Gus Pawson - 63949878

Anirudh Revathi - 36899816

Julian Maranan - 81456731

Pretty Mathew Punnoor - 34541658

Uditha Mayadunna- 39768450

DATA 422 – GROUP PROJECT

Investigating football statistics across the top two leagues

1 Contents

2 Introduction

Football is considered by many as the most popular sport in the world, with 207 countries and territories currently holding a FIFA ranking. The beauty of the sport is its simplicity and ability to be played by nearly anyone across the globe; all that is needed is a football. With this popularity and accessibility, it comes as no surprise that it holds the most registered players of any sport. This results in a highly competitive sporting environment, with some of the largest fan bases globally.

The Premier League and La Liga are arguably the top two club leagues in the world. The Premier League is based in the United Kingdom, whilst La Liga is the Spanish equivalent. Even those who aren't typically interested in football are likely to recognise the top teams from both leagues, like Manchester United and Barcelona FC. Given their widespread recognition and passionate fan bases, it's hardly surprising that both leagues have become highly lucrative, evolving into billion-dollar industries.

The goal of this assignment was to compare statistics across these two leagues, to see what is needed to become a championship winning team. The variety and depth of football statistics is astounding, with every possible team or player statistic available online. However, retrieving this data was not as easy as finding it. The report presented below outlines the challenges faced in gathering this data and converting it into a relational database as well as the subsequent analyses.

3 Targets

The aim of these analyses is to define the characteristics of a winning team in the top two football leagues in the world. The analyses are focused on the top seven teams in both leagues as these will be the teams advancing to the Champions League and Europa League, respectively. In addition, the stats investigated was only for the 2015 to 2022 seasons as the wage data only goes far back to 2015. Also, going beyond 2015 will be too much information in terms of visualising the data.
Based on the scraped data, a relational database was created for the purpose of investigating how wins, wages, goals, and fantasy points relate to each other. The data wrangling and analysis of La Liga and Premier League stats were guided by the following objectives:

1. Determining the minimum number of goals required to secure a spot in the top 7.
2. Determining the minimum number of wins required to secure a spot in the top 7.
3. Investigating the influence of player wages on achieving a “winning season”.
4. Analysing the fantasy points distributed between different player positions.
5. Analysing the amount spent by each team to acquire a single league point.
6. Analysing the amount spent by each team to acquire a single fantasy point.

4 Data Scraping

4.1 Data Sources

The majority of data found within our relational database was scraped from *www.fbref.com*. This website contains an abundance of data across all top tier football competitions; if there was a statistic you needed to know, you could find it here. This led to the majority of the datasets found within our database being retrieved from here. However, one component we wished to investigate was missing, fantasy points. This led to the other key source *www.statbunker.com*. Similarly, to *fbref,* Statbunker contained a wealth of data about anything and everything football related. However, for these analyses we were only interested in fantasy points. While data was exclusively collected from these two sources, the distinct tables we hoped to scrape presented different challenges.

4.2 Difficulties Faced

The data available from both websites is relatively tidy. Both have the data presented in easy-to-read tables and with a rather high completeness. If tasked with scraping one table from one year, this would be relatively straightforward process. However, difficulties arise when attempting to collect data from multiple years and the necessary metadata to label it. Both challenges are described in depth below.

4.3 Fbref

As stated above the tables available on *www.fbref.com* are presented in an easy to read format with high completeness. The main issue when scraping from this website is the lack of metadata that comes with the tables. The layout of the website has the year and league presented separately to the tables. When viewing the website online this layout is easy to follow, but when scraping, it presents a clear challenge. As we are scraping solely the information contained within each table, we miss out on this key information, which is necessary for later use in our analyses. Without this information we end up with a table that cannot be formatted into a relational database (Figure 1).

Rk	Squad	MP	W	D	L	GF	GA	GD	⋯	Pts.MP	Attendance	Top.Team.Scorer	Goals Scored	Goalkeeper
<int>	<chr>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	⋯	<dbl>	<int>	<chr>	<int>	<chr>
1	Chelsea	38	27	5	6	103	32	71	⋯	2.26	41423	Didier Drogba	29	Petr Čech

Figure 1:An example of the original data scraped from www.fbref.com, lacking year and league information.

They way in which we combatted this problem was through utilising the information contained in the web links. The formatting of each link is consistent throughout each year and contains vital information

for constructing our relational database. For example, when collecting data on the end of year tables for the 2022/2023 Premier League season, the link is formatted like so:

https://fbref.com/en/comps/9/2022-2023/2022-2023-Premier-League-Stats

Within this link we can find the year and league information we require. As opposed to iterating over an *href* element, like those seen in the lab, we chose to simply create new links based on this format and iterate over these to collect our tables. Through building these links we are able to easily to extract the year and league from each link and create new columns retaining this information. There is likely a way to do this through the *href* element as well, but we found this was an easier option. The result is the additional *Season* and *League* columns seen in *Figure 2*.

Season	League	Rk	Squad	MP	W	D	L	GF	⋯	Pts.MP	Attendance	Top.Team.Scorer	Goals Scored	Goalkeeper
<chr>	<chr>	<int>	<chr>	<int>	<int>	<int>	<int>	<int>	⋯	<dbl>	<int>	<chr>	<int>	<chr>
2009-2010	Premier League	1	Chelsea	38	27	5	6	103	⋯	2.26	41423	Didier Drogba	29	Petr Čech

Figure 2:Additional Season and League columns – information extracted through the created links.

The additional advantage of creating our own links, is the ability to control how far back we wish to collect information. Building the link begins by taking a year (e.g., 2022) and converting this into the format seen above. By iterating over a list of years, we are able to create as many links as necessary and collect only the information needed for our analyses.

