Analyzing and Predicting Market Value of Soccer Players

Research Question

Primary Question: Can we use transfer-market and performance data to build a predictive model that estimates a player's expected market value?

Secondary Question: Using the residuals from this model (actual value – expected value), can we identify which players are over- or under-valued and assess the efficiency of the transfer market

Motivation

Player market value is often treated as an indicator of talent and potential, but transfer fees are influenced by many non-performance factors such as media attention, reputation, or league exposure. From a sports-analytics perspective, understanding whether market prices accurately reflect measurable on-field contributions can help clubs avoid overpaying or identify undervalued players in the transfer market.

This question is important because transfer spending among top clubs reaches hundreds of millions of dollars each season, and mispricing even a single player can have large financial consequences. Previous work from StatsBomb, FBref, and academic studies has analyzed market value or performance metrics independently, but relatively few examine how well the market prices players relative to their actual performance. Our project extends this gap by applying the idea of an "expected market value", which is similar to expected metrics like xG or xwOBA, to evaluate pricing fairness.

Objective

Our objective is to estimate expected market value using machine-learning models and use the resulting residuals as an interpretable measure of pricing efficiency. By comparing actual market prices to model-based expectations, we aim to determine

1. whether the transfer market systematically over- or under-values certain players
2. whether valuations align with data-driven performance contributions
3. whether any patterns or biases (e.g., position, nationality, age) appear in mispricing

Through this approach, we provide a unified framework that connects predictive modeling with economic interpretation in soccer analytics.

Data collection and preparation

Data Sources

We used three distinct public soccer datasets: 1. FBref (https://fbref.com/en/) – provides detailed player- and team-level performance statistics compiled from official league data. 2. Understat (https://understat.com/) – provides advanced, model-based xG/xA and shot-level performance metrics. 3. Transfermarkt

(https://transfermarkt.com/) – provides player market values, transfer fees, contract information, club history, and positional data. To collect these datasets, we used the worldfootballR R package (https://github.com/JaseZiv/worldfootballR) as a data-access tool to download FBref and Understat statistics. Transfermarkt data is collected manually due to site structure and access restrictions. This ensures that all three sources remain clearly distinct. FBref and Understat supply performance metrics, while Transfermarkt supplies market valuation data.

Unit of Observation and Time Frame

The unit of observation is player-season across Europe's top five leagues (Premier League, La Liga, Bundesliga, Serie A, Ligue 1). The dataset covers the 2015–2024 seasons (10 years), representing the modern analytics era where detailed event data and xG metrics are consistently available. The combined dataset contains approximately 8,000–12,000 player-season observations.

Key Variables

Performance variables (from FBref & Understat) include: - Offensive: Goals, Assists, xG, xA, Shots, Progressive Passes, Progressive Carries - Defensive: Tackles, Pressures, Interceptions, Clearances - Goalkeeping: Saves, Save Percentage, Clean Sheets, Goals Against Valuation variables (from Transfermarkt) include: - Market Value (EUR), Transfer Fee, Age, Club, League, Contract Expiry, and Position These variables jointly allow us to examine the relationship between measurable on-field performance and market valuation, train a model predicting expected market value, and analyze residuals to identify under- and over-valued players as indicators of pricing efficiency

```
# library(worldfootballR)
# library(dplyr)
#
# # Helper function: scrape player market values for one league + season
# get_league_market_values <- function(country_name, start_year, sleep_time = 1) {
#
#   team_urls <- tm_league_team_urls(
#     country_name = country_name,
#     start_year   = start_year
#   )
#
#   league_df <- data.frame()
#
#   for (each_team in team_urls) {
#     message(paste0("Scraping ", country_name, " ", start_year, " - ", each_team))
#
#     Sys.sleep(sleep_time)
#
#     df <- tryCatch(
#       tm_each_team_player_market_val(each_team_url = each_team),
#       error = function(e) {
#         message("Failed on: ", each_team)
#         return(NULL)
#       }
#     )
#
#     if (!is.null(df)) {
#       league_df <- bind_rows(league_df, df)
```

```
#     }
#   }
#
#   league_df %>%
#     mutate(
#       country = country_name,
#       season_start_year = start_year
#     )
# }
#
# # Big 5 leagues
# big5_countries <- c("England", "Spain", "Germany", "Italy", "France")
#
# # Seasons from 2014 onward
# seasons <- 2014:2024
#
# all_leagues_mv <- data.frame()
#
# for (ctry in big5_countries) {
#   for (yr in seasons) {
#     message(paste0("===== STARTING ", ctry, " ", yr, " ====="))
#     league_df <- get_league_market_values(
#       country_name = ctry,
#       start_year   = yr,
#       sleep_time   = 1
#     )
#     all_leagues_mv <- bind_rows(all_leagues_mv, league_df)
#   }
# }
#
# write.csv(all_leagues_mv, "market_values.csv", row.names = FALSE)
```

```
#
# # FBref Big 5 player stats, 2014-2024, "standard" table
# fbref_big5_standard <- load_fb_big5_advanced_season_stats(
#   season_end_year = 2014:2024,
#   stat_type       = "standard",   # can change to "shooting", "gca", etc.
#   team_or_player  = "player"      # we want player-level stats
# )
#
# glimpse(fbref_big5_standard)
#
# # (Optional) Save to CSV
# write.csv(
#   fbref_big5_standard,
#   "all_leagues_stats.csv",
#   row.names = FALSE
# )
```

```
mv = read.csv("market_values.csv")
stats = read.csv("all_leagues_stats.csv")
```

```r
# Market values: keep relevant cols
library(tidyverse)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.3
```

```
## Warning: package 'tibble' was built under R version 4.4.3
```

```
## Warning: package 'stringr' was built under R version 4.4.3
```

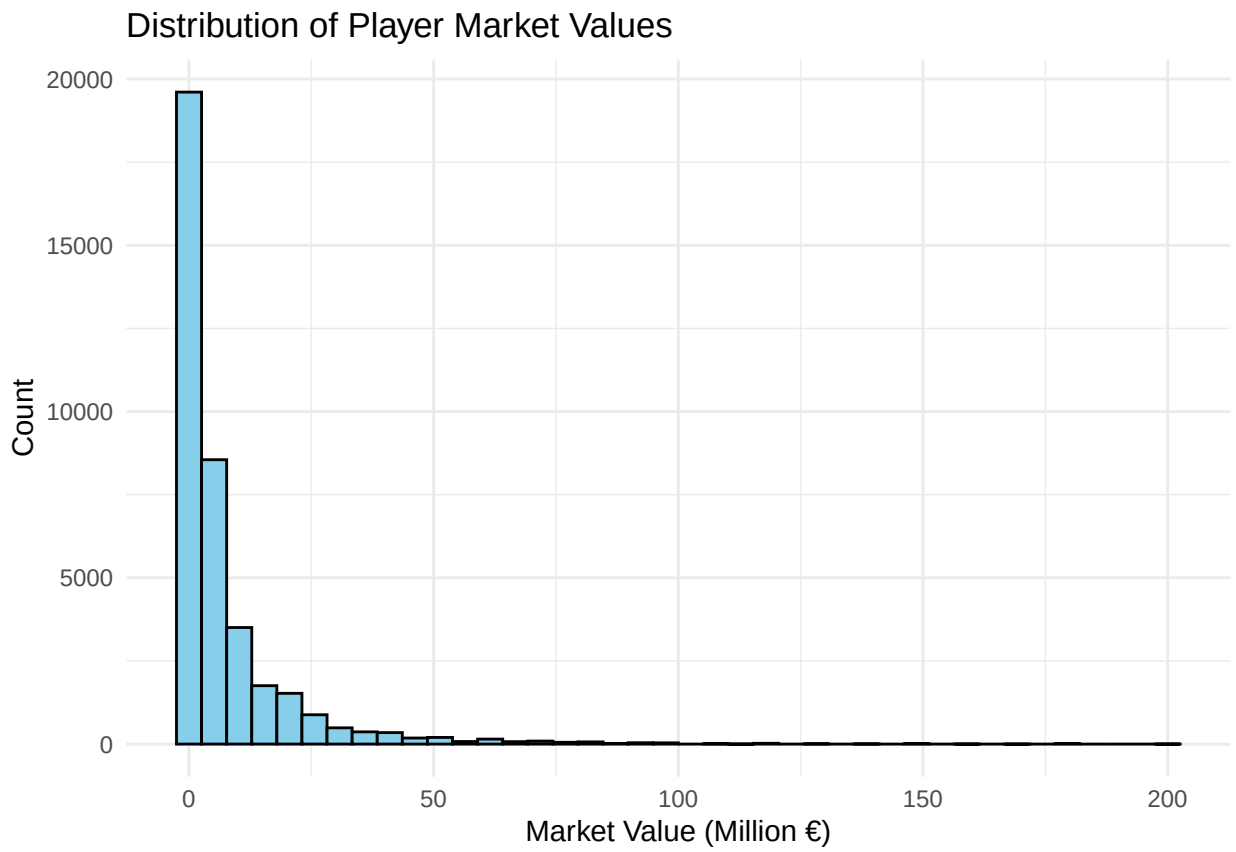
```
## Warning: package 'lubridate' was built under R version 4.4.3
```

```
## -- 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   1.5.2
## v ggplot2   4.0.0     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
mv_clean <- mv %>%
  select(player_name,
         player_age,
         player_position,
         player_market_value_euro,
         country,
         season_start_year,
         squad) %>%
  filter(!is.na(player_market_value_euro))

# FBref stats: keep key performance vars
stats_clean <- stats %>%
  select(Season_End_Year,
         Squad,
         Comp,
         Player,
         Nation,
         Pos,
         Age,
         Min_Playing,
         Gls,
         Ast,
         xG_Expected,
         npxG_Expected)

mv_clean <- mv_clean %>%
  mutate(
    market_value_millions = player_market_value_euro / 1e6
  )
```

