MATH 2820L Final Project

Matthew Galvez

2024-12-10

Introduction:

We will analyze the dataset titled **Football - Soccer - UEFA EURO, 1960 - 2024**, which is publicly available on Kaggle. The dataset can be accessed **here**. This dataset includes match results, team statistics, player performance, and country rankings, from 1960 - 2024, providing a comprehensive resource for analyzing historical and recent trends in the UEFA EURO championships. We will answer questions on how countries performed over the years, how individual players performed and impacted their respective countries, how the game has changed from 1960 to 2024, as well as using the results to predict the likely UEFA 2024 champions.

```
# Download the necessary libraries to analyze data.
library(ggplot2)
library(tidyverse)
library(jsonlite)
library(caret)
library(randomForest)
```

Team Performance

```
# Load summary data
euro_summary_data <- read.csv("~/vanderbilt/math_stats/archive/euro_summary.csv")

# Inspect the summary data
print(tibble(euro_summary_data$year, euro_summary_data$winner, euro_summary_data$result))
```

```
## # A tibble: 17 x 3
##    `euro_summary_data$year` `euro_summary_data$winner` euro_summary_data$resul~1
##                       <int> <chr>                      <chr>
##  1                     1960 USSR                       1 - 1
##  2                     1964 Spain                      2 - 2
##  3                     1968 Italy                      2 - 2
##  4                     1972 West Germany               3 - 3
##  5                     1976 Czechoslovakia             2 - 2 (5 - 3)
##  6                     1980 West Germany               1 - 1
##  7                     1984 France                     2 - 2
##  8                     1988 Netherlands                0 - 0
##  9                     1992 Denmark                    2 - 2
## 10                     1996 Germany                    1 - 1
## 11                     2000 France                     1 - 1
## 12                     2004 Greece                     0 - 0
## 13                     2008 Spain                      0 - 0
## 14                     2012 Spain                      4 - 4
## 15                     2016 Portugal                   0 - 0
## 16                     2020 Italy                      1 - 1 (3 - 2)
```

```
## 17                    2024 Spain                       2 - 2
## # i abbreviated name: 1: `euro_summary_data$result`
```

```
# Load coaches and lineups data
coaches_data <- read.csv("~/vanderbilt/math_stats/archive/euro_coaches.csv")

# Inspect the coaches data
print(tail(tibble(coaches_data$year, coaches_data$country, coaches_data$name)))
```

```
## # A tibble: 6 x 3
##   `coaches_data$year` `coaches_data$country` `coaches_data$name`
##                 <int> <chr>                  <chr>
## 1                2024 Croatia                Zlatko Dalić
## 2                2024 Spain                  Luis de la Fuente
## 3                2024 Switzerland            Murat Yakin
## 4                2024 Italy                  Marco Rossi
## 5                2024 Scotland               Steve Clarke
## 6                2024 Germany                Julian Nagelsmann
```

```
# Load all match files
match_files <- list.files("~/vanderbilt/math_stats/archive/matches/matches/euro",
full.names = TRUE, pattern = "*.csv")

# Function to read and process each CSV file
read_and_process_csv <- function(file) {
  data <- read.csv(file)

  # Ensure matchday_name is a character
  if ("matchday_name" %in% names(data)) {
    data$matchday_name <- as.character(data$matchday_name)
  }

  return(data)
}

# Map the function to parse only matchday data
matches_data <- map_df(match_files, read_and_process_csv)

# Inspect the matches data
print(head(tibble(matches_data$year, matches_data$home_team, matches_data$away_team, matches_data$home_s
```

```
## # A tibble: 6 x 5
##   `matches_data$year` `matches_data$home_team` `matches_data$away_team`
##                 <int> <chr>                    <chr>
## 1                1960 USSR                     Yugoslavia
## 2                1960 Czechoslovakia           France
## 3                1960 Czechoslovakia           USSR
## 4                1960 France                   Yugoslavia
## 5                1964 Spain                    USSR
## 6                1964 Hungary                  Denmark
## # i 2 more variables: `matches_data$home_score` <dbl>,
## #   `matches_data$away_score` <dbl>
```

```
# Standardize and clean match data
matches_data <- matches_data %>%
  mutate(
```

```
    date = as.Date(date, format = "%Y-%m-%d"),
    year = year(date),
    home_team = str_to_title(home_team),
    away_team = str_to_title(away_team)
  ) %>% arrange(desc(date))


# Make sure data is up to date
head(
  select(matches_data, home_team, away_team, home_score, away_score, date)
)
```

```
##     home_team   away_team home_score away_score       date
## 1       Spain     England          2          1 2024-07-14
## 2 Netherlands     England          1          2 2024-07-10
## 3       Spain      France          2          1 2024-07-09
## 4 Netherlands     Türkiye          2          1 2024-07-06
## 5     England Switzerland          1          1 2024-07-06
## 6    Portugal      France          0          0 2024-07-05
```

```
# Check old data
tail(
  select(matches_data, home_team, away_team, home_score, away_score, date)
)
```

