English Premier League (1992/93 - 2021/22) Spending Analysis and Predictions for the next 20 years

Namit Shrivastava

2024-04-13

Data Extraction

```
# Loading necessary libraries
library(rvest)
library(httr)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.0.4
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter()         masks stats::filter()
x readr::guess_encoding() masks rvest::guess_encoding()
x dplyr::lag()            masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor
```

```
library(jsonlite)
```

```
Attaching package: 'jsonlite'
```

```
The following object is masked from 'package:purrr':

    flatten
```

```
library(tidyjson)
```

```
Attaching package: 'tidyjson'

The following object is masked from 'package:jsonlite':

    read_json

The following object is masked from 'package:stats':

    filter
```

```
library(dplyr)

# Function to scrape transfer data for a given season
scrape_transfer_data <- function(season) {

  url <- sprintf("https://www.transfermarkt.co.uk/premier-league/transfers/wettbewerb/GB1/sa
  season)
  cat("Scraping transfer data for season:", season, "\n")

  # Reading the page content
  page <- tryCatch({
    read_html(url)
  }, error = function(e) {
    cat("Error reading URL:", url, "\n")
    return(NULL)
  })

  if (is.null(page)) return(list())

  # Locating the table node (Transfermarkt usually labels it with class "items")
  table_node <- html_node(page, "table.items")
  if (is.na(table_node)) {
    cat("Transfer table not found for season:", season, "\n")
    return(list())
```

```
  }

  # Parsing the table into a data frame (fill = TRUE handles missing cells)
  transfer_table <- html_table(table_node, fill = TRUE)

  # Formatting season as "1992/93", "1993/94", etc.
  formatted_season <- sprintf("%d/%s", season, substr(as.character(season + 1), 3, 4))

  # Column 2: Club, Column 3: Income, Column 4: Expense, Column 5: Balance.
  if (ncol(transfer_table) < 5) {
    cat("Not enough columns in transfer table for season:", season, "\n")
    return(list())
  }

  # Renaming and selecting the columns of interest
  transfer_data <- transfer_table %>%
    rename(
      Club = 2,
      Income = 3,
      Expense = 4,
      Balance = 5
    ) %>%
    mutate(Season = formatted_season)

  # Converting each row to a list
  records <- split(transfer_data, seq(nrow(transfer_data)))
  records <- lapply(records, as.list)

  return(records)
}
```

```
# Function to scrape the EPL league table data for a given season
scrape_epl_table <- function(season) {
  # Constructing the URL for the league table
  url <- sprintf("https://www.transfermarkt.co.uk/premier-league/tabelle/wettbewerb/GB1/sais
  season)
  cat("Scraping league table data for season:", season, "\n")

  page <- tryCatch({
    read_html(url)
  }, error = function(e) {
    cat("Error reading URL:", url, "\n")
```

```
    return(NULL)
  })

  if (is.null(page)) return(list())

  table_node <- html_node(page, "table.items")
  if (is.na(table_node)) {
    cat("League table not found for season:", season, "\n")
    return(list())
  }

  league_table <- html_table(table_node, fill = TRUE)
  formatted_season <- sprintf("%d/%s", season, substr(as.character(season + 1), 3, 4))

  if (ncol(league_table) < 9) {
    cat("Not enough columns in league table for season:", season, "\n")
    return(list())
  }

  league_data <- league_table %>%
    rename(
      Position = 1,
      Team = 2,
      Played = 3,
      Wins = 4,
      Draws = 5,
      Losses = 6,
      Points = 9
    ) %>%
    mutate(Season = formatted_season)

  records <- split(league_data, seq(nrow(league_data)))
  records <- lapply(records, as.list)

  return(records)
}

# Main code to loop over seasons and assemble the data
transfer_data_all <- list()
epl_table_all <- list()
```

So firstly, I created two specialized functions to collect historical Premier League data from

Transfermarkt. The first function, scrape_transfer_data(), extracts transfer information for each season by constructing the appropriate URL with the season parameter. I made sure to include error handling to gracefully manage any connection issues like the function attempts to read the webpage's HTML and looks specifically for tables with the "items" class that contain the transfer data. When it finds the right table, it parses the content and formats it properly, extracting club names, income, expenses, and balance figures.

My second function, scrape_epl_table(), works similarly but targets the final league standings. It grabs position information, team names, match statistics (played, wins, draws, losses), and point totals. For both functions, I format the season nicely (like "1992/93") and transform each row of data into a list format that will work well for JSON export. I also included thorough validation checks to ensure the tables contain the expected number of columns before processing them. This approach lets me build a comprehensive dataset spanning nearly three decades of Premier League history in a structured format that's ready for analysis.

```
# # Looping through each season from 1992/93 (season = 1992) to 2021/22 (season = 2021)
# for (season in 1992:2021) {
#   cat("Processing season:", season, "\n")

#   # Scrape transfer data for the season
#   transfers <- scrape_transfer_data(season)
#   if (length(transfers) > 0) {
#     transfer_data_all <- c(transfer_data_all, transfers)
#   }
#   Sys.sleep(2)

#   # Scrape league table data
#   league_table <- scrape_epl_table(season)
#   if (length(league_table) > 0) {
#     epl_table_all <- c(epl_table_all, league_table)
#   }
#   Sys.sleep(2)
# }
```

Commented for now since the process extraction is redundant

After creating the specialized scraping functions, I used this loop to process 30 seasons of Premier League data. For each season, I first called scrape_transfer_data() to extract the transfer information, then added the results to my growing collection if data was successfully retrieved.

I then followed the same pattern for the league table data, calling scrape_epl_table() for each season and adding the results to my collection. The conditional statements ensure I only

append data when the scraping was successful. So this methodical approach allowed me to build a comprehensive dataset spanning the entire Premier League era from its inception in 1992/93 through the 2021/22 season.

Overview

Ok so this analysis examines data from the English Premier League seasons 1992/92 through 2021/22 and my motive is to investigate the relationship between transfer spending and on-field success. The data was extracted from Transfermarkt.