By overcoming these obstacles, we are able to combine the scraped data with the necessary metadata for later analyses. Each table now includes the year and league information from which we can begin to form our relational database.

4.4 Statbunker

The website *www.statbunker.com* presented comparable difficulties to those mentioned above. Tables were easy to scrape and relatively tidy but lacked the year and league information. On this website, the user is able to select from separate drop-down menus the year and league they wish to view. As a result, there were no *href* elements which could be utilised to scrape from a separate web page. However, similarly to *fbref*, the web links we wished to scrape were formatted consistently. However, this format did not contain the necessary league and year information like that above. Instead, each web link had a *comp_id* element that contained the information we sought. An example is given below:

https://www.statbunker.com/competitions/FantasyFootballPlayersStats?comp_id=718

For each link, the *comp_id* is the only component which changes. The process begins by selecting a base link, such as the one provided above, which contains data relevant to the league we intend to scrape. In this example, comp_id=718 is relevant to the Premier League. From this, we were able to select all options available for the *comp_id* element that related to the Premier League. Each option contained two components. The first, is the attribute, which contained the comp_id number for all Premier League links. This was used to build a list of links that we later iterated over. The second, is the

text element, which held the league and year information that we need. From this base link, we were able to create two lists, one which held each link we wished to iterate over, and a second containing the corresponding year and league information. As a result, when scraping each link, the corresponding year and season columns were produced simultaneously (Figure 3).

Season	League	Players	Points	Clubs	Position	Start	Goals	A	···	Yellow	Red	Sub	CO	Off	Pen.SV	Pen.M	Goals.conceded
<chr>	<chr>	<chr>	<int>	<chr>	<chr>	<int>	<int>	<int>	···	<int>	<int>	<int>	<int>	<int>	<chr>	<int>	<chr>
2022/2023	Premier League	Erling Haaland	267	MCFC	Forward	33	36	8	···	5	0	3	2	12	-	0	-
2022/2023	Premier League	Harry Kane	227	SPURS	Forward	38	30	3	···	6	0	0	0	3	-	1	-

Figure 3:Example dataset collected from www.statbunker.com, containing Season and League columns. Columns derived from information retained in comp_id components.

Unlike the *fbref* links, the selection of years was not possible for *statbunker*. However, as we were iterating over a list of web links, the number of years could be controlled through indexing.

5 Data Wrangling

The resulting dataframes from the scraping component of this report contained all the data necessary for our analyses. However, these alone were not sufficient for the tasks we wished to complete. In order for these dataframes to become a relational database, primary and foreign key were needed. Additionally, further data processing steps were necessary for us to be able to plot the data. The steps taken to accomplish this are outlined below.

5.1 Relational Database

The creation of primary and foreign keys was needed to allow the individual tables scraped from our sources to be joined and create our final relational database. These were produced from existing columns held within each dataframe. The steps taken to create the three keys needed are outlined below, as well as an example from our *summary* dataframe (Figure 4):

1. Season_ID

   The *Season* was created through combining the values from the *Season* and *League* columns. To avoid creating excessively long IDs, both the *Season* and *League* columns were abbreviated. For example, the 2019-2020 Premier League season was condensed to 19/20-PL (Figure 4).

2. Team_ID

   The *Team_ID* was generated by using the corresponding values from the *Squad* (or *Champions* or *Clubs* – dependent on which dataframe) and *Season* columns. Similarly, to

Season_ID, the *Season* was component was shortened to avoid excessively long IDs. For example, the Liverpool squad from the 2019-2020 season was condensed to 19/20-Liverpool (Figure 4).

3. Player_ID

This was identical to *Team_ID* but held the *Top_Goal_scorer* (or *Player*) and *Season* values. For example, Jamie Vardy was the top goal scorer for the 2019-2020 season, and this was converted to 19/20-Jamie Vardy (Figure 4).

Season_ID	Season	League	Team_ID	Champions	Games_Won	Player_ID	Top_Goal_Scorer	Goals_Scored
<chr>	<chr>	<chr>	<chr>	<chr>	<int>	<chr>	<chr>	<int>
19/20-LL	2019-2020	La Liga	19/20-Real Madrid	Real Madrid	26	19/20-Lionel Messi	Lionel Messi	25
19/20-PL	2019-2020	Premier League	19/20-Liverpool	Liverpool	32	19/20-Jamie Vardy	Jamie Vardy	23

Figure 4:Excerpt from the summary table – highlighting the 3 keys necessary to create our relational database.

By creating these IDs, any information on each season, team, or player, could easily be retrieved from the several datasets available. As a result, our several distinct datasets became a functioning relational database.

From the extensive web scraping and wrangling detailed above, we generated a database containing just under 29,000 rows of data and just under 50 distinct variables. This database is comprised of seven individual tables, a brief description of each is detailed below:

- *Summary* - This contains a summary of the end of season results. Included is the championship winning team and the top goal scorer for a given year.

- *TablesPL* - This is the end of year tables for the Premier League competition. Aside from each team's final ranking, it also contains a variety of statistics detailing how well each team performed.

- *TablesLL* - This contains the end of year tables for the La Liga competition.

- *FantasyPL* - This contains detailed statistics for each player, in the Premier League, for a given season, including their fantasy points.

- *FantasyLL* - This is the equivalent for the La Liga competition.

- *WagesPL* - This contains the annual and weekly wages paid by teams in the Premier League.

- *WagesLL* - This is the La Liga equivalent.