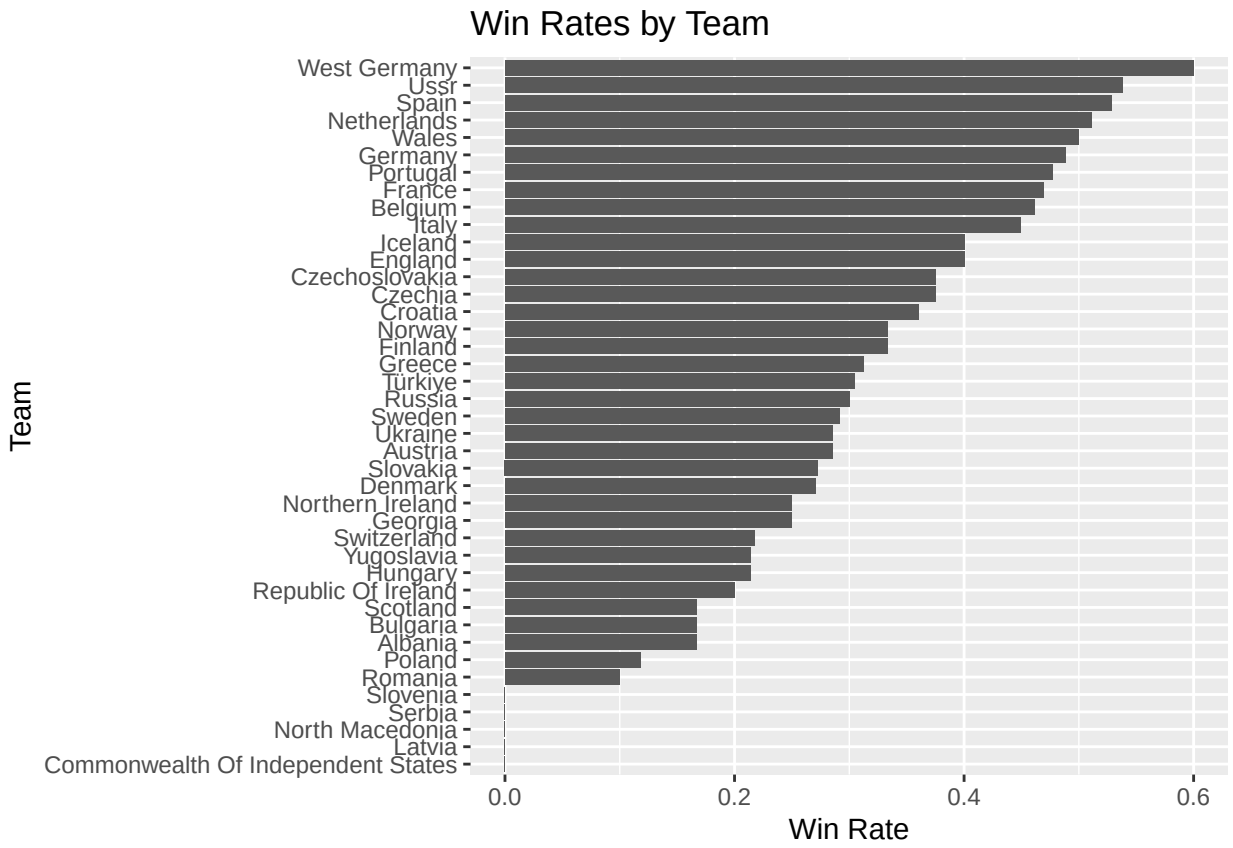
```
##          home_team  away_team home_score away_score       date
## 383        Denmark       Ussr          0          3 1964-06-17
## 384          Spain    Hungary          1          1 1964-06-17
## 385           Ussr Yugoslavia          1          1 1960-07-10
## 386 Czechoslovakia     France          2          0 1960-07-09
## 387 Czechoslovakia       Ussr          0          3 1960-07-06
## 388         France Yugoslavia          4          5 1960-07-06
```

```
# In order to calculate team win rates for each country, we must get data from
# both home and away since the data is set up so that there is no overcounting.

# Get team performances for each team (from home and away)
team_performance_home <- matches_data %>%
  mutate(
    home_penalty = coalesce(home_penalty, 0),
    away_penalty = coalesce(away_penalty, 0),
    is_penalty_shootout = home_penalty > 0 | away_penalty > 0
  ) %>%
  group_by(home_team_code, home_team) %>%
  summarize(
    matches_played = n(),
    # Calculate wins / draws / losses based on score_totals
    # i.e. total number of goals scored, including goals from penalty shootouts
    wins = sum(home_score_total > away_score_total),
    draws = sum(home_score_total == away_score_total),
    losses = sum(home_score_total < away_score_total),
    win_rate = wins / matches_played
  ) %>%
  rename(team = home_team, country_code = home_team_code)
```

```r
# Calculate win rates for away performance
team_performance_away <- matches_data %>%
  mutate(
    home_penalty = coalesce(home_penalty, 0),
    away_penalty = coalesce(away_penalty, 0),
    is_penalty_shootout = (home_penalty > 0 | away_penalty > 0)
  ) %>%
  group_by(away_team_code, away_team) %>%
  summarize(
    matches_played = n(),
    wins = sum(away_score_total > home_score_total),
    draws = sum(away_score_total == home_score_total),
    losses = sum(away_score_total < home_score_total),
    win_rate = wins / matches_played
  ) %>%
  rename(team = away_team, country_code = away_team_code)

# Combine home and away performance data
combined_performance <- bind_rows(team_performance_home, team_performance_away)

# Aggregate the data by team and calculate totals
total_team_performance <- combined_performance %>%
  group_by(country_code, team) %>%
  summarize(
    total_matches = sum(matches_played),
    total_wins = sum(wins),
    total_draws = sum(draws),
    total_losses = sum(losses),
    overall_win_rate = total_wins / total_matches
  )

# View the result
head(total_team_performance)
```

```
## # A tibble: 6 x 7
## # Groups:   country_code [6]
##   country_code team            total_matches total_wins total_draws total_losses
##   <chr>        <chr>                   <int>      <int>       <int>        <int>
## 1 ALB          Albania                     6          1           1            4
## 2 AUT          Austria                    14          4           2            8
## 3 BEL          Belgium                    26         12           3           11
## 4 BUL          Bulgaria                    6          1           1            4
## 5 CIS          Commonwealth O~             3          0           2            1
## 6 CRO          Croatia                    25          9           8            8
## # i 1 more variable: overall_win_rate <dbl>
```