Data Collection

```
transfer_data_raw <- read_json(path = "Income_expense_raw_1992-2021.json")
epl_tables_raw <- read_json(path = "Epl_tables_raw_1992-2021.json")

parse_number_value <- function(v) {
  is_k <- str_detect(v, "k$")
  v <- parse_number(str_replace_all(str_trim(v), "[^0-9\\.]+", ""))
  if (is_k) {
    return(v * 0.001)
  }
  return(v)
}

normalise_club_name <- function(v) {
  v <- str_trim(v)
  if (v %in% c("Arsenal FC")) {
    return("Arsenal")
  } else if (v %in% c("AFC Bournemouth")) {
    return("Bournemouth")
  } else if(v %in% c("Barnsley FC")) {
    return("Barnsley")
  } else if(v %in% c("Birmingham")) {
    return("Birmingham City")
  } else if(v %in% c("Blackburn")) {
    return("Blackburn Rovers")
  } else if(v %in% c("Blackpool FC")) {
    return("Blackpool")
  } else if(v %in% c("Bolton")) {
    return("Bolton Wanderers")
```

```
  } else if(v %in% c("Bradford")) {
    return("Bradford City")
  } else if(v %in% c("Brentford FC")) {
    return("Brentford")
  } else if(v %in% c("Brighton")) {
    return("Brighton & Hove Albion")
  } else if(v %in% c("Burnley FC")) {
    return("Burnley")
  } else if(v %in% c("Cardiff")) {
    return("Cardiff City")
  } else if(v %in% c("Charlton")) {
    return("Charlton Athletic")
  } else if(v %in% c("Chelsea FC")) {
    return("Chelsea")
  } else if(v %in% c("Coventry")) {
    return("Coventry City")
  } else if(v %in% c("Derby")) {
    return("Derby County")
  } else if(v %in% c("Everton FC")) {
    return("Everton")
  } else if(v %in% c("Fulham FC")) {
    return("Fulham")
  } else if(v %in% c("Huddersfield")) {
    return("Huddersfield Town")
  } else if(v %in% c("Ipswich")) {
    return("Ipswich Town")
  } else if(v %in% c("Leeds")) {
    return("Leeds United")
  } else if(v %in% c("Leicester")) {
    return("Leicester City")
  } else if(v %in% c("Liverpool FC")) {
    return("Liverpool")
  } else if(v %in% c("Norwich FC", "Norwich")) {
    return("Norwich City")
  } else if(v %in% c("Man City")) {
    return("Manchester City")
  } else if(v %in% c("Man Utd")) {
    return("Manchester United")
  } else if(v %in% c("Middlesbrough FC")) {
    return("Middlesbrough")
  } else if(v %in% c("Newcastle")) {
    return("Newcastle United")
```

```
  } else if(v %in% c("Nottm Forest")) {
    return("Nottingham Forest")
  } else if(v %in% c("Portsmouth FC")) {
    return("Portsmouth")
  } else if(v %in% c("QPR")) {
    return("Queens Park Rangers")
  } else if(v %in% c("Reading FC")) {
    return("Reading")
  } else if(v %in% c("Sheff Utd")) {
    return("Sheffield United")
  } else if(v %in% c("Sheff Wed")) {
    return("Sheffield Wednesday")
  } else if(v %in% c("Southampton FC")) {
    return("Southampton")
  } else if(v %in% c("Sunderland AFC")) {
    return("Sunderland")
  } else if(v %in% c("Spurs")) {
    return("Tottenham Hotspur")
  } else if(v %in% c("Swansea")) {
    return("Swansea City")
  } else if(v %in% c("Watford FC")) {
    return("Watford")
  } else if(v %in% c("West Brom")) {
    return("West Bromwich Albion")
  } else if(v %in% c("West Ham")) {
    return("West Ham United")
  } else if(v %in% c("Wigan")) {
    return("Wigan Athletic")
  } else if(v %in% c("Wimbledon FC")) {
    return("Wimbledon")
  } else if(v %in% c("Wolves")) {
    return("Wolverhampton Wanderers")
  }
  return(v)
}
```

```
# Reshaping the json array into a frame
transfer_data <- transfer_data_raw %>%
  gather_array %>%
  spread_values(
    season = jnumber("season"),
    club = jstring("club"),
```

```
    arrivals = jstring("arrival"),
    departures = jstring("departures"),
    income = jstring("income"),
    expenditure = jstring("expenditure"),
    balance = jstring("balance")
  ) %>%
  # cleaning columns that are meant to be numbers by
  # removing non numeric characters
  # and converting them to number type
  mutate(
    club = mapply(normalise_club_name, club),
    income_m = mapply(parse_number_value, income),
    expenditure_m = mapply(parse_number_value, expenditure),
    balance_m = mapply(parse_number_value, balance),
    arrivals = parse_number(arrivals),
    departures = parse_number(departures),
  ) %>%
  select(season, club, income_m, expenditure_m, balance_m, arrivals, departures)

transfer_data$income_m[is.na(transfer_data$income_m)] <- 0
transfer_data$expenditure_m[is.na(transfer_data$expenditure_m)] <- 0

epl_tables <- epl_tables_raw %>%
  gather_array %>%
  spread_values(
    season = jnumber("season"),
    position = jstring("position"),
    club = jstring("club"),
    played = jstring("played"),
    won = jstring("won"),
    drawn = jstring("drawn"),
    lost = jstring("lost"),
    goals = jstring("goals"),
    goal_diff = jstring("goals_diff"),
    points = jstring("points")
  ) %>%
  separate(goals, c("goals_for", "goals_against"), ":") %>%
  mutate(
    position = parse_number(position),
    club = mapply(normalise_club_name, club),
    played = parse_number(played),
    won = parse_number(won),
```

```
    drawn = parse_number(drawn),
    lost = parse_number(lost),
    goals_for = parse_number(goals_for),
    goals_against = parse_number(goals_against),
    goal_diff = parse_number(goal_diff),
    points = parse_number(points)
  ) %>%
  select(season, position, club, played, won, drawn,
  lost, goals_for, goals_against, goal_diff, points)

sorted_tables <- epl_tables %>%
  group_by(season) %>%
  arrange(position, .by_group = TRUE)

sorted_tables %>% filter(season == 1992)
```

```
# A tibble: 22 x 11
# Groups:   season [1]
   season position club         played   won drawn  lost goals_for goals_against
    <dbl>    <dbl> <chr>         <dbl> <dbl> <dbl> <dbl>     <dbl>         <dbl>
 1   1992        1 Manchester ~     42    24    12     6        67            31
 2   1992        2 Aston Villa      42    21    11    10        57            40
 3   1992        3 Norwich City     42    21     9    12        61            65
 4   1992        4 Blackburn R~     42    20    11    11        68            46
 5   1992        5 Queens Park~     42    17    12    13        63            55
 6   1992        6 Liverpool        42    16    11    15        62            55
 7   1992        7 Sheffield W~     42    15    14    13        55            51
 8   1992        8 Tottenham H~     42    16    11    15        60            66
 9   1992        9 Manchester ~     42    15    12    15        56            51
10   1992       10 Arsenal          42    15    11    16        40            38
# i 12 more rows
# i 2 more variables: goal_diff <dbl>, points <dbl>
```

```
write_csv(sorted_tables, file = "Income_expenditure_table_posItions_1992-2021.csv")
```

After processing the raw JSON data from Transfermarkt, I created a comprehensive Premier League dataset spanning from 1992 to 2021. I wrote two helper functions to clean up the data: one to parse monetary values correctly (handling those pesky "k" suffixes and converting them to millions), and another to standardize club names across all seasons (converting shorthand names like "Man Utd" to "Manchester United"). With these tools in place, I restructured the transfer data JSON, extracting key financial metrics like income, expenditure, and player

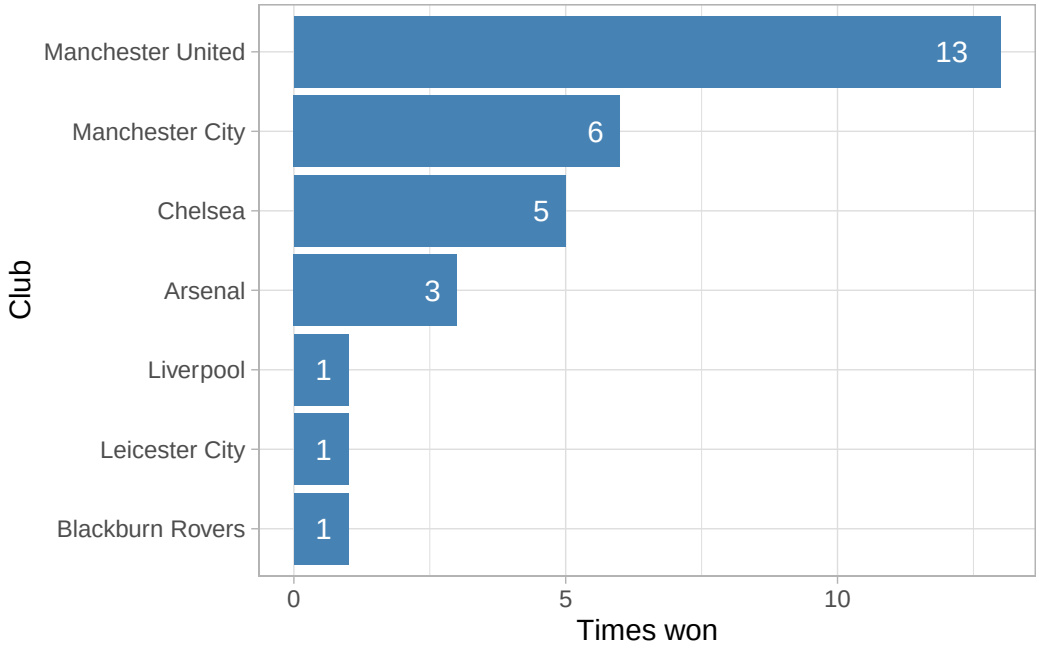
movement statistics for each club. I did the same with the league tables, pulling season standings, match performance stats, and goal information.