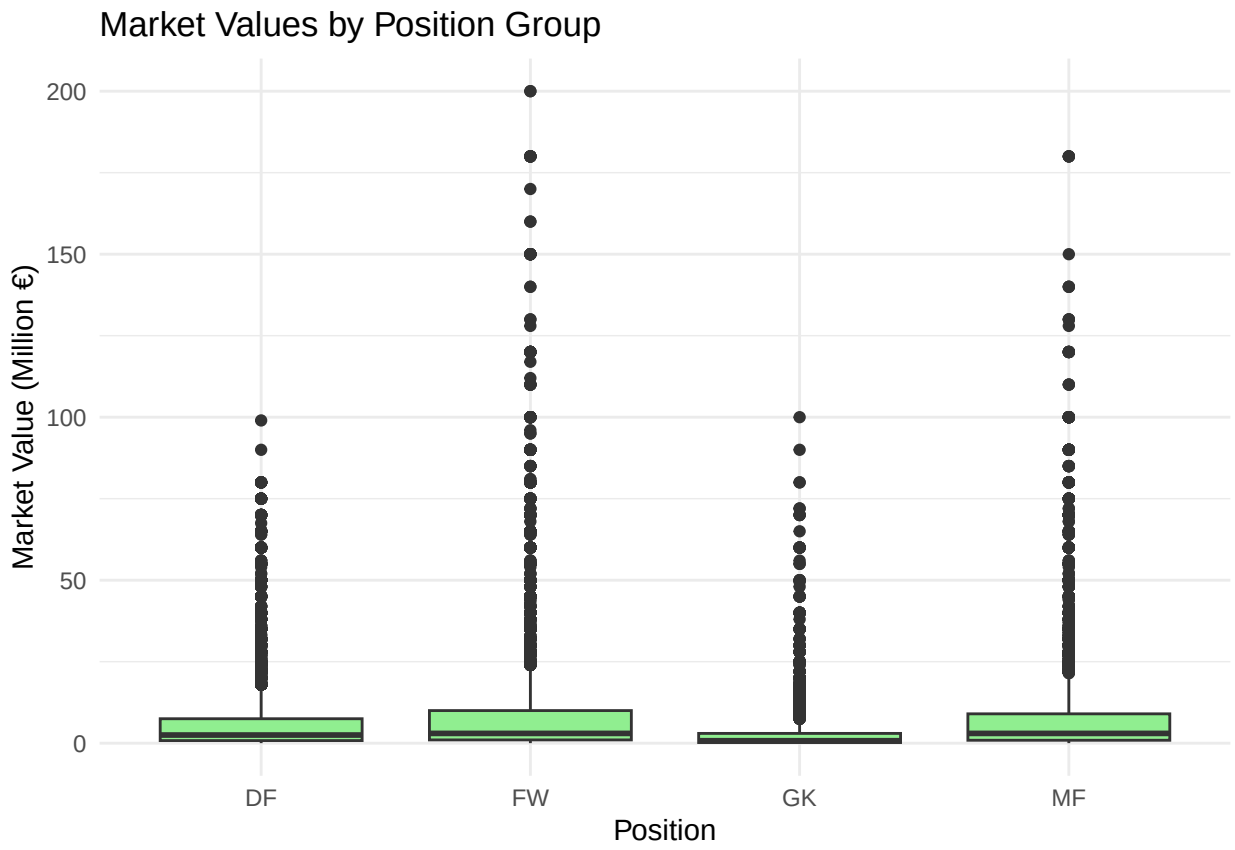
Transfrmarket Data

```
ggplot(mv_clean, aes(x = market_value_millions)) +
  geom_histogram(bins = 40, fill = "skyblue", color = "black") +
  labs(
    title = "Distribution of Player Market Values",
    x = "Market Value (Million €)",
    y = "Count"
  ) +
  theme_minimal()
```



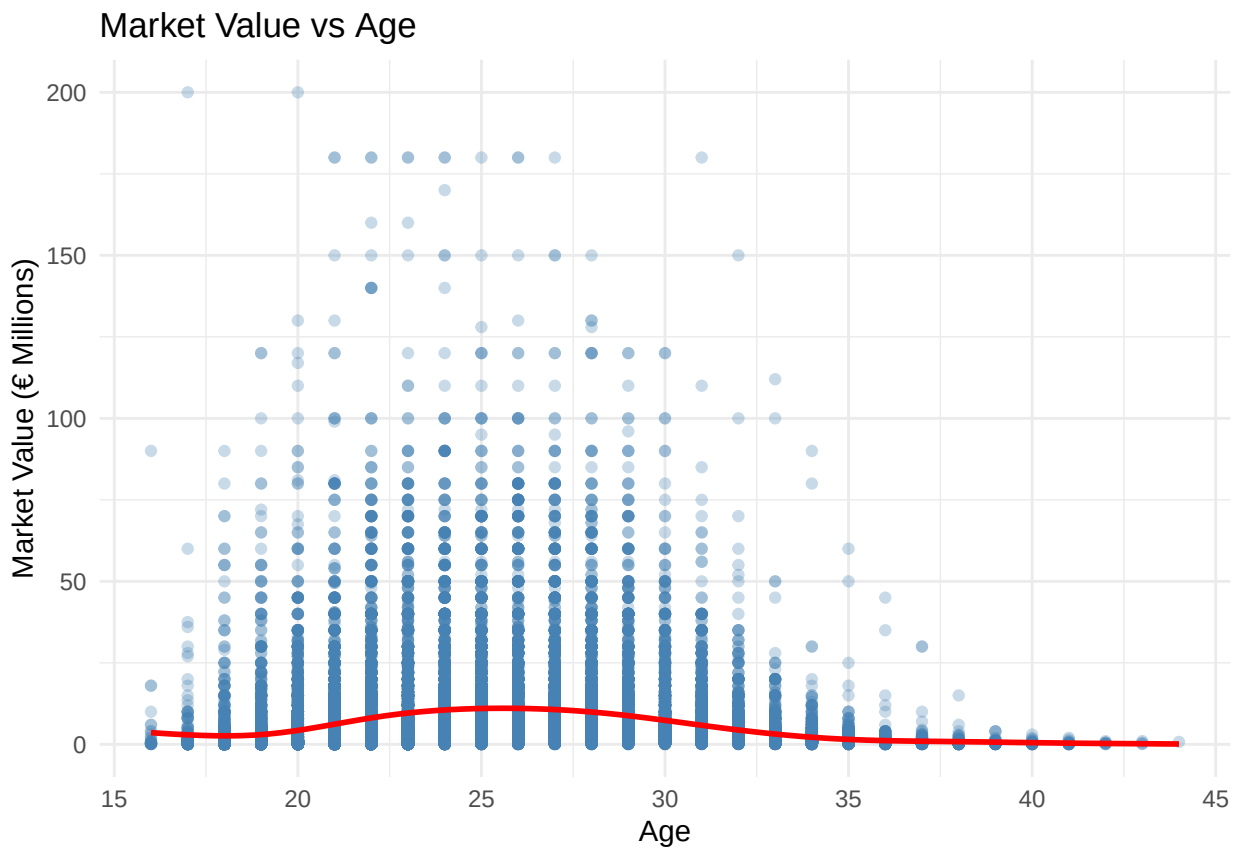

```
mv_clean %>%
  mutate(pos_simple = case_when(
    grepl("keeper", player_position, ignore.case = TRUE) ~ "GK",
    grepl("back|defender", player_position, ignore.case = TRUE) ~ "DF",
    grepl("midfield", player_position, ignore.case = TRUE) ~ "MF",
    grepl("forward|striker|winger", player_position, ignore.case = TRUE) ~ "FW",
    TRUE ~ "Other"
  )) %>%
  ggplot(aes(x = pos_simple, y = market_value_millions)) +
  geom_boxplot(fill = "lightgreen") +
  labs(
    title = "Market Values by Position Group",
    x = "Position",
```

```
    y = "Market Value (Million €)"
  ) +
  theme_minimal()
```