Without the *Season_ID, Team_ID,* and *Player_ID* these distinct datasets cannot be joined sensibly. These three columns are interchangeable as primary or foreign keys depending on which table you wish to merge with. The outcome is the capacity to effectively consolidate data from multiple tables,

providing insights at three different levels: season, team, and player. The data we have collected all pertain to the same entity and thus no predefined relationships are required. The relational database can be visualised below (Figure 5).

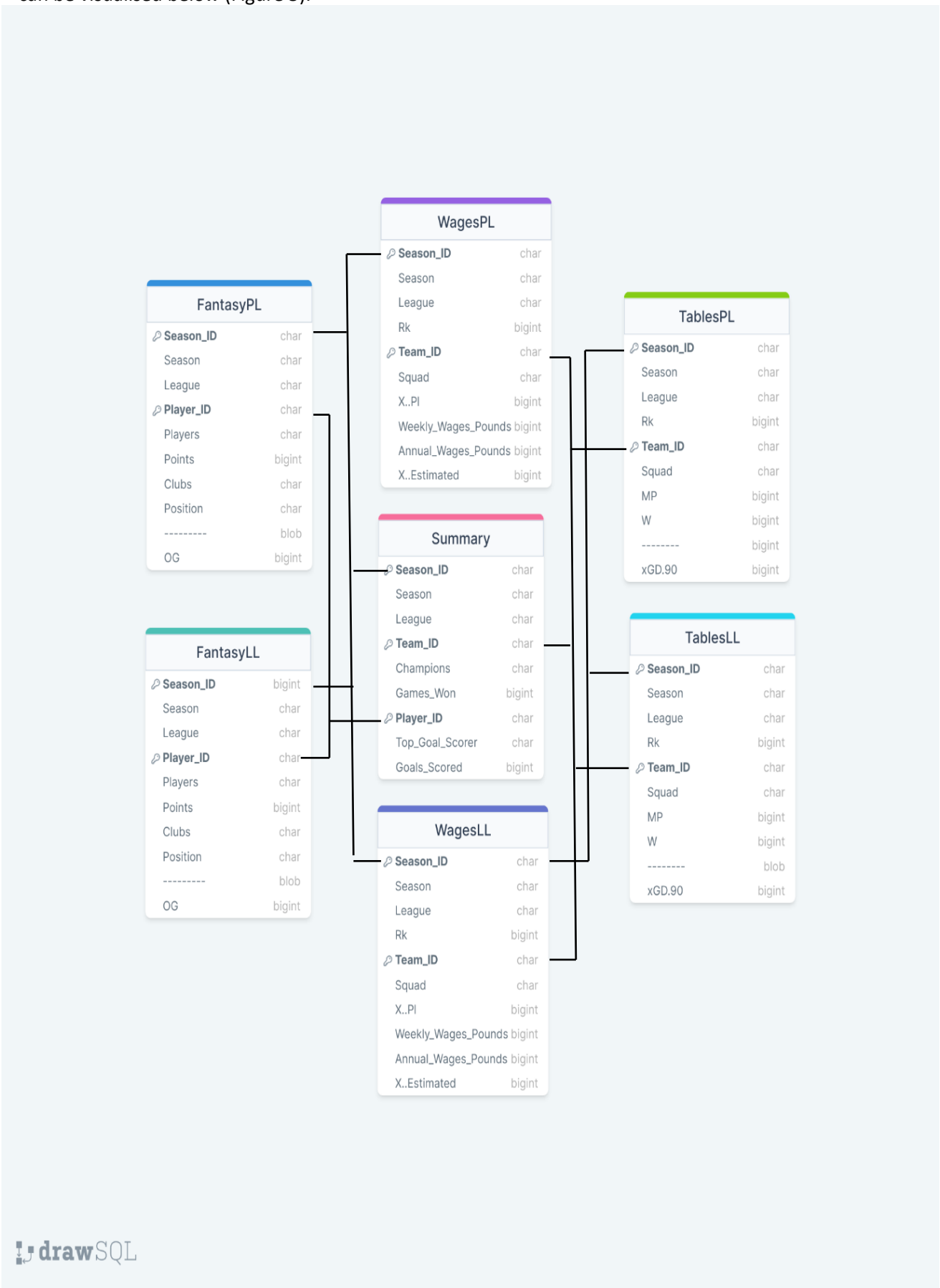


Figure 5:Diagram of the relational database generated by the web scraping processes conducted by this group

5.2 Additional Pre-processing

Although the data was relatively clean from our sources, there were a few extra steps necessary to present the data in a tidy format, ready for plotting. A few examples are given below.

5.2.1 Ordering and cleaning

When adding on the three *ID* columns, they would always be added on as the last three columns. This creates a rather messy format with the IDs located far away from their original values. Thus, each dataframe would have to be ordered correctly, to ensure the IDs preceded their original columns. This created a more readable dataset.

Additionally, a number of columns were not of the correct type. This was particularly prominent in columns that were meant to be numeric but were classed as characters. For example, the *Attendance* column in the *Tables* dataframes would contain values such as 48,500. The presence of the "," would result in this column being converted to character type. Consequently, the comma had to be removed and the resulting figure converted to numeric type. There were several other occurrences of this, but we won't delve into the details of each one.

5.2.2 Wages

One key variable we wished to investigate was the effect of wages on performance. However, the scraped data presented wage figures in a rather messy format for analyses (Figure 6).

Weekly.Wages	Annual.Wages
<chr>	<chr>
€ 6,766,923 (£ 5,674,122, $6,896,250)	€ 351,880,000 (£ 295,054,323, $358,605,024)
€ 6,044,038 (£ 5,067,977, $6,159,550)	€ 314,290,000 (£ 263,534,797, $320,296,616)

Figure 6:Original wage data, all currencies placed in one cell. Unusable for data analyses

Although the different currencies were useful to know, having them all represented in one cell was not ideal for analyses. We chose to only select the value in pounds (£). This required some rather complex regular expressions and the *mutate* function to create two new columns: *Weekly_Wages_pounds* and *Annual_Wages_pounds* (Figure 7).