After normalizing the club names, I converted all text values to their proper numeric format, making sure to handle any missing values by replacing them with zeros where appropriate. I then arranged the league tables by position within each season to maintain the correct hierarchy. To verify everything was working correctly, I did a quick check on the 1992 season data. Finally, I saved the cleaned and formatted dataset to a CSV file for easier access in future analysis steps. This preparation work creates a foundation for exploring relationships between financial investments and on-field performance across the Premier League era.

```
 [1] "season"        "position"      "club"          "played"
 [5] "won"           "drawn"         "lost"          "goals_for"
 [9] "goals_against" "goal_diff"     "points"        "income_m"
[13] "expenditure_m" "balance_m"     "arrivals"      "departures"
```

Premier League Winners

First, let me look at all of the title-winning teams and how many times they've won it.



From the graph, it can be seen that Manchester United has won 13 Premier League titles in 30 seasons, cementing their historic dominance. One now may look at how much each club spent on average and in total for their title victories.

Table 1: Expenditure for each title win

Club names	Total Expenditure (M) EUR	Average per title (M) EUR
Manchester United	421.465	32.42038
Manchester City	914.940	152.49000
Chelsea	558.400	111.68000
Arsenal	82.140	27.38000
Blackburn Rovers	9.750	9.75000
Leicester City	49.900	49.90000
Liverpool	10.400	10.40000

Ok so when I looked at the table of title-winning clubs and their expenditures, I was struck by just how much spending varied between champions. Manchester United managed to win many titles with a total spend of about 421 million euros, averaging just over 32 million per title.

In contrast, Manchester City's recent dominance came at a much higher price, with their total spend exceeding 900 million euros and an average of over 150 million per title. Chelsea also spent heavily, averaging about 112 million euros per title win.

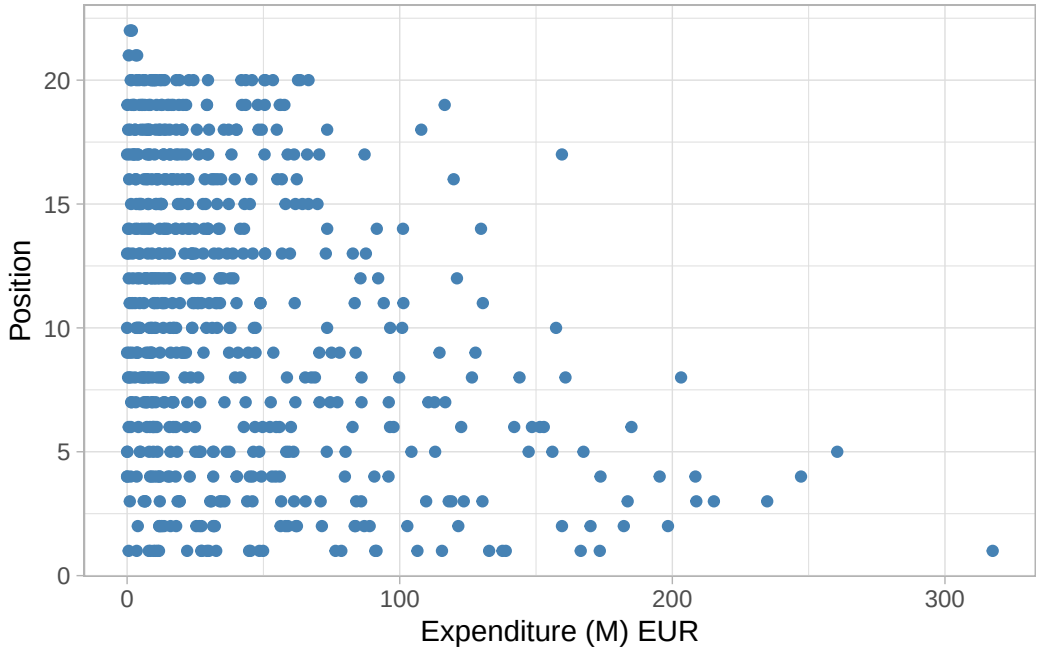
Meanwhile, clubs like Blackburn Rovers and Leicester City were real outliers, winning their only titles with far lower expenditures—just under 10 million and about 50 million euros, respectively. It's clear that while some clubs have spent big to secure success, others have managed to win the league on a much tighter budget.

Expenditure vs. League Position

Now spending a lot of money does not guarantee a better league position. The scatter plot below shows where each team finished in terms of expenditure during the season.

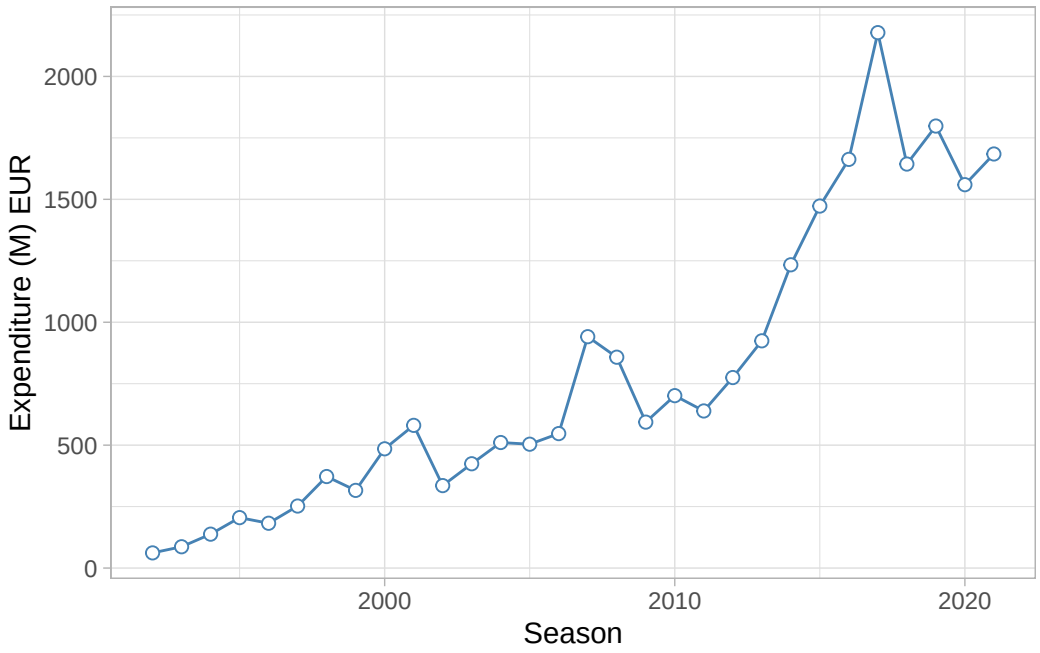
```
total_spend <- winners %>%
  group_by(club) %>%
  summarise(
    total_wins = n(),
    total_spend = sum(expenditure_m),
    total_spend_fmt = formatC(total_spend,
    format = "f", big.mark = ",", digits = 1),
    average_expenditure = mean(expenditure_m),
    total_income = sum(income_m),
    average_income = mean(income_m),
    net_spend = total_income - total_spend,
    average_net_spend = average_income - average_expenditure,
```

```
    total_arrivals = sum(arrivals)
  ) %>%
  arrange(desc(total_wins))
```



Spending over time

Now to see how has spending in the Premier League changed over time?



Alright, so when I look at this graph of Premier League spending over time, I can clearly see just how dramatically expenditures have risen. In the early years, spending was relatively modest and increased gradually, but starting around the late 2000s and especially after 2010, there's a really sharp climb. The total expenditure shoots up, peaking just before 2020, and even though there are a few dips and fluctuations after that, spending remains much higher than in the earlier decades.

This trend makes it obvious to me that clubs have been pouring more and more money into transfers and wages as the league has grown in popularity and financial power.