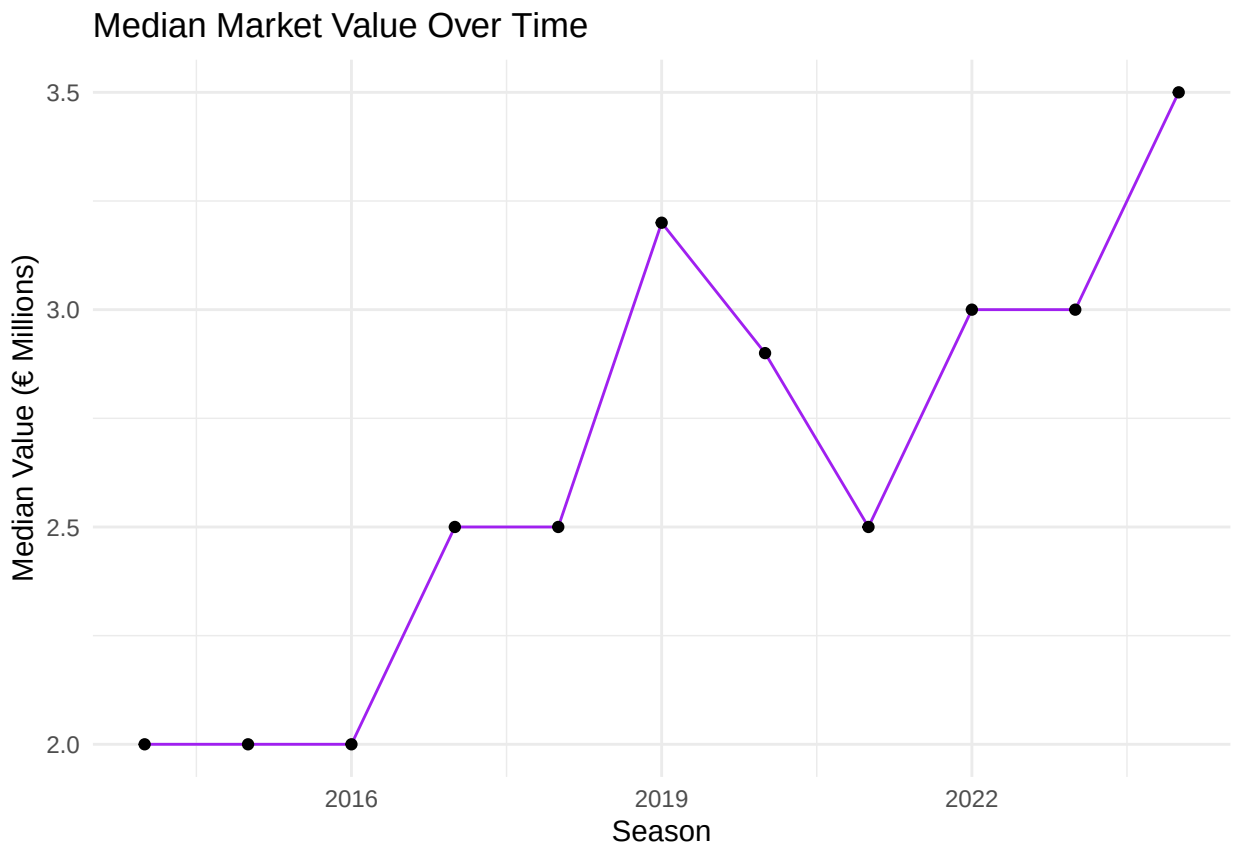



```
ggplot(mv_clean, aes(x = player_age, y = market_value_millions)) +
  geom_point(alpha = 0.3, color = "steelblue") +
  geom_smooth(se = FALSE, color = "red") +
  labs(
    title = "Market Value vs Age",
    x = "Age",
    y = "Market Value (€ Millions)"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

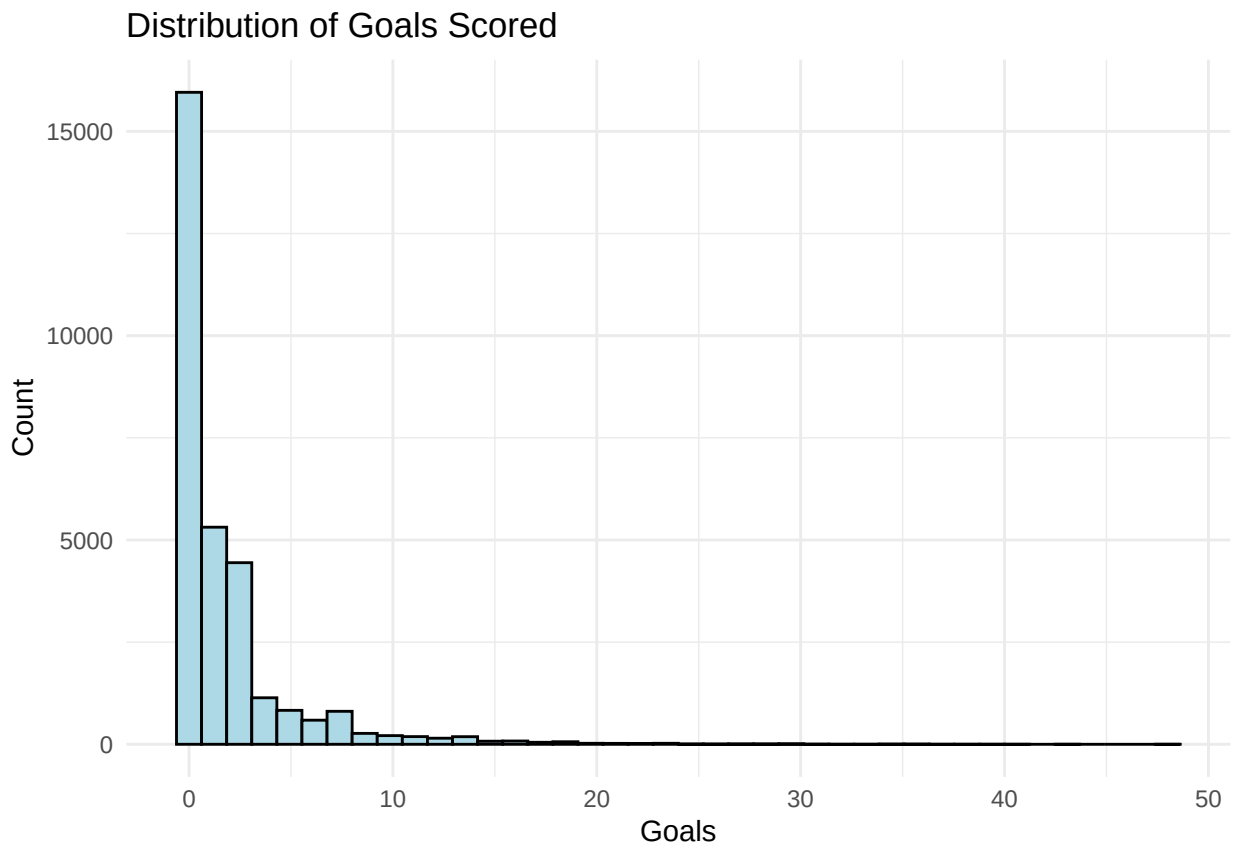



