab), size=3,
                  fontface="italic", color="black", max.overlaps=20, force=10) +
  geom_point(data=data2, aes(x=Long, y=Lat, fill=Team), size=3, shape=21,
             alpha=0.8) +
  scale_fill_manual(values=team_pal, name="Team") +
  coord_fixed(1.3) +
  labs(title="EPL Stadium Locations in the UK (2017-2018)",
       x="Longitude (degrees)", y="Latitude (degrees)") +
  theme_minimal() +
  theme(plot.title=element_text(size=16, face="bold", hjust=0.5),
        legend.title=element_text(size=12, face="bold", hjust=0.5),
        axis.text.x=element_text(size=11), axis.text.y=element_text(size=11),
        axis.title.x=element_text(size=13), axis.title.y=element_text(size=13))
}
create_map_plot(segments.dat, stadiums.dat)
```



Option 3 Visualization:

```r
# Option 3 (Visualization 3): Categorical Visualization (Violin-Box Hybrid Plot)
# Create violin-box hybrid plot for attendance by match result
create_violin_box_plot <- function(data) {
  ggplot(data=data, aes(x=Match_Result, y=Attendance, fill=Match_Result)) +
  geom_violin(alpha=0.5, color="black", trim=FALSE) +
  geom_boxplot(width=0.15, fill="white") +
  scale_fill_viridis_d(option="plasma", name="Outcome") +
  labs(title="EPL Attendance Distribution by Match Outcome (2017-2018)",
       x="Match Outcome", y="Attendance") +
  theme_minimal() +
  theme(plot.title=element_text(size=16, face="bold", hjust=0.5),
        legend.title=element_text(size=12, face="bold", hjust=0.5),
        axis.text.x=element_text(size=11), axis.text.y=element_text(size=11),
        axis.title.x=element_text(size=13), axis.title.y=element_text(size=13))
}
create_violin_box_plot(epl_results)
```



Option 5 Visualization:

```
## Option 5 (Visualization 4): Formatted Table (Grammar of Tables for Teams)
# Create table of team statistics sorted by wins with large conditional
# formatting elements for match outcomes, positive goal differences, and
# average attendance of over 40000 people
create_gt_table <- function(data) {
  data %>%
    mutate(Team=if_else(Goal_Difference > 0, paste0(Team, " *"), Team)) %>%
    arrange(desc(Wins)) %>%
    gt() %>%
    tab_header(
      title=md("**EPL Team Performance Statistics (2017-2018)**"),
      subtitle=md("**Conditional Formatting on Match Outcomes, Goals,
                  and Attendance**")
    ) %>%
    tab_source_note(
      source_note="Note 1: MP = Matches Played; W = Wins; D = Draws; L = Losses."
    ) %>%
    tab_source_note(
      source_note="Note 2: xG = Expected Goals; GF = Goals For;
      GA = Goals Against; GD = Goal Difference."
    ) %>%
    tab_source_note(
      source_note="Additional Note: Attendance = Average Attendance per match."
    ) %>%
    tab_spanner(
      label="Actual",
      columns=c(Goals_For, Goals_Against, Goal_Difference)
    ) %>%
    tab_spanner(
      label="Expected",
      columns=c(xG_For, xG_Against, xG_Difference)
    ) %>%
    tab_style(
      style=cell_text(weight="bold"),
      locations=cells_body(
        columns=Average_Attendance,
        rows=(Average_Attendance >= 40000)
      )
    ) %>%
    tab_options(
      table.font.names="Roboto"
    ) %>%
    cols_hide(
      columns=c(Season, Initials)
    ) %>%
    cols_label(
      Team=html("Squad"),
      Matches_Played=html("MP"),
      Wins=html("W"),
      Draws=html("D"),
      Losses=html("L"),
```

```
      Goals_For=html("GF"),
      Goals_Against=html("GA"),
      Goal_Difference=html("GD"),
      xG_For=html("xGF"),
      xG_Against=html("xGA"),
      xG_Difference=html("xGD"),
      Average_Attendance=html("Attendance")
    ) %>%
    data_color(
      columns=Wins,
      colors=col_numeric(
        palette=c("lightgreen", "forestgreen"),
        domain=range(team_summary_stats$Wins)
      )
    ) %>%
    data_color(
      columns=Draws,
      colors=col_numeric(
        palette=c("lightblue", "blue"),
        domain=range(team_summary_stats$Draws)
      )
    ) %>%
    data_color(
      columns=Losses,
      colors=col_numeric(
        palette=c("mistyrose", "firebrick"),
        domain=range(team_summary_stats$Losses)
      )
    ) %>%
    fmt_number(
      columns=c(Goals_For, Goals_Against, Goal_Difference, xG_For, xG_Against,
                xG_Difference, Average_Attendance),
      decimals=1
    )
}
create_gt_table(team_summary_stats)
```

EPL Team Performance Statistics (2017-2018)

Conditional Formatting on Match Outcomes, Goals, and Attendance

					Actual			Expected			
Squad	MP	W	D	L	GF	GA	GD	xGF	xGA	xGD	Attendance
Manchester City *	38	32	4	2	106.0	27.0	79.0	78.5	23.6	54.9	**46,060.2**
Manchester Utd *	38	25	6	7	68.0	28.0	40.0	55.8	40.9	14.9	**56,225.4**
Tottenham *	38	23	8	7	74.0	36.0	38.0	64.6	33.8	30.8	**52,190.7**
Chelsea *	38	21	7	10	62.0	38.0	24.0	54.1	33.7	20.4	39,944.2
Liverpool *	38	21	12	5	84.0	38.0	46.0	72.7	33.9	38.8	**45,769.6**
Arsenal *	38	19	6	13	74.0	51.0	23.0	68.4	47.8	20.6	**48,851.7**
Burnley	38	14	12	12	36.0	39.0	-3.0	32.7	51.1	-18.4	29,827.6
Everton	38	13	10	15	44.0	58.0	-14.0	40.5	52.4	-11.9	38,681.4
Leicester City	38	12	11	15	56.0	60.0	-4.0	50.1	48.3	1.8	35,350.3
Newcastle Utd	38	12	8	18	39.0	47.0	-8.0	41.6	52.4	-10.8	**44,563.4**
Bournemouth	38	11	11	16	45.0	61.0	-16.0	38.8	59.1	-20.3	25,356.8
Crystal Palace	38	11	11	16	45.0	55.0	-10.0	53.5	48.4	5.1	31,805.1
Watford	38	11	8	19	44.0	64.0	-20.0	45.2	51.8	-6.6	29,218.1
West Ham	38	10	12	16	48.0	68.0	-20.0	36.7	55.4	-18.7	**46,691.5**
Brighton	38	9	13	16	34.0	54.0	-20.0	37.2	50.7	-13.5	33,941.1
Huddersfield	38	9	10	19	28.0	58.0	-30.0	31.4	48.6	-17.2	31,563.8
Swansea City	38	8	9	21	28.0	56.0	-28.0	31.6	58.5	-26.9	29,677.4
Southampton	38	7	15	16	37.0	56.0	-19.0	39.7	45.8	-6.1	34,415.5
Stoke City	38	7	12	19	35.0	68.0	-33.0	36.5	62.5	-26.0	33,752.7
West Brom	38	6	13	19	31.0	56.0	-25.0	36.9	47.8	-10.9	31,588.8

Note 1: MP = Matches Played; W = Wins; D = Draws; L = Losses.

Note 2: xG = Expected Goals; GF = Goals For; GA = Goals Against; GD = Goal Difference.

Additional Note: Attendance = Average Attendance per match.