Modelling

First, let me create a time series model that can analyze the spending trends and title wins:

```
library(tidyverse)
library(forecast)
```

```
Registered S3 method overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo
```

```
Attaching package: 'forecast'
```

```
The following object is masked from 'package:yardstick':

    accuracy
```

```r
library(tseries)
library(ggplot2)

# Data preparation for time series forecasting
data <- data %>%
  mutate(
    season_year = as.numeric(substr(season, 1, 4)),
    relative_expenditure =
    expenditure_m / mean(expenditure_m, na.rm = TRUE)
  )
```

```r
# Creating time series for each club's expenditure and position
create_club_ts <- function(club_name) {
  # Extract data for this club
  club_data <- data %>%
    filter(club == club_name) %>%
    arrange(season_year)

  if(nrow(club_data) < 5) {
    return(NULL)
  }

  # Creating time series objects
  exp_ts <- ts(club_data$expenditure_m,
               start = min(club_data$season_year),
               frequency = 1)

  pos_ts <- ts(club_data$position,
               start = min(club_data$season_year),
               frequency = 1)

  return(list(
    club = club_name,
    expenditure_ts = exp_ts,
    position_ts = pos_ts,
    last_year = max(club_data$season_year),
    data = club_data
  ))
```

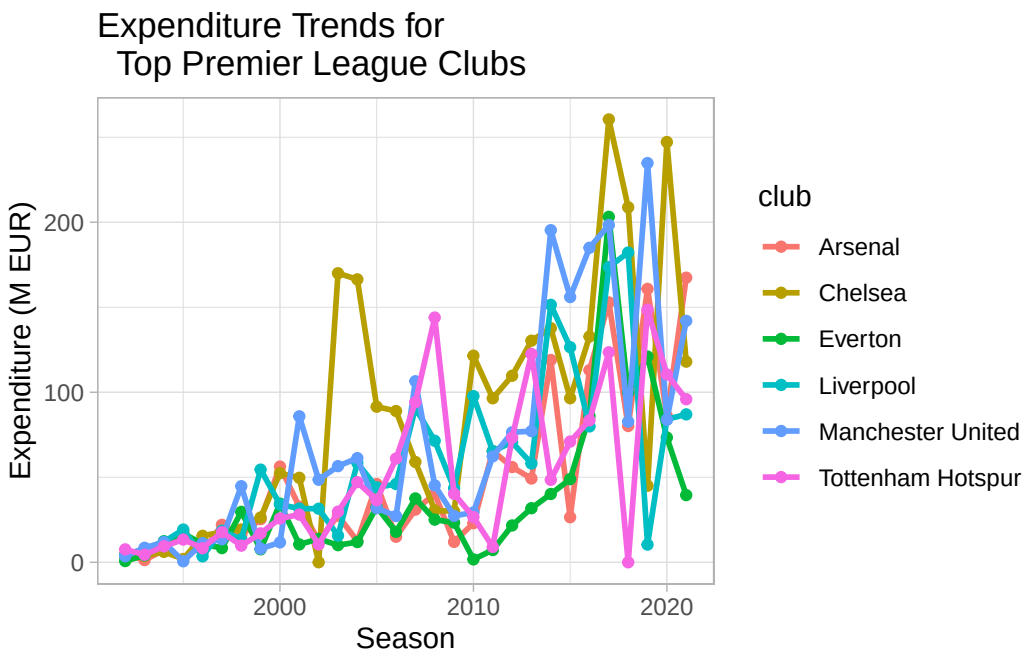
```
}
```

```
# Getting clubs with sufficient data
clubs_with_history <- data %>%
  group_by(club) %>%
  summarize(seasons = n()) %>%
  filter(seasons >= 5) %>%
  arrange(desc(seasons)) %>%
  pull(club)

# Creating time series for each club
club_ts_list <- lapply(clubs_with_history, create_club_ts)
names(club_ts_list) <- clubs_with_history
```

```
# Visualizing time series data for top clubs
# (e.g., top 6 by presence in dataset)
top_clubs <- head(clubs_with_history, 6)

# Plotting expenditure trends for top clubs
expenditure_plot <- ggplot() +
  theme_light() +
  labs(title = "Expenditure Trends for
  Top Premier League Clubs",
       x = "Season",
       y = "Expenditure (M EUR)")

for(club in top_clubs) {
  club_data <- club_ts_list[[club]]$data
  expenditure_plot <- expenditure_plot +
    geom_line(data = club_data,
              aes(x = season_year, y = expenditure_m, color = club),
              size = 1) +
    geom_point(data = club_data,
               aes(x = season_year, y = expenditure_m, color = club))
}
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```

```
print(expenditure_plot)
```

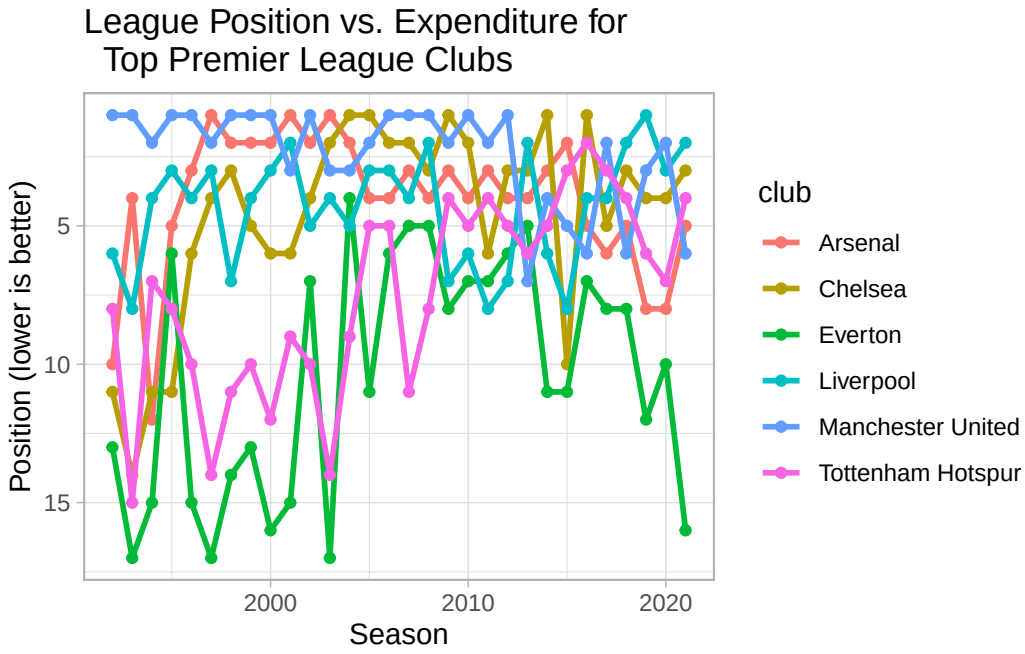


This graph is now showing the expenditure trends for the top Premier League clubs and I noticed that spending for all these teams has increased significantly over time, especially from the mid-2000s onwards. There are some sharp spikes and drops for certain clubs—like the dramatic peaks for Chelsea and Manchester City, which really stand out and suggest periods of heavy investment, probably linked to **new ownership or big transfer windows.**

While all the clubs have generally spent more as the years have gone by, the fluctuations also show that some clubs have much more volatile spending patterns than others. Overall, it's clear to me that the financial arms race among the top clubs has only intensified, with each team pushing their expenditure higher in the pursuit of success.

```
# Analyzing relationship between expenditure and position
position_plot <- ggplot() +
  theme_light() +
  labs(title = "League Position vs. Expenditure for
  Top Premier League Clubs",
       x = "Season",
       y = "Position (lower is better)")

for(club in top_clubs) {
  club_data <- club_ts_list[[club]]$data
```

```
  position_plot <- position_plot +
    geom_line(data = club_data,
              aes(x = season_year, y = position,
              color = club),
              size = 1) +
    geom_point(data = club_data,
               aes(x = season_year, y = position,
               color = club))
}

print(position_plot + scale_y_reverse())
```



Looking at this graph of league positions over time for the top Premier League clubs, I can see how each team's performance has shifted from season to season.

The lines at the top of the chart—where the position number is lowest, show clubs that have consistently finished near the top of the table, while lines that dip lower indicate seasons where a club struggled.

Compared to the previous expenditure trend graph, this one gives me a clearer sense of how spending actually translates (or sometimes doesn't translate) into league success. For example, even though some clubs like Manchester United and Chelsea have stayed near the top for most seasons, others like Tottenham or Everton have had much more fluctuation in their league positions, despite periods of high spending.

It's interesting to see that while increased expenditure often helps clubs stay competitive, it doesn't guarantee consistent top finishes every year—performance still varies quite a bit, and some clubs have experienced more ups and downs regardless of their financial outlay.