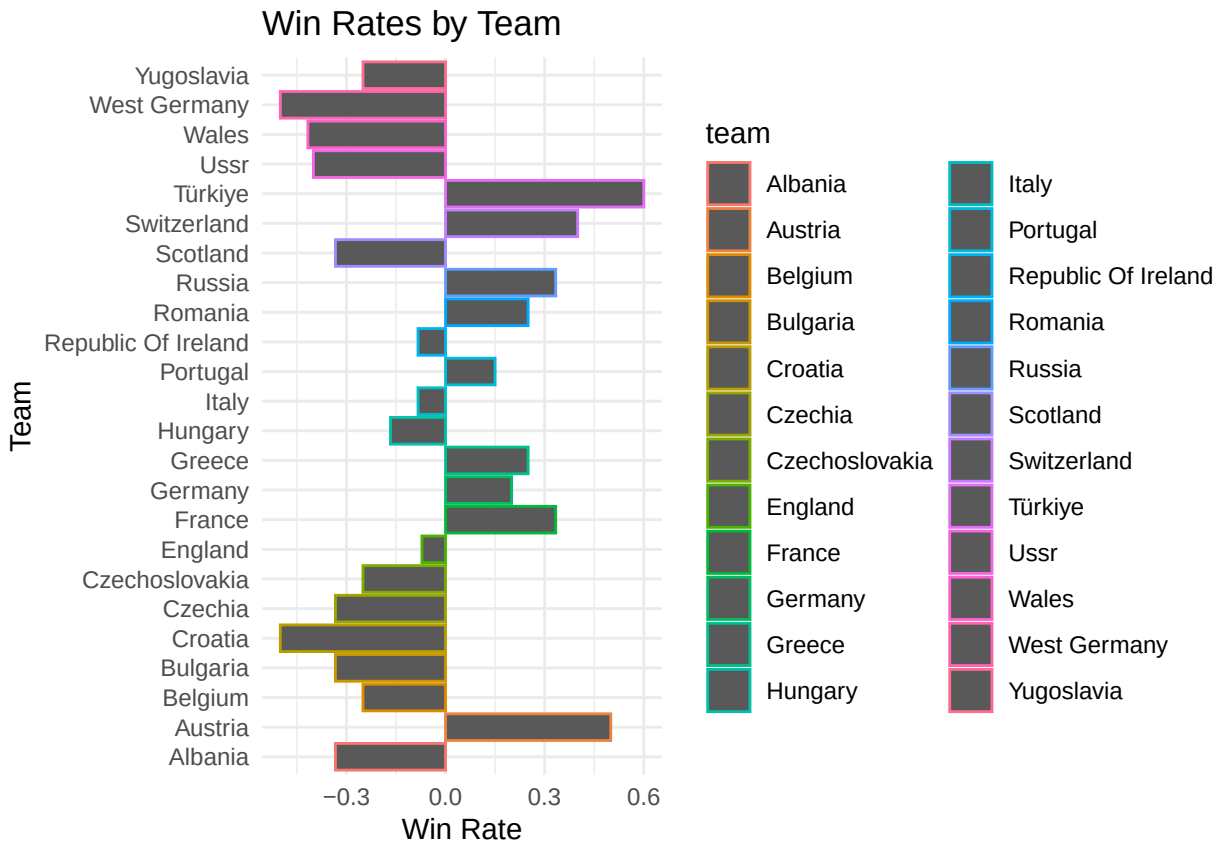
```r
# Visualize cumulative win rates from 1960 - 2024 (including countries that no longer exist)
ggplot(total_team_performance, aes(x = reorder(team, overall_win_rate), y = overall_win_rate)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Win Rates by Team", x = "Team", y = "Win Rate")
```