Weekly_Wages_pounds	Annual_Wages_pounds
<dbl>	<dbl>
5674122	295054323
5067977	263534797

Figure 7:Cleaned wage data available for analyses

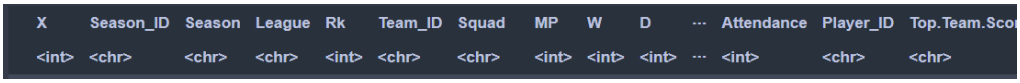
5.3 Data Wrangling with R

The primary aim of the data wrangling process in R was to prepare datasets suitable for in-depth analysis and the extraction of meaningful insights. These datasets were characterised by numerous variables and values, requiring an approach to define a specific target for narrowing down the focus of the analysis. Additionally, the wrangling process aimed to structure the data in a way that facilitates the creation of informative visualisations through plotting.

Several well-known R libraries played a crucial role in organizing the datasets for visualisation, including "tidyverse," "Scales," "hrbrthemes," "GGally," "viridis," and "ggthemes." Key functions such as "filter," "select," "mutate," and "ggplot" were employed during the wrangling phase to manipulate and shape the data effectively.

To enhance the clarity and relevance of the visualisations, the datasets underwent a filtration process, selectively retaining only the columns essential for a comprehensive and insightful visual analysis. This approach ensures that the visualised plots align with the research objectives.

The data related to points and wages in LaLiga, and the Premier League was organised. This systematic process ensures that the data is well-structured for meaningful analysis and visualisation. The resulting datasets are now ready to offer a clear picture of how points and wages unfold in these football leagues.



X	Season_ID	Season	League	Rk	Team_ID	Squad	MP	W	D	···	Attendance	Player_ID	Top.Team.Sco
<int>	<chr>	<chr>	<chr>	<int>	<chr>	<chr>	<int>	<int>	<int>	···	<int>	<chr>	<chr>

Figure 8:Columns from the tables describe standings as per seasons prior to wrangling

Season	Squad	W	D	L	Pts
<chr>	<chr>	<int>	<int>	<int>	<int>

Figure 9:Columns from the tables describe standings as per seasons after wrangling using “select” and “filter” functions

A similar method was adopted for filtering out the columns of the wages table. After organising the datasets, the next step involved creating graphs to visually represent the data. Considerable research was dedicated to choosing the most suitable graph types, and additional time was invested in enhancing the visual appeal of the plots to ensure clarity and better comprehension.

5.3.1 Graphical Examples of Wrangling:

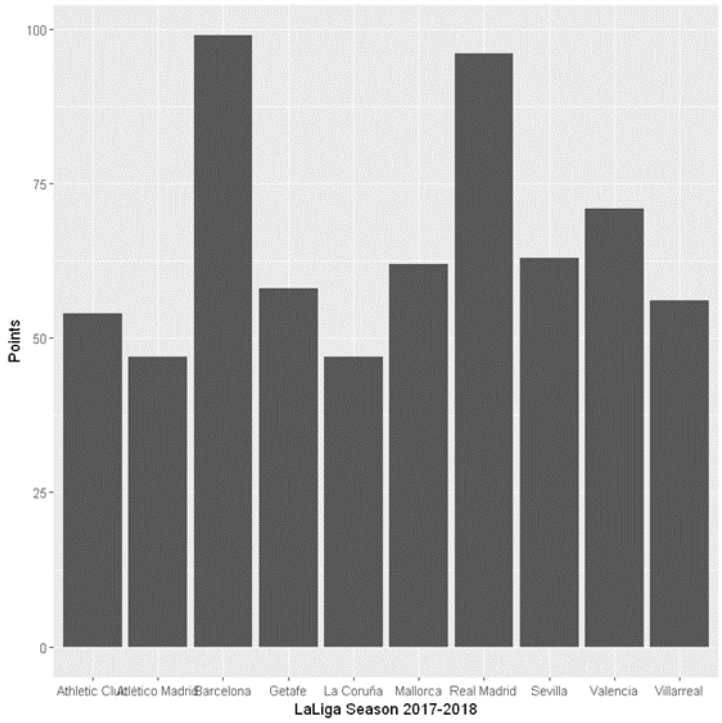


Figure 11: Top 10 teams in La Liga for the season 2017-2018

The above graph illustrates the performance of the leading 10 teams in the Premier League during the 2017-2018 season. Yet, it suffers from a lack of clarity and informational depth. To address this, enhancements were made to the graph using various functions within the "ggplot" package, an integral part of the "tidyverse" package.