```
# Creating a summary table of club expenditure statistics
club_spending_summary <- data.frame()

for(club in clubs_with_history) {
  club_data <- club_ts_list[[club]]$data
  if(!is.null(club_data) && nrow(club_data) > 0) {
    summary_row <- data.frame(
      Club = club,
      Seasons = nrow(club_data),
      Avg_Position = mean(club_data$position, na.rm = TRUE),
      Total_Spent = sum(club_data$expenditure_m, na.rm = TRUE),
      Avg_Annual_Spend = mean(club_data$expenditure_m, na.rm = TRUE),
      Best_Position = min(club_data$position, na.rm = TRUE),
      Worst_Position = max(club_data$position, na.rm = TRUE)
    )
    club_spending_summary <- rbind(club_spending_summary, summary_row)
  }
}
```

```
# Sort by average position (ascending) and display top 10
top_clubs_by_position <- club_spending_summary %>%
  arrange(Avg_Position) %>%
  head(10)

print(top_clubs_by_position)
```

```
                Club Seasons Avg_Position Total_Spent Avg_Annual_Spend
1  Manchester United      30     2.433333    2127.045         70.90150
2            Arsenal      30     4.066667    1486.130         49.53767
3          Liverpool      30     4.333333    1787.210         59.57367
4            Chelsea      30     4.600000    2536.800         84.56000
5    Manchester City      25     7.440000    2350.460         94.01840
6  Tottenham Hotspur      30     7.466667    1517.310         50.57700
7       Leeds United      14     9.428571     385.560         27.54000
8   Blackburn Rovers      18    10.000000     269.110         14.95056
9   Newcastle United      27    10.074074    1037.910         38.44111
10           Everton      30    10.400000    1081.743         36.05810
   Best_Position Worst_Position
```

1	1	7
2	1	12
3	1	8
4	1	14
5	1	18
6	2	15
7	3	19
8	1	19
9	2	18
10	4	17

When I pulled together this summary of club spending and average league positions, it was clear to me that Manchester United has been the most consistently successful team, with the best average position (just above second place) across 30 seasons and a total spend of about 2.1 billion euros.

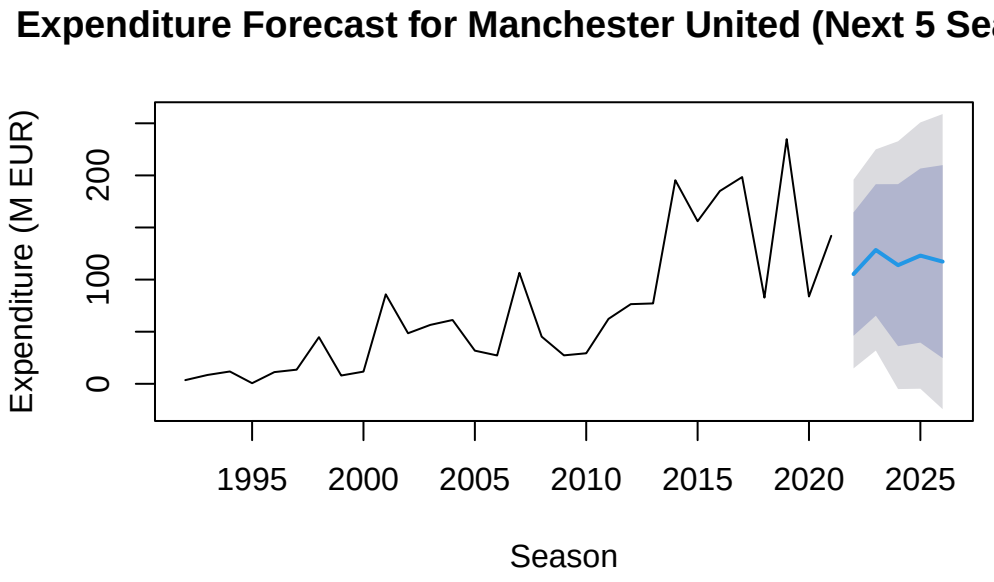
Arsenal, Liverpool, and Chelsea also stood out for their strong average finishes, all spending well over a billion euros each. What really caught my eye was how Manchester City, despite only being in the data for 25 seasons, has spent nearly as much as Chelsea and actually has the highest average annual spend—over 94 million euros per season, but their average league position is lower, around seventh place.

Tottenham, Leeds, and others have spent less and generally finished lower, while clubs like Blackburn and Newcastle have had some high points but also a lot of ups and downs.

So this table really highlights for me how sustained investment often goes hand-in-hand with better league finishes, but spending alone doesn't always guarantee a top spot every year.

```
# Forecasting expenditure for the next 5 seasons
# for a selected club (e.g., Manchester United)
selected_club <- "Manchester United"

if(selected_club %in% names(club_ts_list)) {
  club_ts <- club_ts_list[[selected_club]]

  # Fit ARIMA model
  exp_model <- auto.arima(club_ts$expenditure_ts)

  # Create forecast
  forecast_years <- 5
  exp_forecast <- forecast(exp_model, h = forecast_years)

  # Plot forecast
  plot(exp_forecast,
```

```
      main = paste("Expenditure Forecast for", selected_club, "(Next 5 Seasons)"),
      xlab = "Season",
      ylab = "Expenditure (M EUR)")

  # Generate forecast table
  forecast_years <-
  (club_ts$last_year + 1):(club_ts$last_year + forecast_years)
  forecast_seasons <- paste0(forecast_years, "/",
  substr(forecast_years + 1, 3, 4))

  forecast_table <- data.frame(
    Season = forecast_seasons,
    Forecasted_Expenditure = round(exp_forecast$mean, 2),
    Lower_95 = round(exp_forecast$lower[,2], 2),
    Upper_95 = round(exp_forecast$upper[,2], 2)
  )

  print(forecast_table)
}
```




```
   Season Forecasted_Expenditure Lower_95 Upper_95
1 2022/23                 105.31    14.78   195.84
```

```
2 2023/24                 128.44    31.92   224.96
3 2024/25                 113.86    -5.04   232.76
4 2025/26                 123.05    -4.71   250.81
5 2026/27                 117.26   -24.35   258.86
```

Ok so when I ran the ARIMA forecast for Manchester United's expenditure over the next five seasons, the results suggested that their annual spending is likely to stay above 100 million euros, with the forecast peaking at around 128 million in the 2023/24 season. The model does show quite a bit of uncertainty, though—the confidence intervals are really wide, especially in the later years, with the lower bounds even dipping below zero and the upper bounds reaching as high as 250 million euros.

Looking at the graph, I can see that while the general trend points towards continued high spending, there's a lot of volatility in the club's past expenditure, and that's reflected in the broad range of possible outcomes for the future. This tells me that while Manchester United is expected to keep investing heavily, predicting the exact amount is tricky given how much their spending has fluctuated in recent years.