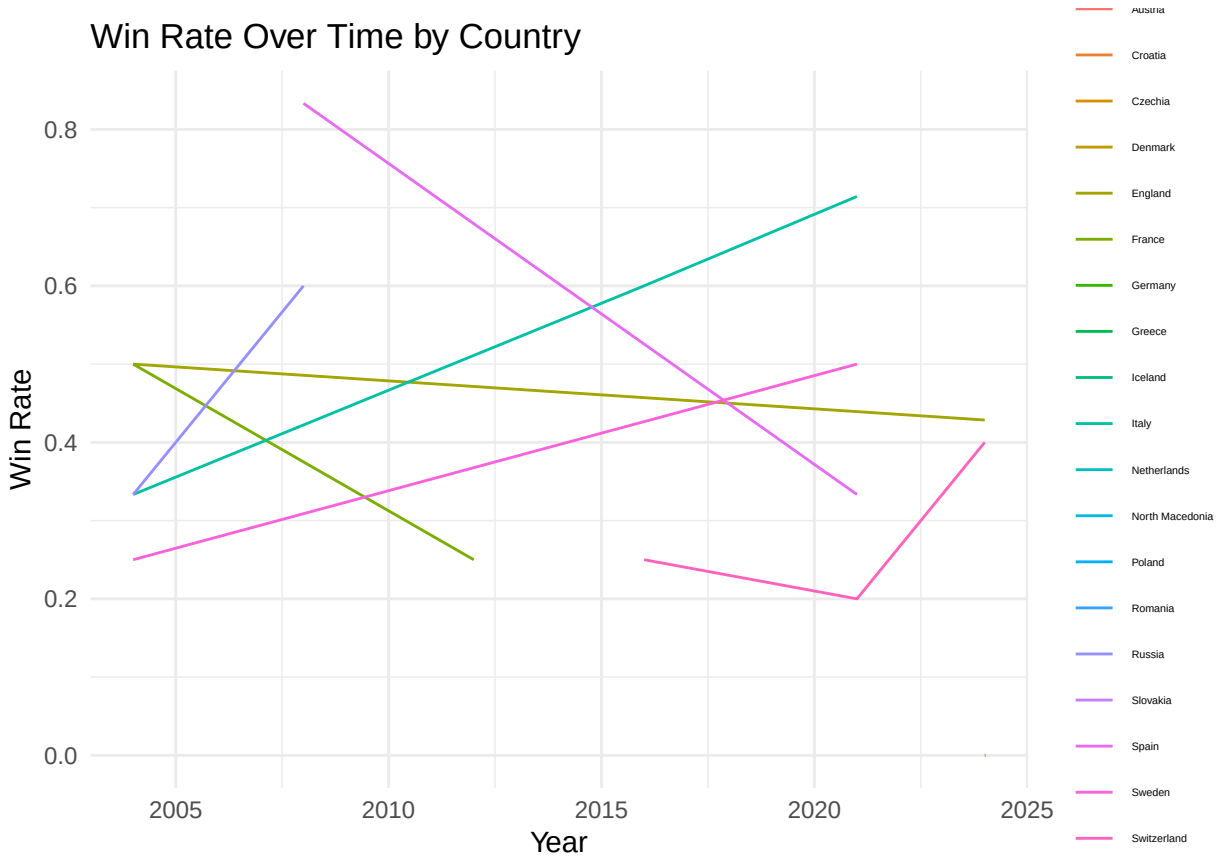
Country Comparisons

```r
# Join match and summary data
master_data <- matches_data %>%
  left_join(euro_summary_data, by = "year")


# In order to calculate yearly win rates for each country, we must get data from
# both home and away since the data is set up so that there is no overcounting.

# Get yearly home performance
yearly_home_performance <- master_data %>%
  mutate(
    home_penalty = coalesce(home_penalty, 0),
    away_penalty = coalesce(away_penalty, 0),
    is_penalty_shootout = (home_penalty > 0 | away_penalty > 0)
  ) %>%
  group_by(year, home_team) %>%
  summarize(
    matches_played = n(),
    wins = sum(home_score_total > away_score_total),
    draws = sum(home_score_total == away_score_total),
    losses = sum(home_score_total < away_score_total),
    win_rate = wins / matches_played
  ) %>%
  rename(team = home_team)
```

```r
# Get yearly away performance
yearly_away_performance <- master_data %>%
  mutate(
    home_penalty = coalesce(home_penalty, 0),
    away_penalty = coalesce(away_penalty, 0),
    is_penalty_shootout = (home_penalty > 0 | away_penalty > 0)
  ) %>%
  group_by(year, away_team) %>%
  summarize(
    matches_played = n(),
    wins = sum(away_score_total > home_score_total),
    draws = sum(away_score_total == home_score_total),
    losses = sum(away_score_total < home_score_total),
    win_rate = wins / matches_played
  ) %>%
  rename(team = away_team)

# Combine home and away performance data
combined_yearly_performance <- bind_rows(yearly_home_performance, yearly_away_performance)

# Aggregate the data by team and calculate totals
total_yearly_performance <- combined_yearly_performance %>%
  group_by(year, team) %>%
  summarize(
    total_matches = sum(matches_played),
    total_wins = sum(wins),
    total_draws = sum(draws),
    total_losses = sum(losses),
    overall_win_rate = total_wins / total_matches
  )


# Calculate win rate change over time
win_rate_change <- total_yearly_performance %>%
  group_by(team) %>%
  summarize(
    initial_win_rate = first(overall_win_rate),
    final_win_rate = last(overall_win_rate),
    change = final_win_rate - initial_win_rate
  ) %>%
  arrange(desc(change))

# Filter out the countries that have barely changed their win rate
win_rate_change <- win_rate_change %>% filter(change > 0.05 | change < -0.05)

# Visualize cumulative win rates from 1960 - 2024 (including countries that no longer exist)
ggplot(win_rate_change, aes(x = team, y = change, color = team)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Win Rates by Team", x = "Team", y = "Win Rate") +
  theme_minimal()
```




```r
# Starting from year 2000, and only sampling 5 so that graph is not too sporadic.
# May include lines intersect or that stop and continue elsewhere, since some countries fail to qualify
# for the Euros, and there is no data to keep track of.
yearly_performance_filtered <- sample_n(total_yearly_performance, 5, replace = TRUE) %>%
  filter(year > 2000)

# Plot line chart to see how countries' win rates changed over time.
ggplot(yearly_performance_filtered, aes(x = year, y = overall_win_rate, color = team)) +
  geom_line() +
  labs(title = "Win Rate Over Time by Country", x = "Year", y = "Win Rate") +
  theme_minimal() +
  theme(
    legend.text = element_text(size = 4),
    legend.title = element_text(size = 8)
  )
```