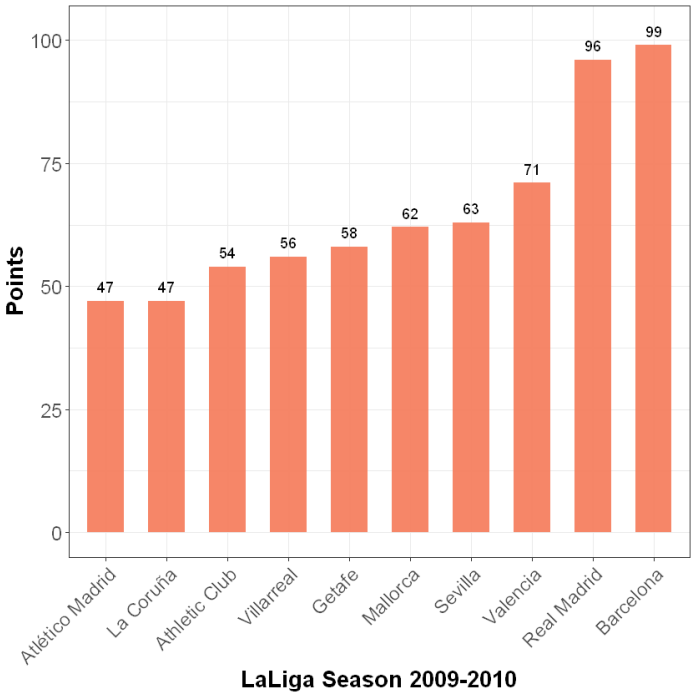


Figure 12:Top 10 teams in La Liga for the season 2017-2018

5.3.2 Fantasy Tables

5.3.2.1 Premier League Fantasy Table

Premier League Fantasy tables contains the data pertaining to each player points of different clubs and their position, goals, assists, red cards, yellow cards etc. The data was loaded and wrangled to find the top 7 highest scoring teams with respect to the total fantasy points achieved by the players. Data was loaded to a PL_Fantasy data frame and missing values/duplicate values are checked. Rows containing missing values were omitted and the cleaned data was loaded to PL_Fantasy_new data frame. Duplicate values were not found in this dataframe. Data Aggregation was done to get the total fantasy points with respect to the season and clubs.

Season	Clubs	Total_Points
<chr>	<chr>	<dbl>
1992/1993	ARSL	1445
1992/1993	AVILLA	1352
1992/1993	BRFC	1482
1992/1993	CCFC	1346
1992/1993	CHEL	1340
1992/1993	EVER	1340
1992/1993	ITFC	1274

Table 1: Aggregate Fantasy Data in Premier League

Aggregated_data was arranged to get the descending order of total points and the highest scores of each season was sliced to top_champions data frame and for better comparison data was filtered from Season 2015 to 2022.top_champions_2015 contains the data from 2015-2022 with columns season, clubs and total_points which is in fact needed for plotting.

A grouped_df: 7 × 3

Season	Clubs	Total_Points
<chr>	<chr>	<dbl>
2015/2016	ARSL	1671
2016/2017	SPURS	1848
2017/2018	MCFC	1946
2018/2019	LPOOL	1958
2019/2020	MCFC	1799
2020/2021	MCFC	1723
2021/2022	LPOOL	1998

Table 2: Sorted Aggregate Date in Premier League

To get the points scored by top scorers with respect to position of players, PL_Fantasy_new data frame is filtered with the top scoring clubs and season. The new data frame filtered_data contains

points, position, season, players, clubs .Further data is aggregated to get the total points per position and clubs. This gives us an idea about the positions of players and how it contributed to the overall play.

A tibble: 144 × 5

Points	Position	Season	Players	Clubs
<dbl>	<chr>	<chr>	<chr>	<chr>
216	Forward	2021/2022	Mohamed Salah	LPOOL
204	Midfielder	2021/2022	Heung Min Son	SPURS
180	Forward	2021/2022	Harry Kane	SPURS
170	Defender	2021/2022	Trent Alexander-Arnold	LPOOL
159	Defender	2021/2022	Virgil van Dijk	LPOOL
158	Defender	2021/2022	Joao Cancelo	MCFC
153	Defender	2021/2022	Andrew Robertson	LPOOL

Table 3: Player Fantasy Points in Premier League

A data frame was made from PL_Fantasy_new by aggregating the total_points and goals along with the players, clubs, and season columns. Objective is to find the highest scoring players over the years 2015-2022.

A grouped_df: 7 × 5

Players	Clubs	Season	Total_Points	Total_Goals
<chr>	<chr>	<chr>	<dbl>	<dbl>
Mohamed Salah	LPOOL	2017/2018	257	32
Harry Kane	SPURS	2020/2021	226	23
Harry Kane	SPURS	2016/2017	220	29
Mohamed Salah	LPOOL	2021/2022	216	23
Mohamed Salah	LPOOL	2018/2019	208	22
Jamie Vardy	LEICSC	2015/2016	202	24
Jamie Vardy	LEICSC	2019/2020	195	23

Table 4: Players and Goals in Premier League

5.3.2.2 LaLiga Fantasy Table

The La Liga data was loaded to laliga_fantasy data frame and it appears to contain information related to La Liga matches such as season information, player details, club details, statistics about player performance (e.g., fantasy points, goals, assists, cards), and other related data. For processing the data, is.na() function is used to create a logical matrix of missing values, and colSums() then calculates the sum of TRUE values (missing) in each column. The result is stored in the missing_count variable, providing a count of missing values in each column. A new dataset named laliga_Fantasy_new is created by removing rows with missing values (NA) from the original laliga_Fantasy dataset. It effectively eliminates rows with any missing values. It uses dplyr to group the data by "Season" and "Clubs," calculates the total points for each group, and displays the summarised data for further analysis or inspection.

A grouped_df: 225 × 3

Season	Clubs	Total_Points
<chr>	<chr>	<dbl>
2007/2008	ATBI	1116
2007/2008	BARC	1392
2007/2008	BETI	1098
2007/2008	MADR	1477
2007/2008	MALL	1207
2007/2008	OSAS	1105
2007/2008	RASA	1283
2007/2008	RCDC	1274

Table 5: Aggregate Fantasy Data in LaLiga

The code arranges the data to find the top highest scorers for each year based on total points. It first sorts the data by year and points, and then selects the top scorers for each year and filtered to get the top 7 highest scoring clubs (2015-2022).