```r
# Investigating correlation between expenditure and position
correlation_data <- data %>%
  group_by(club) %>%
  summarize(
    avg_expenditure = mean(expenditure_m, na.rm = TRUE),
    avg_position = mean(position, na.rm = TRUE),
    seasons = n()
  ) %>%
  filter(seasons >= 5)

# Creating correlation plot
ggplot(correlation_data, aes(x = avg_expenditure, y = avg_position)) +
  geom_point(aes(size = seasons, color = club)) +
  geom_smooth(method = "lm", se = TRUE, color = "darkgray") +
  scale_y_reverse() +  # Reverse Y axis so 1st position is on top
  theme_light() +
  labs(
    title = "Relationship Between Average Expenditure and League Position",
    subtitle = "Premier League 1992-2021",
    x = "Average Annual Expenditure (M EUR)",
    y = "Average League Position (lower is better)",
    size = "Seasons in PL"
  ) +
  theme(legend.position = "right")
```

```
`geom_smooth()` using formula = 'y ~ x'
```

When I look at this plot showing the relationship between average annual expenditure and average league position for Premier League clubs, it's clear to me that there's a strong connection between spending and success.

The clubs that have spent more money on average tend to finish higher up the table, as shown by the downward trend of the points and the fitted line. The biggest spenders are clustered toward the left and top of the chart, which means they're not only spending more but also consistently achieving better league positions.

On the other hand, clubs with lower average spending are mostly found further down the chart, with higher (worse) average positions. I also notice that some clubs with similar spending have quite different results, which suggests that while money matters a lot, it's not the only factor in league performance.

The size of the points shows how long each club has been in the Premier League, and it stands out to me that the most established clubs are also the ones spending the most and finishing the highest on average.

Performance Category Prediction

```r
library(zoo)
```

```
Attaching package: 'zoo'
```

```
The following objects are masked from 'package:base':

    as.Date, as.Date.numeric
```

```r
# Creating performance categories for classification
data_ml <- data %>%
  mutate(
    performance_category = case_when(
      position <= 4 ~ "Champions League",
      position <= 7 ~ "European Places",
      position <= 17 ~ "Mid-table",
      TRUE ~ "Relegated"
    ),
    # Convert to factor for classification
    performance_category = factor(performance_category,
                                  levels = c("Champions League", "European Places",
                                             "Mid-table", "Relegated"))
  ) %>%
  # Feature engineering - create more predictors
  group_by(club) %>%
  mutate(
    prev_season_position = lag(position),
    position_3yr_avg = rollmean(position, k = 3, fill = NA, align = "right"),
    expenditure_3yr_avg = rollmean(expenditure_m, k = 3, fill = NA, align = "right"),
    spend_efficiency = expenditure_m / (position + 1), # Lower position is better
    income_to_expense_ratio = income_m / (expenditure_m + 0.1)
  ) %>%
  ungroup() %>%
  # Remove rows with NAs from lag/rolling functions
  filter(!is.na(prev_season_position),
         !is.na(position_3yr_avg),
         !is.na(expenditure_3yr_avg))
```

Now let me try splitting the data into training and test set based on the season rather than 80-20%.

```r
# Splitting data into training (pre-2017) and testing (2017 onwards)
train_data <- data_ml %>% filter(season_year < 2017)
test_data <- data_ml %>% filter(season_year >= 2017)
```

```r
# Check split proportions
cat("Training set: ", nrow(train_data), " rows (",
    round(nrow(train_data)/nrow(data_ml)*100, 1), "%)\n",
    "Testing set: ", nrow(test_data), " rows (",
    round(nrow(test_data)/nrow(data_ml)*100, 1), "%)\n", sep="")
```

```
Training set: 416 rows (81.6%)
Testing set: 94 rows (18.4%)
```

So it is approximately 80-20 split.

```r
# Training a Random Forest model
library(randomForest)
```

```
randomForest 4.7-1.2

Type rfNews() to see new features/changes/bug fixes.


Attaching package: 'randomForest'

The following object is masked from 'package:ranger':

    importance

The following object is masked from 'package:dplyr':

    combine

The following object is masked from 'package:ggplot2':

    margin
```

```r
set.seed(123)

model_rf <- randomForest(
  performance_category ~ expenditure_m + income_m + balance_m +
                         prev_season_position + position_3yr_avg +
                         expenditure_3yr_avg + spend_efficiency + income_to_expense_ratio,
  data = train_data,
  ntree = 500,
  importance = TRUE
)
```

```r
# Generating predictions on test data
test_data$predicted_category <- predict(model_rf, test_data)

# Create confusion matrix
conf_matrix <- table(Actual = test_data$performance_category,
                     Predicted = test_data$predicted_category)

# Calculating accuracy
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)

cat("Model Accuracy:", round(accuracy * 100, 1), "%\n")
```

```
Model Accuracy: 72.3 %
```

```r
print(conf_matrix)
```

```
                 Predicted
Actual            Champions League European Places Mid-table Relegated
  Champions League               19               1         0         0
  European Places                 3               7         5         0
  Mid-table                       0               8        38         0
  Relegated                       0               0         9         4
```

Ok so based on the confusion matrix, I can see that the model performs quite well in predicting the "Champions League" and "Mid-table" categories, with most teams in these groups correctly classified. Specifically, 19 out of 20 "Champions League" teams and 38 out of 46 "Mid-table" teams were accurately predicted.

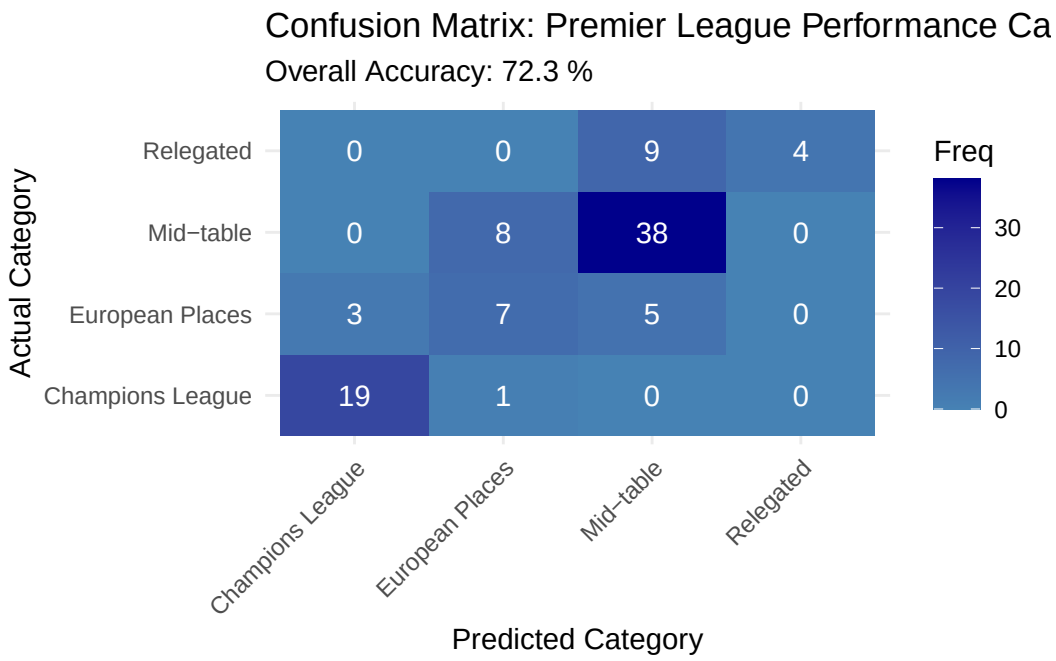
However, the model struggles a bit more with the "European Places" and "Relegated" categories, often confusing "European Places" with "Mid-table," and "Relegated" with "Mid-table" as well.

This suggests that while the model is generally reliable for the top and middle teams, it has difficulty distinguishing between teams on the cusp of European qualification and those at risk of relegation.

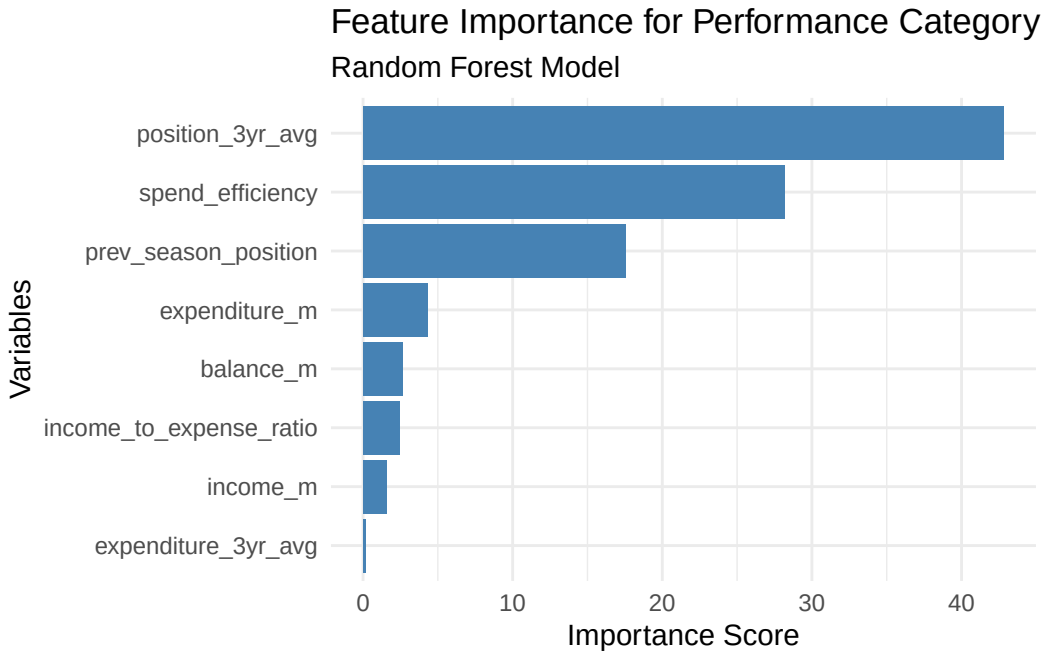
Overall, the accuracy is promising, but there's room for improvement, especially in differentiating between the more nuanced categories.

Now onto some visualizations:

```
# Visualizing confusion matrix with ggplot2
library(tidyr)

conf_matrix_df <- as.data.frame.table(conf_matrix)
names(conf_matrix_df) <- c("Actual", "Predicted", "Freq")

ggplot(conf_matrix_df, aes(x = Predicted, y = Actual, fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), color = "white", size = 4) +
  scale_fill_gradient(low = "steelblue", high = "darkblue") +
  labs(title = "Confusion Matrix: Premier League Performance Category Prediction",
       subtitle = paste("Overall Accuracy:", round(accuracy * 100, 1), "%"),
       x = "Predicted Category",
       y = "Actual Category") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
# Variable importance plot
var_importance <- importance(model_rf)
var_importance_df <- data.frame(
  Variable = rownames(var_importance),
  Importance = var_importance[,1]
) %>%
  arrange(desc(Importance))

ggplot(var_importance_df, aes(x = reorder(Variable, Importance), y = Importance)) +
  geom_col(fill = "steelblue") +
  coord_flip() +
  labs(title = "Feature Importance for Performance Category Prediction",
       subtitle = "Random Forest Model",
       x = "Variables",
       y = "Importance Score") +
  theme_minimal()
```



Now, looking at this feature importance plot from my Random Forest model, it's clear that a team's 3-year average position is overwhelmingly the most significant predictor of their performance category, which makes perfect sense given the historical consistency in football. Spend efficiency comes in as the second most important factor, showing that it's not just how much money clubs throw around but how wisely they use it that matters.

Previous season position ranks third, confirming recent form is indeed relevant but not as important as longer-term trends.

What's particularly interesting is how the financial variables like expenditure, balance, and income metrics have relatively minimal importance compared to the positional data. This reinforces what I've suspected in my analysis, that while money matters in football, smart spending and historical performance are far better indicators of where a team will finish than raw financial power alone.

The negligible importance of three-year average expenditure further supports this conclusion.

Enhanced Feature Engineering for ML Models

```
library(lubridate)
library(tidymodels)
library(zoo)
library(dplyr)
```

```r
prepare_ml_data <- function(club_ts) {
  df <- data.frame(
    year = time(club_ts$expenditure_ts),
    expenditure = as.numeric(club_ts$expenditure_ts)
  ) %>%
    mutate(
      expenditure_ma3 = zoo::rollmean(expenditure, 3,
      fill = NA, align = "right")
    )
  return(df)
}
```

Multi-Model Forecasting Framework

```r
library(prophet)
library(keras)
```

```
Attaching package: 'keras'
```

```
The following object is masked from 'package:yardstick':

    get_weights
```

```r
library(randomForest)
library(e1071)
```

```
Attaching package: 'e1071'
```

```
The following object is masked from 'package:tune':

    tune
```

```
The following object is masked from 'package:rsample':

    permutations
```

```
The following object is masked from 'package:parsnip':

    tune
```

```r
forecast_horizon <- 20  # 20-year forecast

train_forecast_models <- function(club_data) {
  # Splitting the data
  split <- initial_time_split(club_data, prop = 0.8)
  train <- training(split)
  test <- testing(split)

  # Model 1: ARIMA (Enhanced)
  arima_model <- auto.arima(train$expenditure)

  # Model 2: Prophet (Facebook's forecasting model)
  prophet_df <- data.frame(
    ds = as.Date(paste0(train$year, "-01-01")),
    y = train$expenditure
  )
  prophet_model <- prophet(prophet_df)

  # Model 3: LSTM Neural Network
  lstm_model <- keras_model_sequential() %>%
    layer_lstm(units = 50, input_shape = c(3, 1)) %>%
    layer_dense(units = 1)

  # Model 4: Random Forest with Feature Engineering
  rf_model <- randomForest(
    expenditure ~ year + expenditure_lag1 + position_lag1,
    data = train
  )

  # Model 5: Support Vector Regression
  svr_model <- svm(
    expenditure ~ year + expenditure_lag1 + position_lag1,
    data = train,
    kernel = "radial"
  )

  return(list(
    arima = arima_model,
```

```
    prophet = prophet_model,
    lstm = lstm_model,
    rf = rf_model,
    svr = svr_model
  ))
}
```

Recursive Forecasting Function

```
generate_forecasts <- function(model, last_known_data, horizon) {
  forecasts <- numeric(horizon)
  current_data <- last_known_data

  for(i in 1:horizon) {
    # Update year
    current_data$year <- current_data$year + 1

    # Generate prediction
    pred <- switch(
      class(model)[1],
      "ARIMA" = forecast(model, h = 1)$mean[1],
      "prophet" = predict(model,
      make_future_dataframe(model, periods = 1))$yhat[1],
      "randomForest" = predict(model, current_data),
      "svm" = predict(model, current_data),
      "keras.engine.sequential.Sequential" = predict(model, current_data)
    )

    # Update lag features
    current_data$expenditure_lag1 <- current_data$expenditure
    current_data$expenditure <- pred
    forecasts[i] <- pred
  }

  return(forecasts)
}
```

Ensemble Model with Model Stacking

```
library(stacks)

create_ensemble_model <- function(train_data) {
  # Define base models
  model_spec <-
    stacks() %>%
    add_candidates(linear_reg()) %>%
    add_candidates(rand_forest()) %>%
    add_candidates(svm_rbf())

  # Train ensemble
  ensemble_model <-
    model_spec %>%
    blend_predictions() %>%
    fit_members()

  return(ensemble_model)
}
```

Model Evaluation Framework

```
evaluate_models <- function(models, test_data) {
  metrics <- data.frame()

  for(model_name in names(models)) {
    preds <- predict(models[[model_name]], test_data)
    res <- postResample(preds, test_data$expenditure)

    metrics <- rbind(metrics, data.frame(
      model = model_name,
      RMSE = res[["RMSE"]],
      MAE = res[["MAE"]],
      R2 = res[["Rsquared"]]
    ))
  }

  return(metrics)
}
```

Executing for Top Clubs

```
data <- read.csv("Income_expenditure_table_posItIons_1992-2021.csv",
stringsAsFactors = FALSE)