Player Impact

```r
# Function to extract player stats from JSON columns
extract_player_stats_from_json <- function(master_data) {

  parse_goals <- function(goals_json) {
  if (isTRUE(is.null(goals_json)) || isTRUE(is.na(goals_json)) || goals_json == "") {
    return(data.frame(country_code = character(), international_name = character(), goals_scored = inte
  }

  # Replace single quotes with double quotes, None with null, and nan with null
  goals_json <- gsub("'", "\"", goals_json)
  goals_json <- gsub("None", "null", goals_json)
  goals_json <- gsub("nan", "null", goals_json)

  # Parse the corrected JSON
  goals_data <- tryCatch(fromJSON(goals_json), error = function(e) return(data.frame(country_code = cha

    if (is.null(goals_data) || nrow(goals_data) == 0) {
      return(data.frame(country_code = character(), international_name = character(), goals_scored = in
    }

  goals_data %>%
    # filter(goal_type == "SCORED") %>%
    group_by(international_name, country_code) %>%
    summarize(goals_scored = n(),.groups = "drop")
```

```r
}

parse_home_lineups <- function(lineups_json) {
  if (isTRUE(is.null(lineups_json)) || isTRUE(is.na(lineups_json)) || lineups_json == "") {
    return(data.frame(country_code = character(), name = character()))
  }

  # Replace single quotes with double quotes, None with null, and nan with null
  lineups_json <- gsub("'", "\"", lineups_json)
  lineups_json <- gsub("None", "null", lineups_json)
  lineups_json <- gsub("nan", "null", lineups_json)

  # Parse the corrected JSON
    lineup_data <- tryCatch(fromJSON(lineups_json), error = function(e) return(data.frame(country_code =
    if (is.null(lineup_data) || nrow(lineup_data) == 0) {
      return(data.frame(country_code = character(), international_name = character()))
    }

  lineup_data %>%
    rename(international_name = name) %>%
    group_by(international_name, country_code) %>%
    summarize(total_appearances = n(), .groups = "drop")
}

  # Extract and combine player stats from goals and home lineups
  player_stats <- master_data %>%
    rowwise() %>%
    do({
      goals_stats <- parse_goals(.$goals.x)
      home_lineup_stats <- parse_home_lineups(.$home_lineups)
      away_lineup_stats <- parse_home_lineups(.$away_lineups)
      lineup_stats <- bind_rows(home_lineup_stats, away_lineup_stats)
      full_join(goals_stats, lineup_stats, by = c("international_name", "country_code"))
    }) %>%
    ungroup() %>%
    group_by(international_name, country_code) %>%
    summarize(
      total_goals = sum(goals_scored, na.rm = TRUE),
      total_appearances = sum(total_appearances, na.rm = TRUE)
    ) %>%
    arrange(desc(total_goals))

  return(player_stats)
}

# Extract the statistics for each player
player_stats <- extract_player_stats_from_json(master_data)

# For some of the data in that was extracted, showed missing/incomplete information for lineups and goa
# Thus, the line below should return true.
any(is.na(player_stats))
```

```
## [1] TRUE
```

```r
# One example is that the country_code and total_appearances for Alan Shearer seem to be missing.
head(player_stats)
```

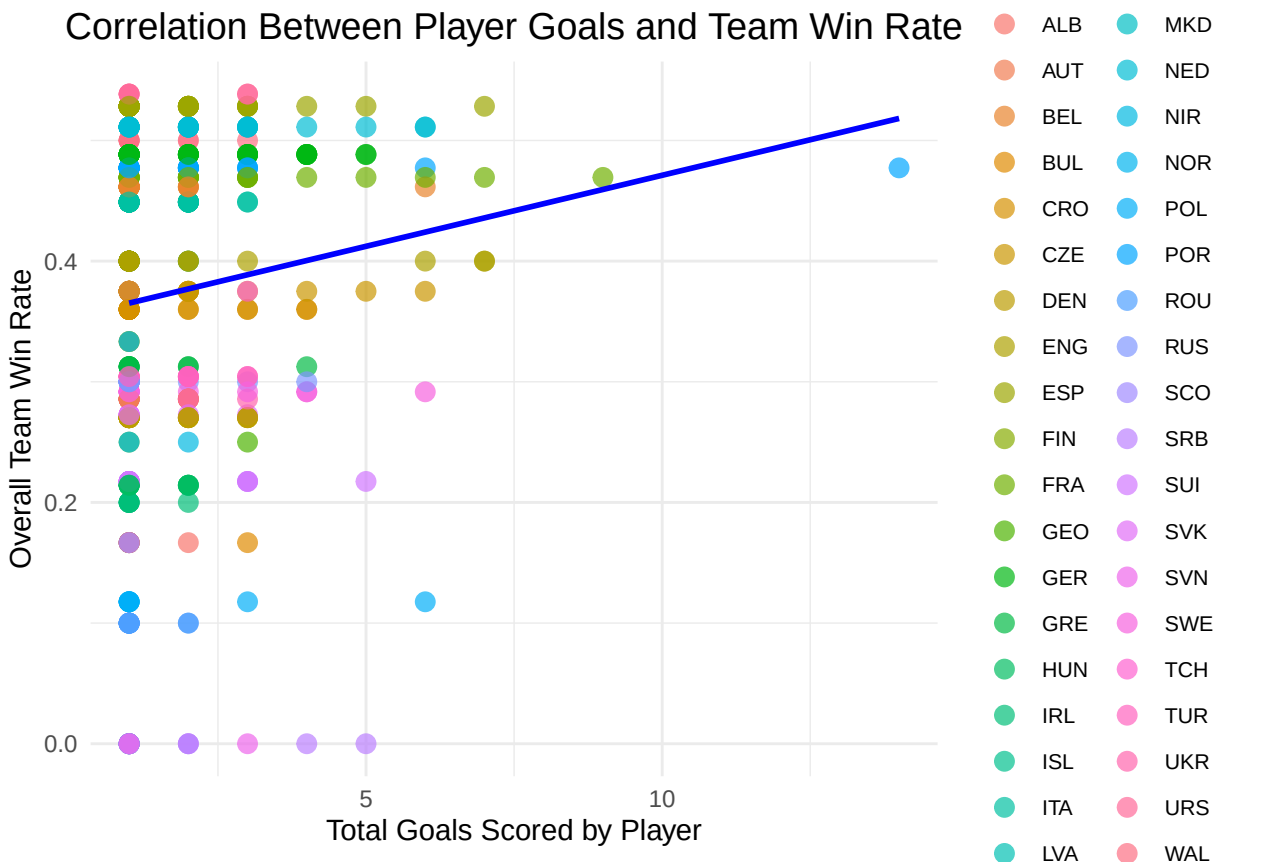
```
## # A tibble: 6 x 4
## # Groups:   international_name [6]
##   international_name country_code total_goals total_appearances
##   <chr>              <chr>              <int>             <int>
## 1 Cristiano Ronaldo  "POR"                 14                31
## 2 Michel Platini     "FRA"                  9                 5
## 3 Alan Shearer       ""                     7                 0
## 4 Antoine Griezmann  "FRA"                  7                 1
## 5 Harry Kane         "ENG"                  7                18
## 6 Álvaro Morata      "ESP"                  7                17
```

```r
# However, the issue is that the info for total_goals is counted into 1 row,
# and the info for total_appearances is in another row.
# To overcome, this issue for Alan Shearer specifically, although there exists other cases like this,
# I decided to manually fix the values for Alan Shearer since he's one of the top scorers.

# Get all instances of Alan Shearer from player_stats
alan <- player_stats %>%
  group_by(international_name) %>%
  summarize(
    total_goals = sum(total_goals, na.rm = TRUE),
    total_appearances = sum(total_appearances, na.rm = TRUE),
    country_code = first(country_code),
    .groups = "drop"
  ) %>%
  filter(international_name == "Alan Shearer" & !is.na(country_code)) %>%
  arrange(desc(total_goals))

# Manually change the country_code to ENG
alan <- alan %>% mutate(country_code = "ENG")

# Remove all instances of Alan Shearer from player_stats
player_stats <- player_stats[ !(player_stats$international_name %in% c("Alan Shearer")), ]
# Add the Alan Shearer data back into player_stats accordingly
player_stats <- bind_rows(player_stats, alan) %>% arrange(desc(total_goals))
head(player_stats)
```

```
## # A tibble: 6 x 4
## # Groups:   international_name [6]
##   international_name country_code total_goals total_appearances
##   <chr>              <chr>              <int>             <int>
## 1 Cristiano Ronaldo  POR                   14                31
## 2 Michel Platini     FRA                    9                 5
## 3 Antoine Griezmann  FRA                    7                 1
## 4 Harry Kane         ENG                    7                18
## 5 Álvaro Morata      ESP                    7                17
## 6 Alan Shearer       ENG                    7                11
```

```r
# Although this is 1 example, taking the time to set the data for other players would be too much for t
# as the player_stats data is over 3,210 rows long, so we'll keep moving on.
```

```r
# Calculate team success metrics and correlate them with player stats
# Join player_stats with team performance data
player_team_data <- player_stats %>%
  left_join(total_team_performance %>% select(country_code, overall_win_rate), by = "country_code") %>%
  filter(!is.na(overall_win_rate) & total_goals > 0) # Exclude rows with missing win rates

# Create a scatter plot with regression line
ggplot(player_team_data, aes(x = total_goals, y = overall_win_rate)) +
  # Points for player's goals
  geom_point(aes(color = country_code), size = 3, alpha = 0.7) +
  # Regression line for testing player goals with country win rates
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(
    title = "Correlation Between Player Goals and Team Win Rate",
    x = "Total Goals Scored by Player",
    y = "Overall Team Win Rate"
  ) +
  theme_minimal() +
  theme(
    legend.title = element_text(size = 10),
    legend.text = element_text(size = 8),
    plot.title = element_text(hjust = 0.5, size = 14)
  )
```