A grouped_df: 7 × 3

Season	Clubs	Total_Points
<chr>	<chr>	<dbl>
2015/2016	MADR	1925
2016/2017	BARC	1865
2017/2018	BARC	1850
2018/2019	BARC	1763
2019/2020	BARC	1759
2020/2021	BARC	1718
2021/2022	MADR	1711

Table 6: Sorted Aggregate Date in LaLiga

Using La Liga tables also, highest scorers with respect to position is found out. The dataset is filtered to include only data related to the clubs "BARC" (Barcelona) and "MADR" (Real Madrid) in a particular season. It then selects specific columns, including "Points," "Position," "Season," "Players," and "Clubs." Data is aggregated based on "Position," "Clubs," and "Season," calculating the total points for each combination of these variables. This is useful for analysing the total points scored by players in different positions within specific clubs and seasons.

A grouped_df: 8 × 4

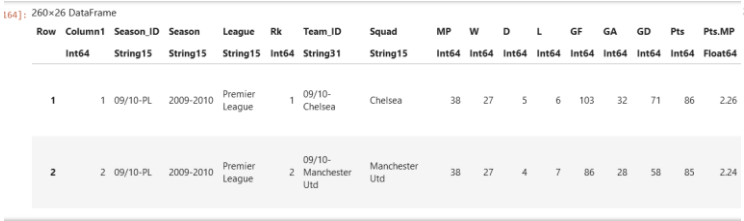
Position	Clubs	Season	Total_Points
<chr>	<chr>	<chr>	<dbl>
Defender	BARC	2021/2022	517
Defender	MADR	2021/2022	425
Forward	BARC	2021/2022	334
Forward	MADR	2021/2022	529
Goalkeeper	BARC	2021/2022	111
Goalkeeper	MADR	2021/2022	140
Midfielder	BARC	2021/2022	597
Midfielder	MADR	2021/2022	617

Table 7: Aggregated Points Based on the Position

5.4 Data Wrangling with Julia

The fundamental objective of wrangling with Julia is to discover the relationship between the wages incurred by the football teams and the points obtained by each football team. After scraping the data, there were six data frames containing various information from the world's two most famous football leagues, LaLiga and Premier League. Since there are more than 40 football teams in both these leagues, analysing all would be time consuming and pointless. So, as a group, it was decided to analyse only seven teams from both leagues that were able to gather maximum league points from 2015 to 2022.

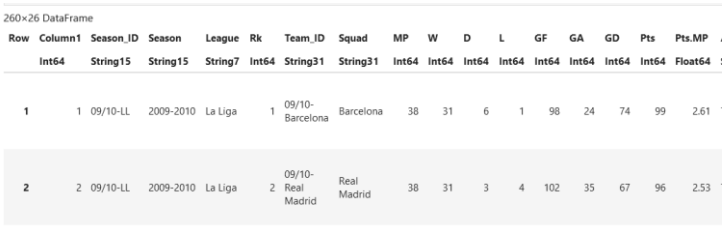
So, the first step in the wrangling was to create a data frame with teams from each league and total points from 2015 to 2022. Innerjoin, combine, groupby, dropmissing, and sort functions were used to obtain the top seven teams in each league. The data frames are listed below.



164]: 260×26 DataFrame

Row	Column1	Season_ID	Season	League	Rk	Team_ID	Squad	MP	W	D	L	GF	GA	GD	Pts	Pts.MP
	Int64	String15	String15	String15	Int64	String31	String15	Int64	Int64	Int64	Int64	Int64	Int64	Int64	Int64	Float64
1	1	09/10-PL	2009-2010	Premier League	1	09/10-Chelsea	Chelsea	38	27	5	6	103	32	71	86	2.26
2	2	09/10-PL	2009-2010	Premier League	2	09/10-Manchester Utd	Manchester Utd	38	27	4	7	86	28	58	85	2.24

Figure 13: LaLiga Points DataFrame (2015-2022)



227]: 260×26 DataFrame

Row	Column1	Season_ID	Season	League	Rk	Team_ID	Squad	MP	W	D	L	GF	GA	GD	Pts	Pts.MP
	Int64	String15	String15	String7	Int64	String31	String31	Int64	Int64	Int64	Int64	Int64	Int64	Int64	Int64	Float64
1	1	09/10-LL	2009-2010	La Liga	1	09/10-Barcelona	Barcelona	38	31	6	1	98	24	74	99	2.61
2	2	09/10-LL	2009-2010	La Liga	2	09/10-Real Madrid	Real Madrid	38	31	3	4	102	35	67	96	2.53

Figure 14:Premier League Points DataFrame (2015-2022)

```
[166]: #Filtering Desired Columns
League_Data_Goals = League_Data[:,[:Season,:Squad,:GF,:GA,:Pts,:Squad_1,:GF_1,:GA_1,:Pts_1 ]]
```

[166]: 260×9 DataFrame

Row	Season	Squad	GF	GA	Pts	Squad_1	GF_1	GA_1	Pts_1
	String15	String31	Int64	Int64	Int64	String15	Int64	Int64	Int64
1	2009-2010	Barcelona	98	24	99	Chelsea	103	32	86
2	2009-2010	Real Madrid	102	35	96	Manchester Utd	86	28	85
3	2009-2010	Valencia	59	40	71	Arsenal	83	41	75
4	2009-2010	Sevilla	65	49	63	Tottenham	67	41	70
5	2009-2010	Mallorca	59	44	62	Manchester City	73	45	67
6	2009-2010	Getafe	58	48	58	Aston Villa	52	39	64
7	2009-2010	Villarreal	58	57	56	Liverpool	61	35	63
8	2009-2010	Athletic Club	50	53	54	Everton	60	49	61
9	2009-2010	Atlético Madrid	57	61	47	Birmingham City	38	47	50
10	2009-2010	La Coruña	35	49	47	Blackburn	41	55	50
11	2009-2010	Espanyol	29	46	44	Stoke City	34	48	47
12	2009-2010	Osasuna	37	46	43	Fulham	39	46	46
13	2009-2010	Almería	43	55	42	Sunderland	48	56	44
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
249	2021-2022	Valencia	48	53	48	Brighton	42	44	51