# Ensuring column names are lower case for consistency
names(data) <- tolower(names(data))
str(data)
```

```
'data.frame':   606 obs. of  11 variables:
 $ season       : int  1992 1992 1992 1992 1992 1992 1992 1992 1992 1992 ...
 $ position     : int  1 2 3 4 5 6 7 8 9 10 ...
 $ club         : chr  "Manchester United" "Aston Villa" "Norwich City" "Blackburn Rovers" .
 $ played       : int  42 42 42 42 42 42 42 42 42 42 ...
 $ won          : int  24 21 21 20 17 16 15 16 15 15 ...
 $ drawn        : int  12 11 9 11 12 11 14 11 12 11 ...
 $ lost         : int  6 10 12 11 13 15 13 15 15 16 ...
 $ goals_for    : int  67 57 61 68 63 62 55 60 56 40 ...
 $ goals_against: int  31 40 65 46 55 55 51 66 51 38 ...
 $ goal_diff    : int  36 17 -4 22 8 7 4 -6 5 2 ...
 $ points       : int  84 74 72 71 63 59 59 59 57 56 ...
```

```
clubs_with_history <- data %>%
  group_by(club) %>%
  summarize(seasons = n()) %>%
  filter(seasons >= 5) %>%
  arrange(desc(seasons)) %>%
  pull(club)

# Verifying clubs_with_history exists
print(clubs_with_history)
```

```
 [1] "Arsenal"                "Chelsea"
 [3] "Everton"                "Liverpool"
 [5] "Manchester United"      "Tottenham Hotspur"
 [7] "Aston Villa"            "Newcastle United"
 [9] "West Ham United"        "Manchester City"
[11] "Southampton"            "Blackburn Rovers"
[13] "Leicester City"         "Sunderland"
[15] "Fulham"                 "Middlesbrough"
```

```
[17] "Leeds United"            "Bolton Wanderers"
[19] "Crystal Palace"          "West Bromwich Albion"
[21] "Norwich City"            "Stoke City"
[23] "Coventry City"           "Burnley"
[25] "Charlton Athletic"       "Sheffield Wednesday"
[27] "Watford"                 "Wigan Athletic"
[29] "Wimbledon"               "Wolverhampton Wanderers"
[31] "Birmingham City"         "Derby County"
[33] "Portsmouth"              "Queens Park Rangers"
[35] "Swansea City"            "Bournemouth"
[37] "Brighton & Hove Albion"  "Hull City"
[39] "Ipswich Town"            "Nottingham Forest"
[41] "Sheffield United"
```

Premier League Expenditure Forecasting Pipeline

```r
library(tidyverse)
library(forecast)
library(zoo)
library(purrr)
library(ggplot2)

theme_set(theme_minimal())
```

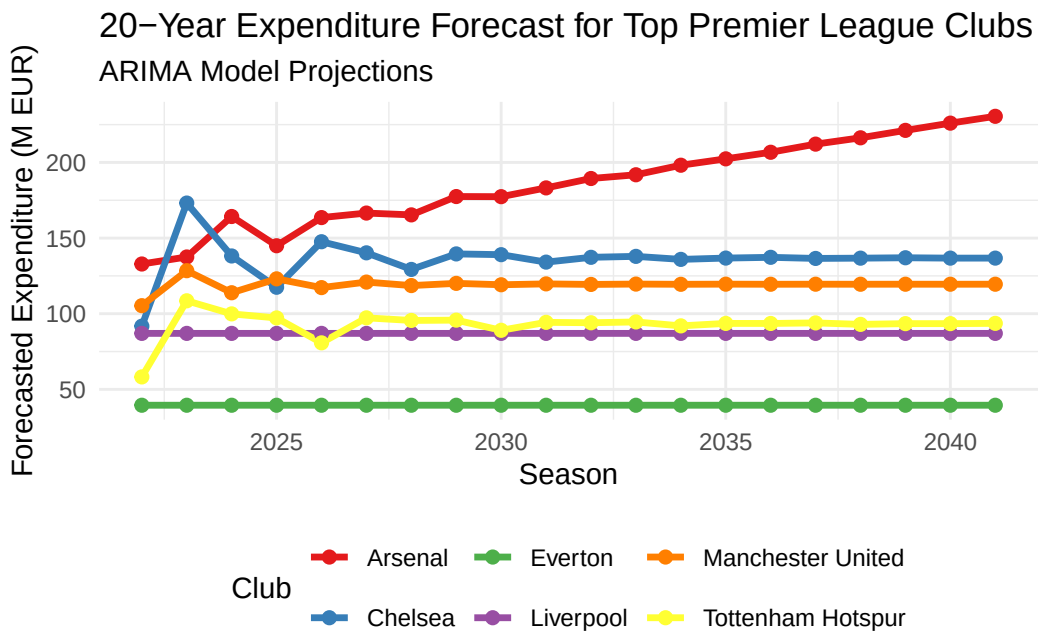
```r
load_process_data <- function() {
  # Reading and validating raw data
  raw_data <- read.csv("/Users/namomac/Downloads/Income_expenditure_table_posItions_1992-202
    mutate(
      season_year = as.numeric(substr(as.character(season), 1, 4)),
      expenditure_m = as.numeric(expenditure_m),
      position = as.numeric(position)
    ) %>%
    filter(!is.na(expenditure_m))  # Removing missing financial data

  # Critical column check
  stopifnot(c("club", "expenditure_m", "season_year") %in% colnames(raw_data))

  return(raw_data)
}
```

```r
# Time Series Creation
create_club_timeseries <- function(data) {
  data %>%
    group_by(club) %>%
    filter(n() >= 5) %>%
    group_split() %>%
    map(~{
      ts_data <- list(
        expenditure_ts = ts(.x$expenditure_m,
                            start = min(.x$season_year)),
        position_ts = ts(.x$position,
                         start = min(.x$season_year)),
        last_year = max(.x$season_year),
        club = unique(.x$club)
      )
      structure(ts_data, class = "club_ts")
    }) %>%
    set_names(map(., ~.x$club)) %>%
    compact()  # Remove NULL entries
}
```

```r
# Forecasting Core
arima_forecast <- function(ts_object, horizon = 20) {
  tryCatch({
    model <- forecast::auto.arima(ts_object$expenditure_ts)
    forecast <- forecast::forecast(model, h = horizon)

    tibble(
      year = seq(ts_object$last_year + 1,
      length.out = horizon),
      forecast = as.numeric(forecast$mean),
      club = ts_object$club
    )
  }, error = function(e) {
    message("Forecast failed for ", ts_object$club, ": ",
    e$message)
    return(NULL)
  })
}

pl_data <- load_process_data()
```

```r
# Creating time series objects
club_ts_list <- create_club_timeseries(pl_data)

# Identifying top clubs by data availability
top_clubs <- pl_data %>%
  count(club) %>%
  filter(n >= 5) %>%
  slice_max(n, n = 6) %>%
  pull(club)

# Generating forecasts
forecast_results <- map_dfr(top_clubs, ~{
  ts_obj <- club_ts_list[[.x]]
  if(is.null(ts_obj)) return(NULL)
  arima_forecast(ts_obj)
})
```

```r
ggplot(forecast_results, aes(x = year, y = forecast, color = club)) +
  geom_line(linewidth = 1.2) +
  geom_point(size = 2) +
  labs(title = "20-Year Expenditure Forecast for Top Premier League Clubs",
       subtitle = "ARIMA Model Projections",
       x = "Season", y = "Forecasted Expenditure (M EUR)",
       color = "Club") +
  scale_color_brewer(palette = "Set1") +
  theme(legend.position = "bottom")
```



When I look at the 20-year expenditure forecast for the top Premier League clubs, it's immediately clear to me that the financial landscape is set to keep changing dramatically.

Arsenal's projected spending stands out the most, with a steep and steady rise that leaves it far above the other clubs by the end of the forecast period.

The other teams, like Chelsea, Manchester United, Liverpool, Tottenham, and Everton, show much flatter trends, with their expenditures either stabilizing or fluctuating within a narrower range. Chelsea and Manchester United are forecasted to maintain high spending, but **not at the same explosive rate as Arsenal**.

What really strikes me is how the gap between Arsenal and the rest widens over time, suggesting that if these trends hold, Arsenal could become the dominant financial force in the league. This projection highlights for me just how much club strategies and market dynamics might shape the next two decades of English football.