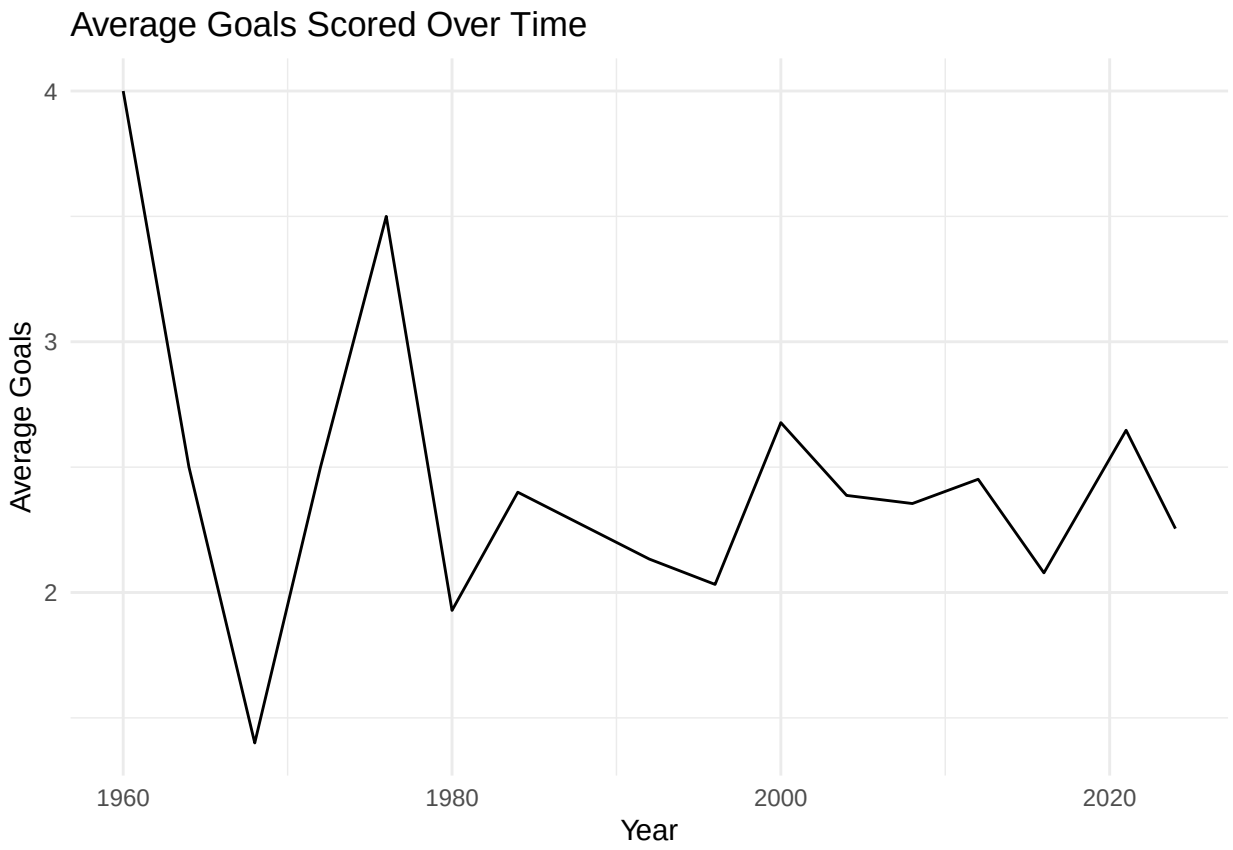



```r
# So as individual players score more goals, then the country has higher rates of success,
# as shown below, with Cristiano Ronaldo having 14 goals, representing Portugal (POR) and
# increasing the odds of Portugal winning. Likewise, players that score fewer goals correlates
```