```
mv_clean %>%
  group_by(season_start_year) %>%
  summarise(median_value_millions = median(market_value_millions)) %>%
  ggplot(aes(x = season_start_year, y = median_value_millions)) +
  geom_line(color = "purple") +
  geom_point() +
  labs(
    title = "Median Market Value Over Time",
    x = "Season",
    y = "Median Value (€ Millions)"
  ) +
  theme_minimal()
```



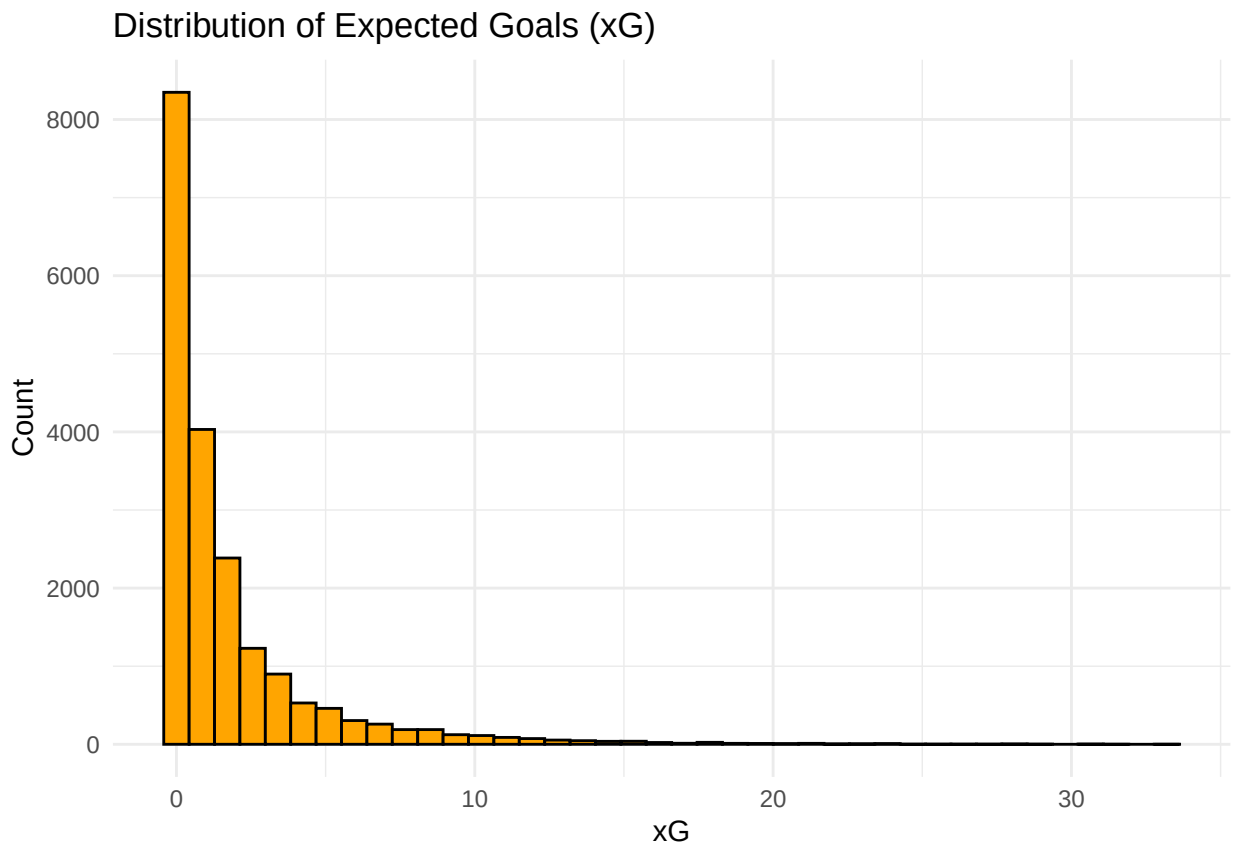
FBREF Data

```
ggplot(stats_clean, aes(x = Gls)) +
  geom_histogram(bins = 40, fill = "lightblue", color = "black") +
  labs(
    title = "Distribution of Goals Scored",
    x = "Goals",
    y = "Count"
  ) +
  theme_minimal()
```



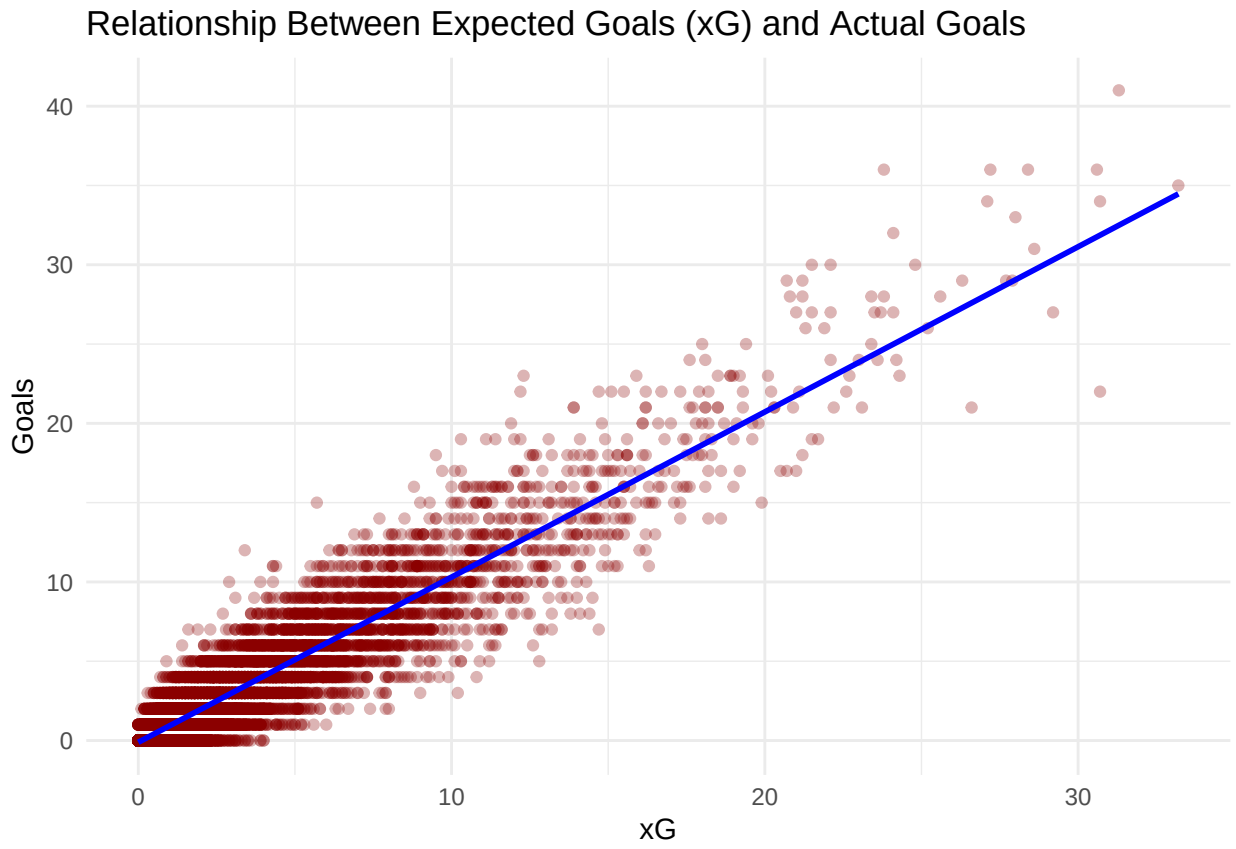

```
ggplot(stats_clean, aes(x = xG_Expected)) +
  geom_histogram(bins = 40, fill = "orange", color = "black") +
  labs(
    title = "Distribution of Expected Goals (xG)",
    x = "xG",
    y = "Count"
  ) +
  theme_minimal()
```

```
## Warning: Removed 10973 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