Figure 15:Combined Points DataFrame

```
Laliga_Points_sort_Top7 = Laliga_Points_sort[1:7,:]
```

7×4 DataFrame

Row	Squad	GF_sum	GA_sum	Pts_sum
	String31	Int64	Int64	Int64
1	Barcelona	656	245	595
2	Real Madrid	590	249	584
3	Atlético Madrid	429	191	548
4	Sevilla	391	301	458
5	Villarreal	392	300	413
6	Real Sociedad	370	329	391
7	Athletic Club	323	298	373

Figure 17:LaLiga Top07 Teams

```
PL_Points_sort_Top7 = PL_Points_sort[1:7,:]
```

7×4 DataFrame

Row	Squad_1	GF_1_sum	GA_1_sum	Pts_1_sum
	String15	Int64	Int64	Int64
1	Manchester City	636	223	602
2	Liverpool	561	253	568
3	Tottenham	494	268	496
4	Chelsea	472	286	492
5	Manchester Utd	432	283	480
6	Arsenal	461	317	465
7	Leicester City	420	357	404

Figure 16: Premier League Top07 Teams

The following step was to determine the relationship between wages, fantasy points, and league points. The primary goal of this study is to determine how much each club has spent to get one fantasy point and one league point. As a result, a relational data frame was required to connect all the information, and by using the "Club" name as the key, two relational data frames for two distinct leagues were created.

5.4.1 LaLiga

```
LL_Fantasy_Points_2015_2022_Top_07_Final
```

7×2 DataFrame

Row	Clubs	Total_Fantasy_Points
	String	Int64
1	Barcelona	14298
2	Real Madrid	13490
3	Sevilla	11481
4	Villarreal	11371
5	Real Sociedad	11153
6	Athletic Club	10554
7	Atlético Madrid	10312

Figure 18:LaLiga Total Fantasy Points

```
LaLigaWages_2015_2022_Top_07_Final
```

7×2 DataFrame

Row	Clubs	Total_Wages
	String31	Int64
1	Barcelona	1563668007
2	Real Madrid	1494577445
3	Atlético Madrid	799884995
4	Sevilla	426414202
5	Villarreal	329384533
6	Athletic Club	288967595
7	Real Sociedad	220558729

Figure 20: LaLiga Total Wages

```
Laliga_Points_Top7_Final = rename!(Laliga_Points_sort_Top7,:Squad => :Clubs,:Pts_sum =>
```

7×4 DataFrame

Row	Clubs	Total_league_Goals_For	Total_league_Goals_Against	Total_league_Points
	String31	Int64	Int64	Int64
1	Barcelona	656	245	595
2	Real Madrid	590	249	584
3	Atlético Madrid	429	191	548
4	Sevilla	391	301	458
5	Villarreal	392	300	413
6	Real Sociedad	370	329	391
7	Athletic Club	323	298	373

Figure 19:LaLiga Total League Points

```
Laliga_Final = innerjoin(Laliga_Points_Top7_Final, LL_Fantasy_Points_2015_2022_Top_07_Final ,LaLigaWages_2015_2022_Top_07_Final
```

7×6 DataFrame

Row	Clubs	Total_league_Goals_For	Total_league_Goals_Against	Total_league_Points	Total_Fantasy_Points	Total_Wages
	String31	Int64	Int64	Int64	Int64	Int64
1	Barcelona	656	245	595	14298	1563668007
2	Real Madrid	590	249	584	13490	1494577445
3	Atlético Madrid	429	191	548	10312	799884995
4	Sevilla	391	301	458	11481	426414202
5	Villarreal	392	300	413	11371	329384533
6	Athletic Club	323	298	373	10554	288967595
7	Real Sociedad	370	329	391	11153	220558729

Figure 21:LaLiga Combined DataFrame

5.4.2 Premier League

```
PL_Fantasy_Points_2015_2022_Top_07_Final
```

7×2 DataFrame

Row	Clubs	Total_Fantasy_Points
	String	Int64
1	Manchester City	14249
2	Liverpool	13482
3	Tottenham	12518
4	Arsenal	12359
5	Chelsea	12248
6	Manchester Utd	12068
7	Leicester City	10901

Figure 22: Premier League Total Fantasy Points

```
PremierLeagueWages_2015_2022_Top07_Final
```

7×2 DataFrame

Row	Clubs	Total_Wages
	String15	Int64
1	Manchester Utd	1197277000
2	Manchester City	1005246000
3	Chelsea	921073988
4	Arsenal	875531000
5	Liverpool	799978000
6	Tottenham	649212000
7	Leicester City	470012000

Figure 22: Premier League Total Wages

```
PL_Points_Top7_Final = rename!(PL_Points_sort_Top7,:Squad_1 => :Clubs,:Pts_1_sum => :Tot
```

7×4 DataFrame

Row	Clubs	Total_league_Goals_For	Total_league_Goals_Against	Total_league_Points
	String15	Int64	Int64	Int64
1	Manchester City	636	223	602
2	Liverpool	561	253	568
3	Tottenham	494	268	496
4	Chelsea	472	286	492
5	Manchester Utd	432	283	480
6	Arsenal	461	317	465
7	Leicester City	420	357	404