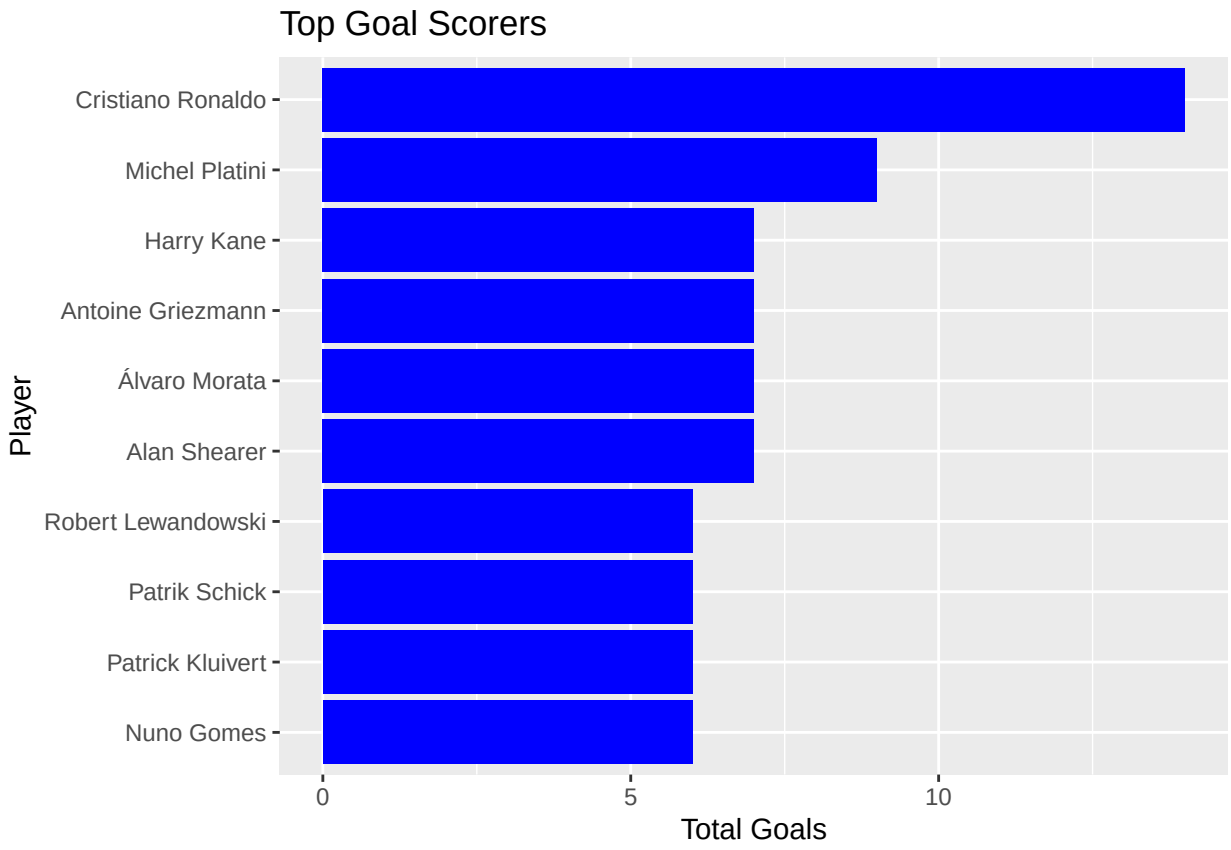
```
# with their countries have a lower win rate.
```

Trend Analysis

```
# Trend analysis for goals scored
goal_trends <- master_data %>%
  group_by(year) %>%
  summarize(
    average_goals = mean(home_score + away_score, na.rm = TRUE)
  )

# Plot for average goals scored from 1960 - 2024
ggplot(goal_trends, aes(x = year, y = average_goals)) +
  geom_line() +
  labs(title = "Average Goals Scored Over Time", x = "Year", y = "Average Goals") +
  theme_minimal()
```