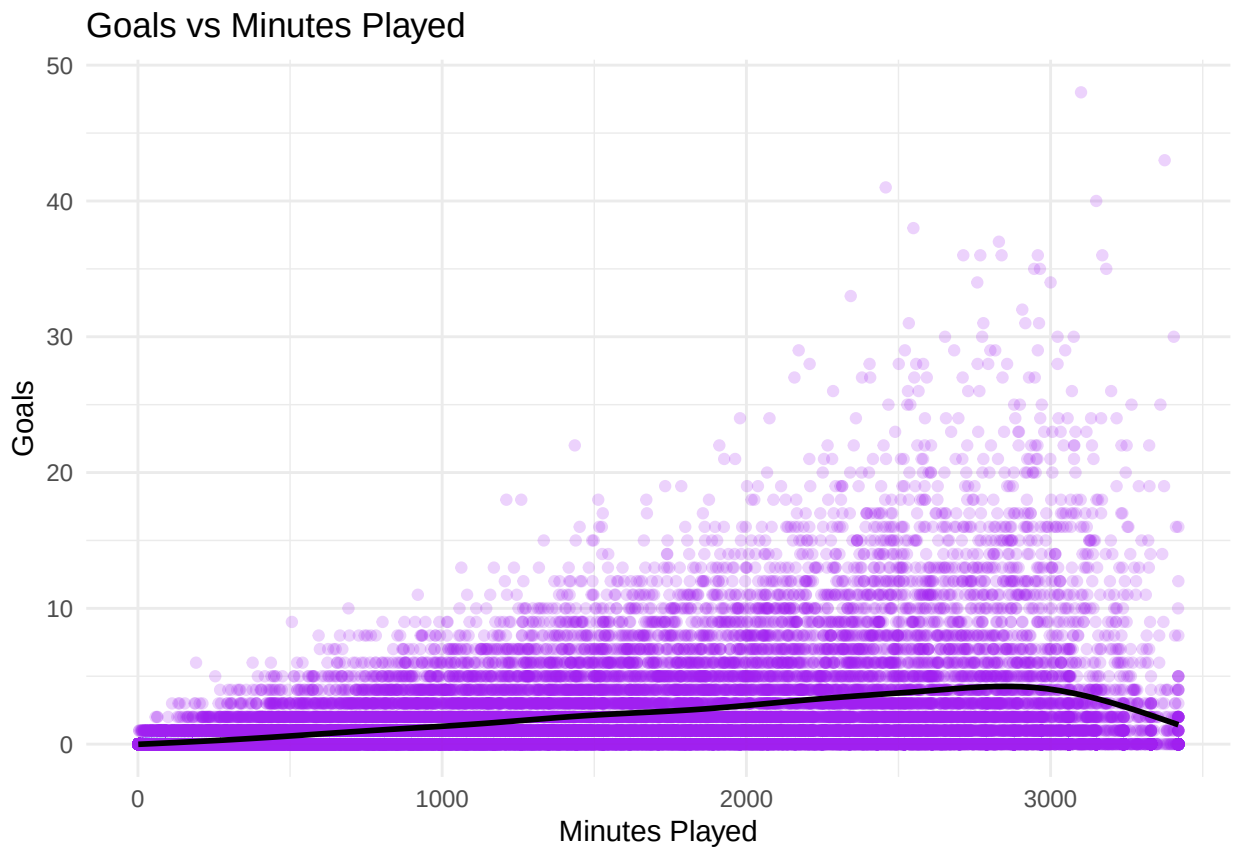

```
stats_clean %>%
  filter(!is.na(xG_Expected), !is.na(Gls)) %>%
  ggplot(aes(x = xG_Expected, y = Gls)) +
  geom_point(alpha = 0.3, color = "darkred") +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(
    title = "Relationship Between Expected Goals (xG) and Actual Goals",
    x = "xG",
    y = "Goals"
  ) +
  theme_minimal()
```

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



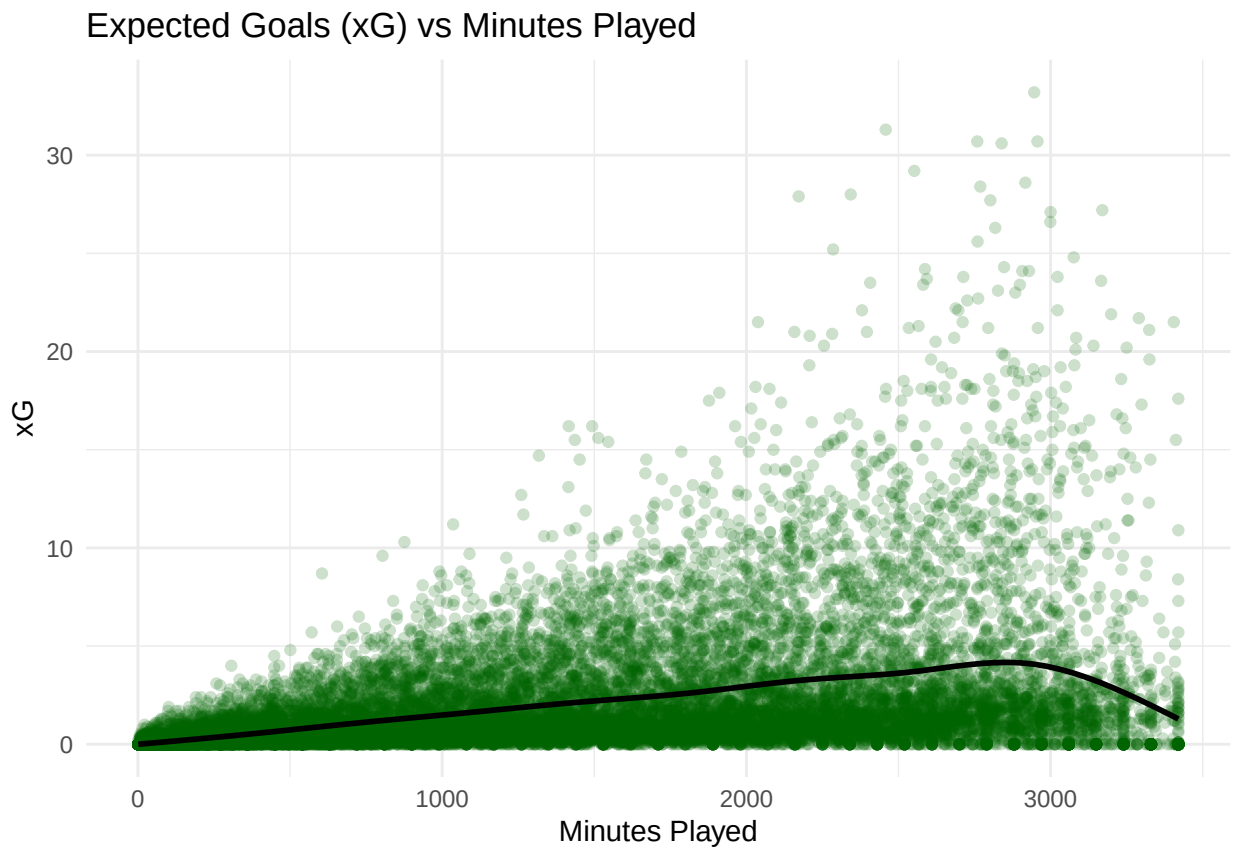

```
stats_clean %>%
  filter(!is.na(Min_Playing), !is.na(Gls)) %>%
  ggplot(aes(x = Min_Playing, y = Gls)) +
  geom_point(alpha = 0.2, color = "purple") +
  geom_smooth(se = FALSE, color = "black") +
  labs(
    title = "Goals vs Minutes Played",
    x = "Minutes Played",
    y = "Goals"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



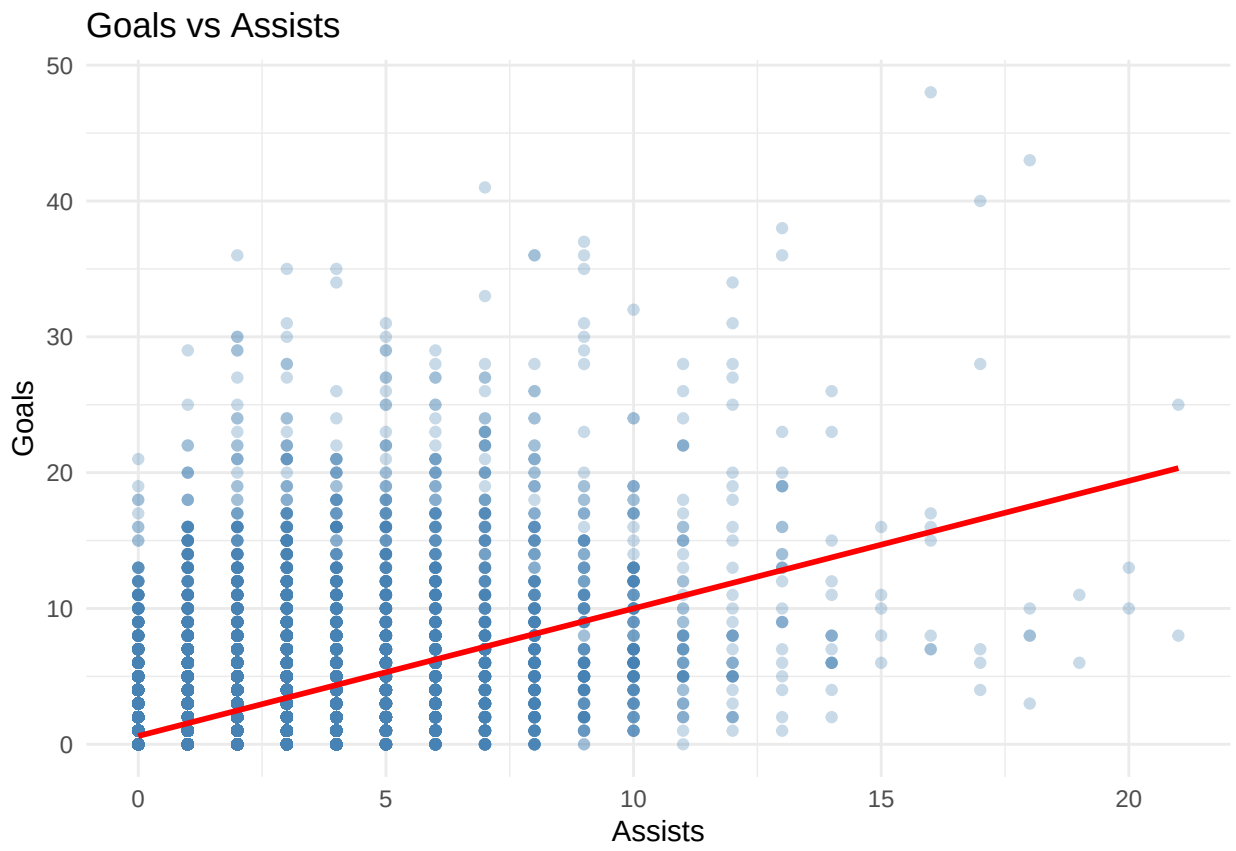

```
stats_clean %>%
  filter(!is.na(Min_Playing), !is.na(xG_Expected)) %>%
  ggplot(aes(x = Min_Playing, y = xG_Expected)) +
  geom_point(alpha = 0.2, color = "darkgreen") +
  geom_smooth(se = FALSE, color = "black") +
  labs(
    title = "Expected Goals (xG) vs Minutes Played",
    x = "Minutes Played",
    y = "xG"
  ) +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```




```
stats_clean %>%
  filter(!is.na(Ast), !is.na(Gls)) %>%
  ggplot(aes(x = Ast, y = Gls)) +
  geom_point(alpha = 0.3, color = "steelblue") +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(
    title = "Goals vs Assists",
    x = "Assists",
    y = "Goals"
  ) +
  theme_minimal()
```

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



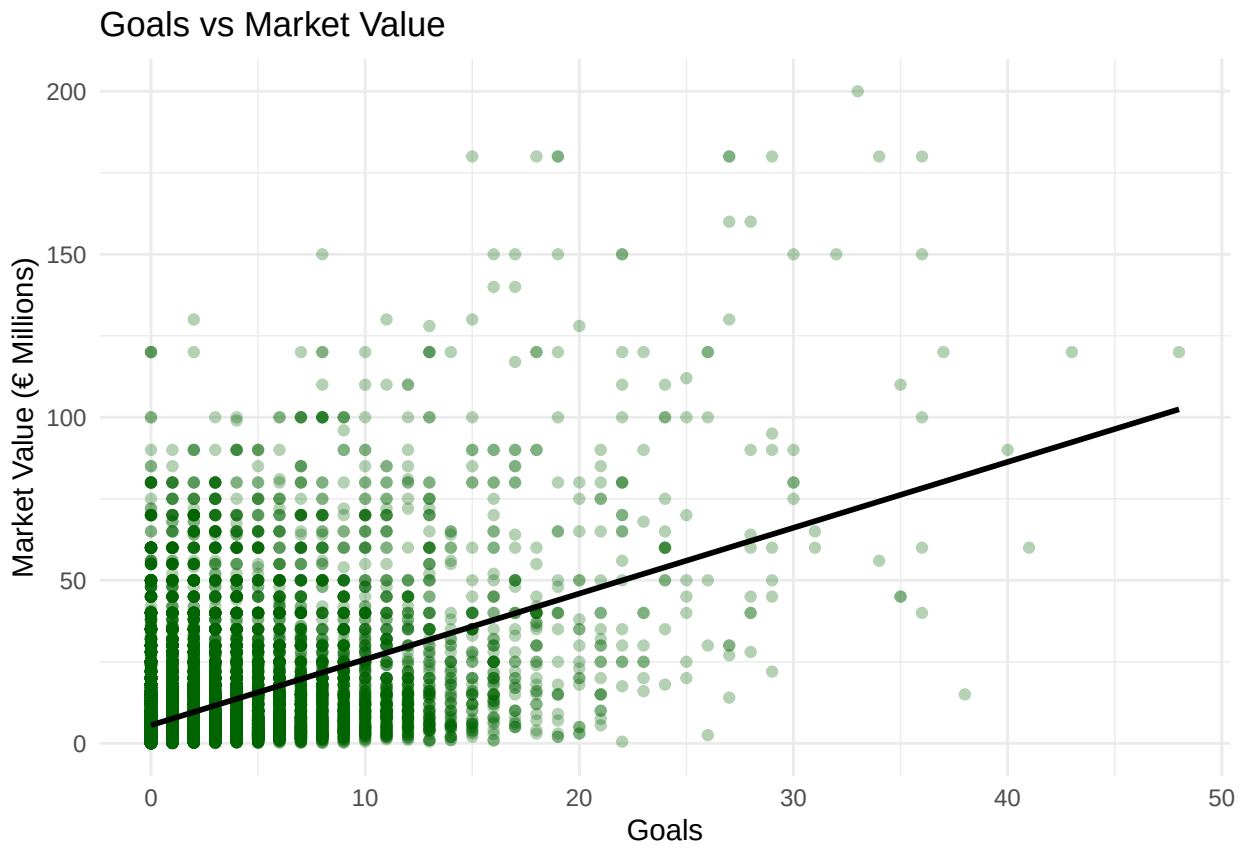

```
stats_key <- stats_clean %>%
  mutate(
    season_start_year = Season_End_Year - 1,
    name_key  = tolower(trimws(Player)),
    squad_key = tolower(trimws(Squad))
  )

mv_key <- mv_clean %>%
  mutate(
    name_key  = tolower(trimws(player_name)),
    squad_key = tolower(trimws(squad))
  )

combined <- inner_join(
  stats_key,
  mv_key,
  by = c("season_start_year", "name_key"),
  relationship = "many-to-many"
)
```

Joined Data

```
combined %>%
  filter(!is.na(Gls), !is.na(market_value_millions)) %>%
```

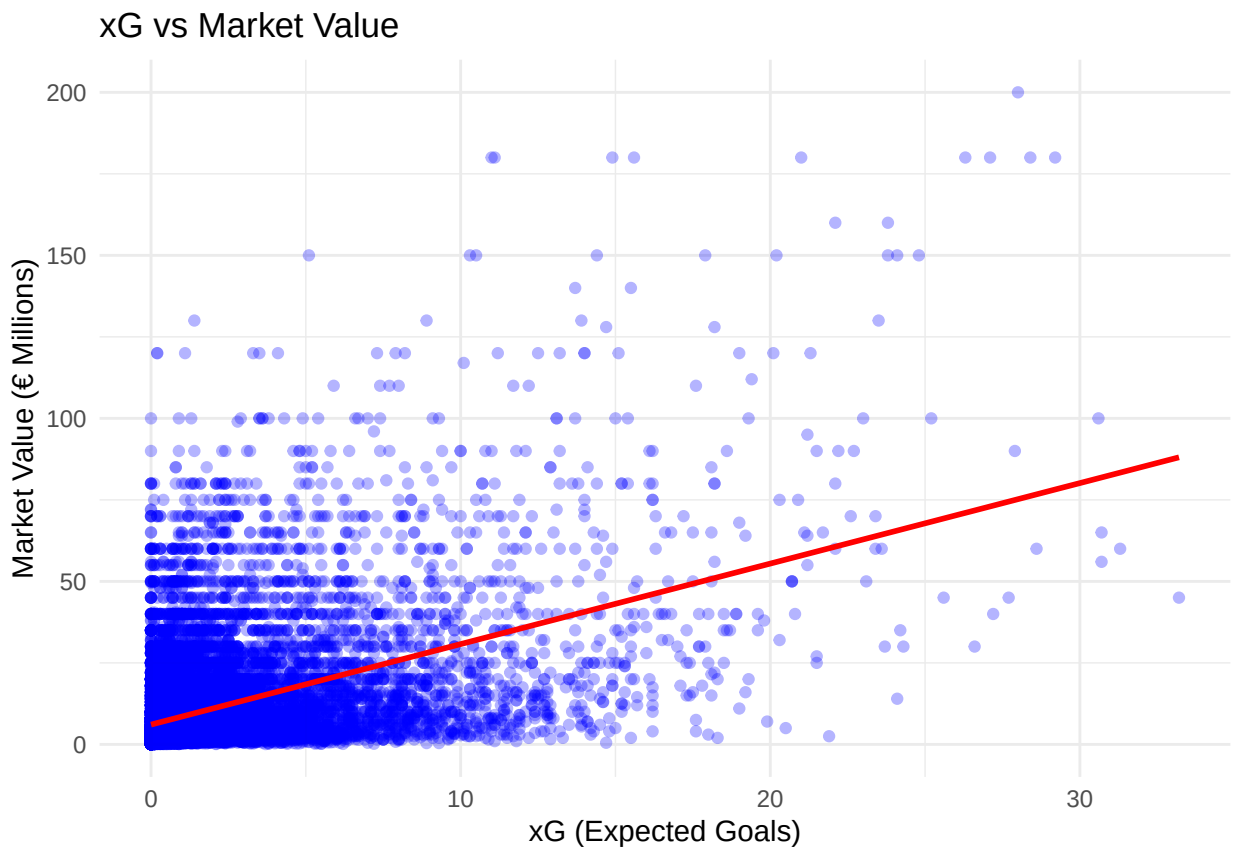
```
  ggplot(aes(x = Gls, y = market_value_millions)) +
    geom_point(alpha = 0.3, color = "darkgreen") +
    geom_smooth(method = "lm", se = FALSE, color = "black") +
    labs(
      title = "Goals vs Market Value",
      x = "Goals",
      y = "Market Value (€ Millions)"
    ) +
    theme_minimal()
```