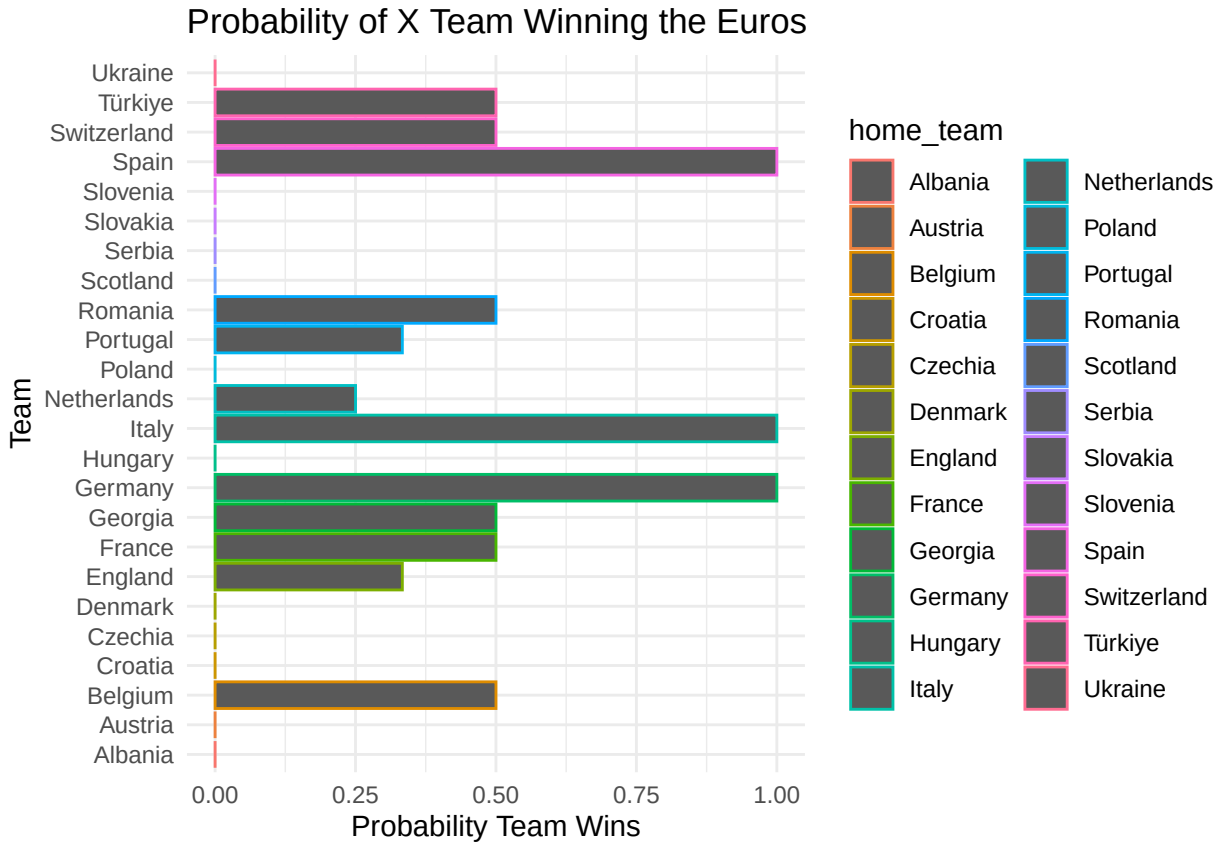

```
# Average number of goals scored have gone down likely due to the evolution of game tactics,
# defensive attitudes, and better skilled players.

# Trend analysis for player statistics
# Plot top ten goal scorers
top_scorers = head(player_stats, 10)
ggplot(top_scorers, aes(x = reorder(international_name, total_goals), y = total_goals)) +
  geom_bar(stat = "identity", fill = "blue") +
  coord_flip() +
  labs(title = "Top Goal Scorers", x = "Player", y = "Total Goals")
```



Country Predictions for 2024

```
# Will only be analyzing past results, as including and testing other factors
# would be a lot more time consuming and relevant to the scope of the project.
model_data <- master_data %>%
  select(home_team, away_team, home_score_total, away_score_total, result) %>%
  mutate(result = as.factor(home_score_total - away_score_total))


# Split into training and testing sets, seed set for repeatable tests
set.seed(123)
train_index <- createDataPartition(model_data$result, p = 0.8, list = FALSE)
train_data <- model_data[train_index, ]

# Train random forest model, used for linear regression on the results (numeric values)
# Set the variable "result" to be the predictor, use data from train_data_clean, set an appropriate tre
# based on trial and error, starting from 10 to 1000, using real-life expectations)
rf_model <- randomForest(result ~ ., data = train_data, ntree = 500, importance = TRUE)

# Get predictions from rf_model
euro_2024_predictions <- master_data %>%
  filter(year == 2024) %>%
  mutate(predicted_result = predict(rf_model, .))

# Summarize predictions
euro_2024_summary <- euro_2024_predictions %>%
```

```r
  group_by(home_team) %>%
  summarize(probability_of_winning = mean(home_score_total > away_score_total))

# Plot the probability of each country in winning the 2024 UEFA Euros, although much more precise predi
# are based on a lot more factors, such as player performance in club and country, coaching tactics, an
# chemistry in qualifying matches, than simply just previous results.
ggplot(euro_2024_summary, aes(x = home_team, y = probability_of_winning, color = home_team)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Probability of X Team Winning the Euros", x = "Team", y = "Probability Team Wins") +
  theme_minimal()
```



Conclusions

The dataset **Football - Soccer - UEFA EURO, 1960 - 2024** from Kaggle that includes match results, team statistics, player performance, and country rankings, from 1960 - 2024, brought a comprehensive overview of the UEFA Euro Championships. The data was used to cover how well each country performed in from 1960 - 2024, showing which countries had successes in the championships, showcased by their respective win rates. We also compared how countries performed throughout the decades, starting at 1960 until 2024, although some of the data is sporadic, particularly from 1960s to 1990s, since some of the data from countries such as USSR, Czechoslavakia, and Yugoslavia were discontinued, and the data from new countries had to start over again. We parsed through player statistics to determine how individual players impacted their respective countries, showing players that scored a lot of goals, such as Cristiano Ronaldo, were correlated with a higher win rate for their countries. The analysis of trends such as average goals from 1960 to 2024 shows that the average number of goals scored in a game has decreased significantly from an average of 4.0 in 1960 to ~2.25 in 2024. We concluded the dataset analysis with predicted which countries were most likely to succeed and in the UEFA 2024 tournament. The prediction was done with a random forest model from the

caret library, which is used for regression, since we are computing probabilities. 80% of the model data was used for training data to be tested against the rest of the 20% of the model data (keeping track of home/away teams and scores). We used the established preProcess from the caret library to get the prediction parameters (correlating variables), coming from the results column. This was used to train the random forest model to predict which countries were likely to succeed. Not surprisingly, countries that have succeeded in the past, such as Italy, Germany, Spain, etc. . . , had a higher probability of winning. Although some of the the more intricate details, such as assists and country rankings, were not included in the dataset, overall, the data provided a complete recap of the UEFA Euro Championship and allowed us to bring about an extensive data analysis.