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



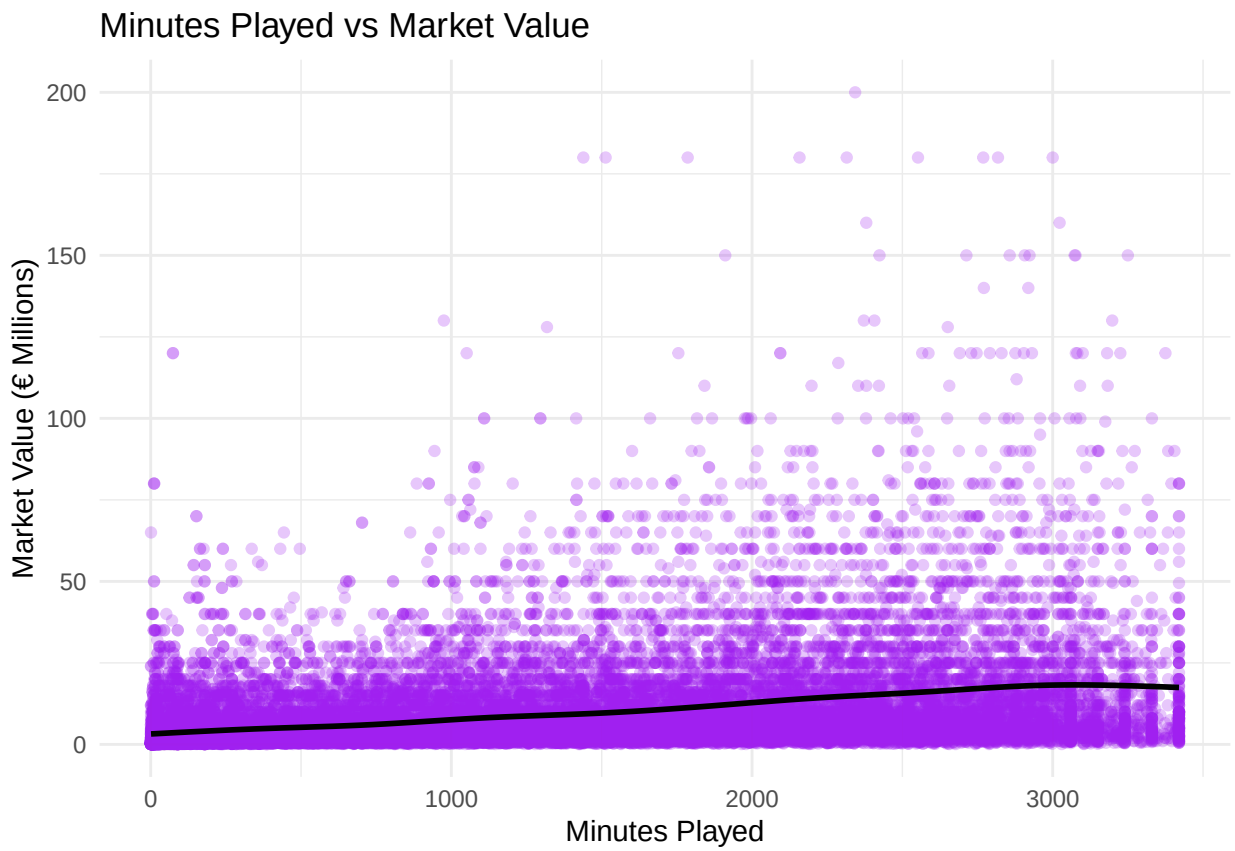

```
combined %>%
  filter(!is.na(xG_Expected), !is.na(market_value_millions)) %>%
  ggplot(aes(x = xG_Expected, y = market_value_millions)) +
    geom_point(alpha = 0.3, color = "blue") +
    geom_smooth(method = "lm", se = FALSE, color = "red") +
    labs(
      title = "xG vs Market Value",
      x = "xG (Expected Goals)",
      y = "Market Value (€ Millions)"
    ) +
    theme_minimal()
```

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



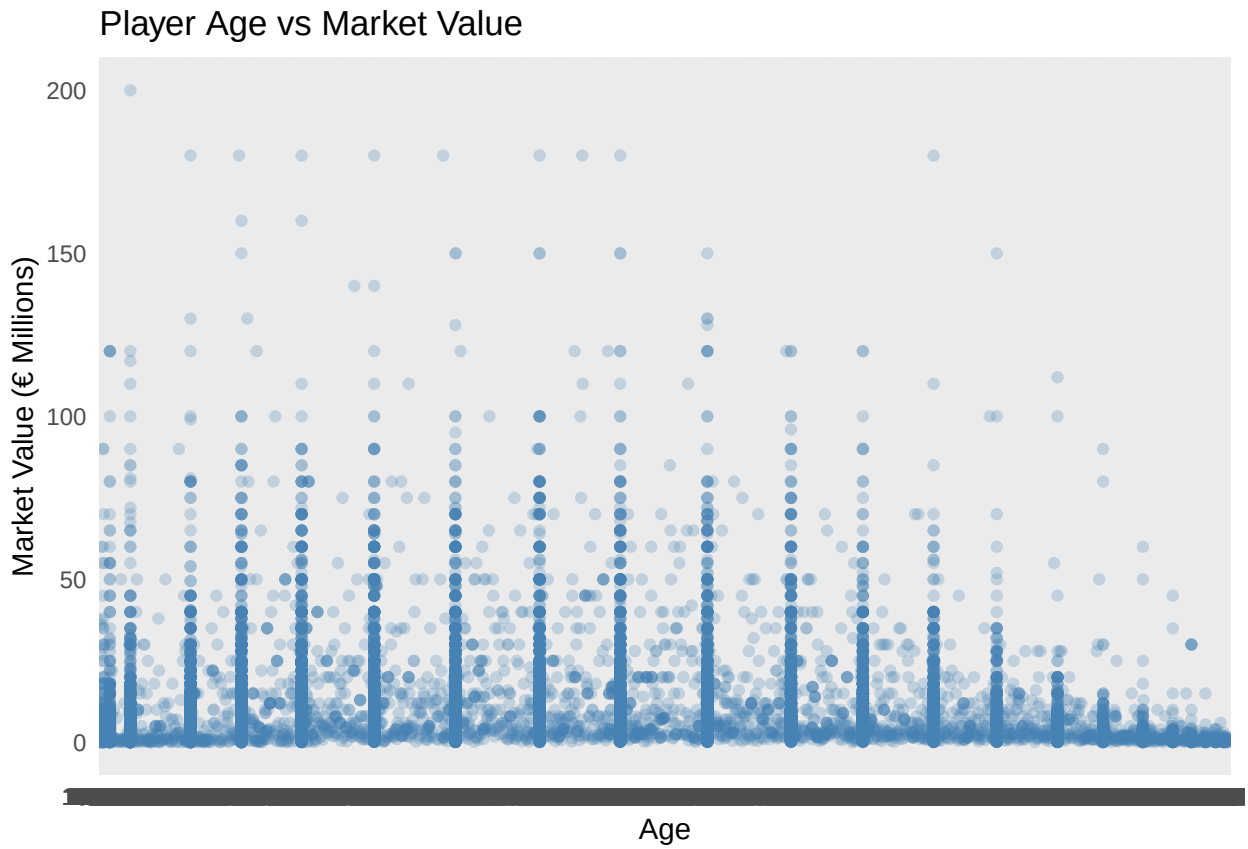

```
combined %>%
  filter(!is.na(Min_Playing), !is.na(market_value_millions)) %>%
  ggplot(aes(x = Min_Playing, y = market_value_millions)) +
    geom_point(alpha = 0.25, color = "purple") +
    geom_smooth(se = FALSE, color = "black") +
    labs(
      title = "Minutes Played vs Market Value",
      x = "Minutes Played",
      y = "Market Value (€ Millions)"
    ) +
    theme_minimal()
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

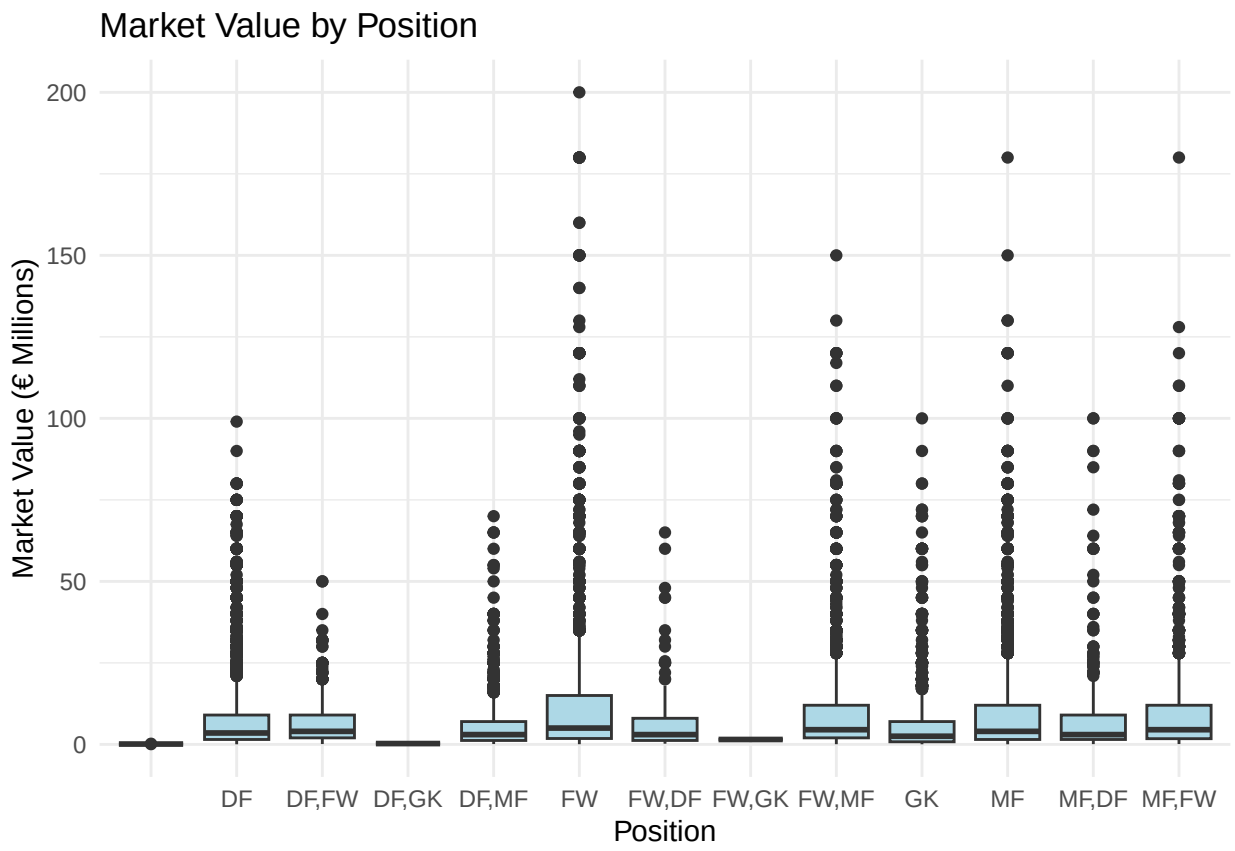



```
combined %>%
  filter(!is.na(Age), !is.na(market_value_millions)) %>%
  ggplot(aes(x = Age, y = market_value_millions)) +
    geom_point(alpha = 0.25, color = "steelblue") +
    geom_smooth(method = "loess", se = FALSE, color = "red") +
    labs(
      title = "Player Age vs Market Value",
      x = "Age",
      y = "Market Value (€ Millions)"
    ) +
    theme_minimal()
```

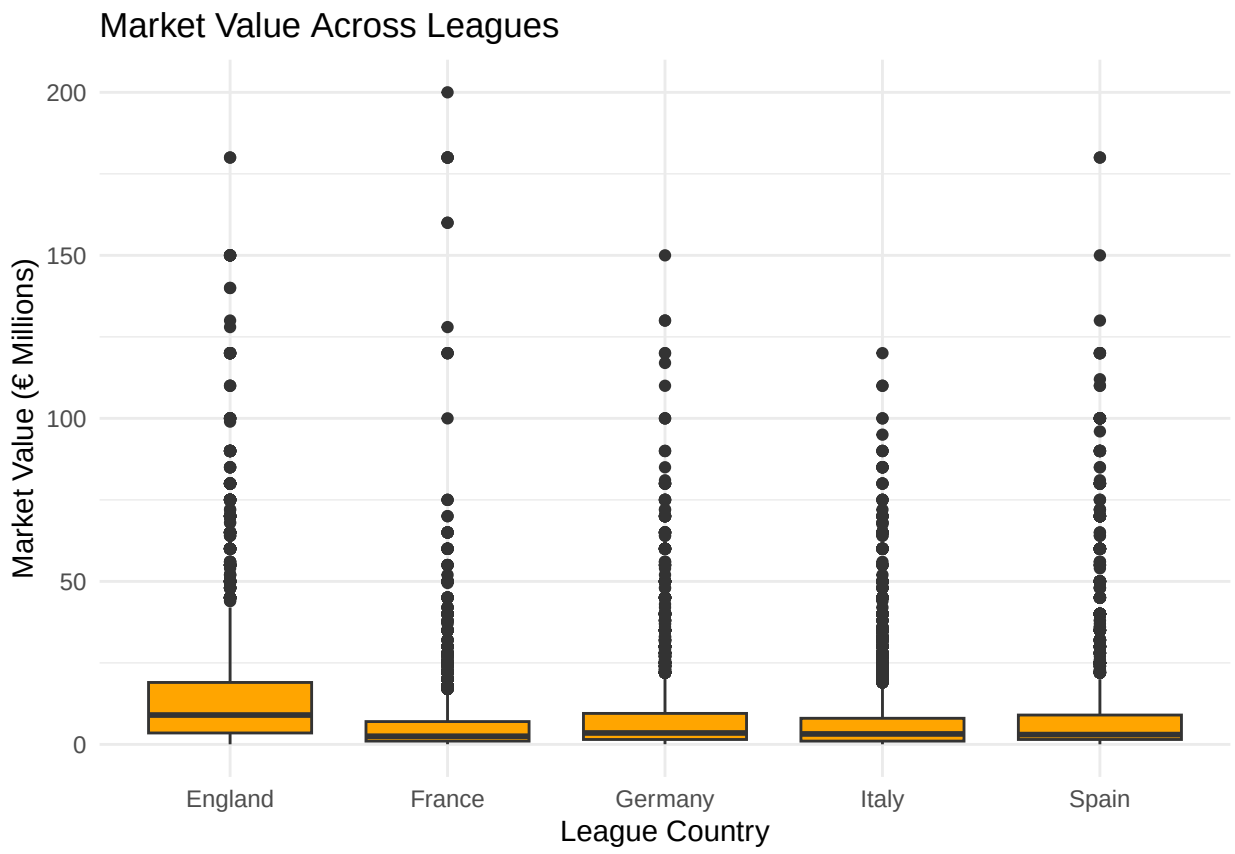
```
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggplot(combined, aes(x = Pos, y = market_value_millions)) +
  geom_boxplot(fill = "lightblue") +
  labs(
    title = "Market Value by Position",
    x = "Position",
    y = "Market Value (€ Millions)"
  ) +
  theme_minimal()
```




```
ggplot(combined, aes(x = country, y = market_value_millions)) +
  geom_boxplot(fill = "orange") +
  labs(
    title = "Market Value Across Leagues",
    x = "League Country",
    y = "Market Value (€ Millions)"
  ) +
  theme_minimal()
```




```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(ggplot2)

# Select relevant numeric columns
corr_df <- combined %>%
  select(
    market_value_millions,
    Min_Playing,
    Gls,
    Ast,
    xG_Expected,
    npxG_Expected
  ) %>%
  mutate(across(everything(), as.numeric)) %>%
  drop_na()

# Compute correlation matrix
```

```r
cor_mat <- cor(corr_df)

# Melt for ggplot
melted <- melt(cor_mat)

# Plot
ggplot(melted, aes(Var1, Var2, fill = value)) +
  geom_tile(color = "white") +
  geom_text(aes(label = round(value, 2)), size = 4) +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0) +
  theme_minimal(base_size = 14) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.title.x = element_blank(),
    axis.title.y = element_blank()
  ) +
  labs(title = "Correlation Heatmap of Key Performance & Market Value Metrics")
```