Figure 22:Premier League Total League Points

```
PremierLeague_Final = innerjoin(PL_Points_Top7_Final, PL_Fantasy_Points_2015_2022_Top_07_Final  ,PremierLeagueWages_2015
```

7×6 DataFrame

Row	Clubs	Total_league_Goals_For	Total_league_Goals_Against	Total_league_Points	Total_Fantasy_Points	Total_Wages
	String15	Int64	Int64	Int64	Int64	Int64
1	Manchester Utd	432	283	480	12068	1197277000
2	Manchester City	636	223	602	14249	1005246000
3	Chelsea	472	286	492	12248	921073988
4	Arsenal	461	317	465	12359	875531000
5	Liverpool	561	253	568	13482	799978000
6	Tottenham	494	268	496	12518	649212000
7	Leicester City	420	357	404	10901	470012000

Figure 23:Premier League Combined DataFrame

The pounds per fantasy point and pounds per league point were then calculated using two modified columns.

```
Laliga_Final
```

7×8 DataFrame

Row	Clubs	Total_league_Goals_For	Total_league_Goals_Against	Total_league_Points	Total_Fantasy_Points	Total_Wages	Pounds_Per_Fantacy_Point	Pounds_Per_League_Point
	String31	Int64	Int64	Int64	Int64	Int64	Float64	Float64
1	Barcelona	656	245	595	14298	1563668007	1.09363e5	2.62801e6
2	Real Madrid	590	249	584	13490	1494577445	1.10792e5	2.55921e6
3	Atlético Madrid	429	191	548	10312	799884995	77568.4	1.45964e6
4	Sevilla	391	301	458	11481	426414202	37140.9	9.31035e5
5	Villarreal	392	300	413	11371	329384533	28967.1	7.97541e5
6	Athletic Club	323	298	373	10554	288967595	27379.9	7.74712e5
7	Real Sociedad	370	329	391	11153	220558729	19775.7	5.64089e5

Figure 24:Modified LaLiga DataFrame

```
PremierLeague_Final
```

7×8 DataFrame

Row	Clubs	Total_league_Goals_For	Total_league_Goals_Against	Total_league_Points	Total_Fantasy_Points	Total_Wages	Pounds_Per_Fantacy_Point	Pounds_Per_League_Point
	String15	Int64	Int64	Int64	Int64	Int64	Float64	Float64
1	Manchester Utd	432	283	480	12068	1197277000	99210.9	2.49433e6
2	Manchester City	636	223	602	14249	1005246000	70548.5	1.66984e6
3	Chelsea	472	286	492	12248	921073988	75202.0	1.8721e6
4	Arsenal	461	317	465	12359	875531000	70841.6	1.88286e6
5	Liverpool	561	253	568	13482	799978000	59336.7	1.40841e6
6	Tottenham	494	268	496	12518	649212000	51862.3	1.3089e6
7	Leicester City	420	357	404	10901	470012000	43116.4	1.1634e6

Figure 25:Modified Premier League DataFrame

6 Results and Discussion

First, we will look at graphs and find the relationship between Season winners and Wages paid by top clubs in both leagues for the period of 2015 to 2022.

6.1 Season winners and wages paid in LaLiga.

Figure 26:LaLiga Season winners between 2015 to 2022

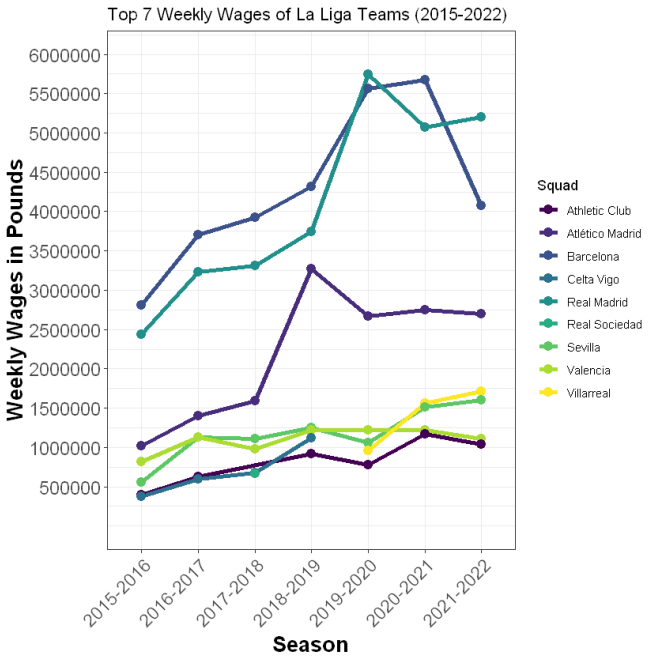


Figure 27:Top 7 weekly wages of La Liga teams between 2015 to 2022

From figure 27 we can infer that based on the observations; it's noted that the winning teams in La Liga have earned points ranging from 86 to 93 in the past seven seasons. Although La Liga consists of 20 teams, the teams listed above, such as Barcelona, Real Madrid, and Atlético Madrid, tend to be the frequent winners.

According to figure 28, it is evident that Barcelona and Real Madrid have consistently surpassed other LaLiga clubs in terms of weekly wages paid to their players, with Atletico Madrid consistently ranking at number 3.

The observation suggests a positive correlation between the wages paid by clubs and the seasons they win in LaLiga. However, the 20/21 season stands out as an exception, as Atletico Madrid managed to clinch victory despite Barcelona and Real Madrid paying higher wages to their players.

6.2 Season winners and wages paid in Premier League.

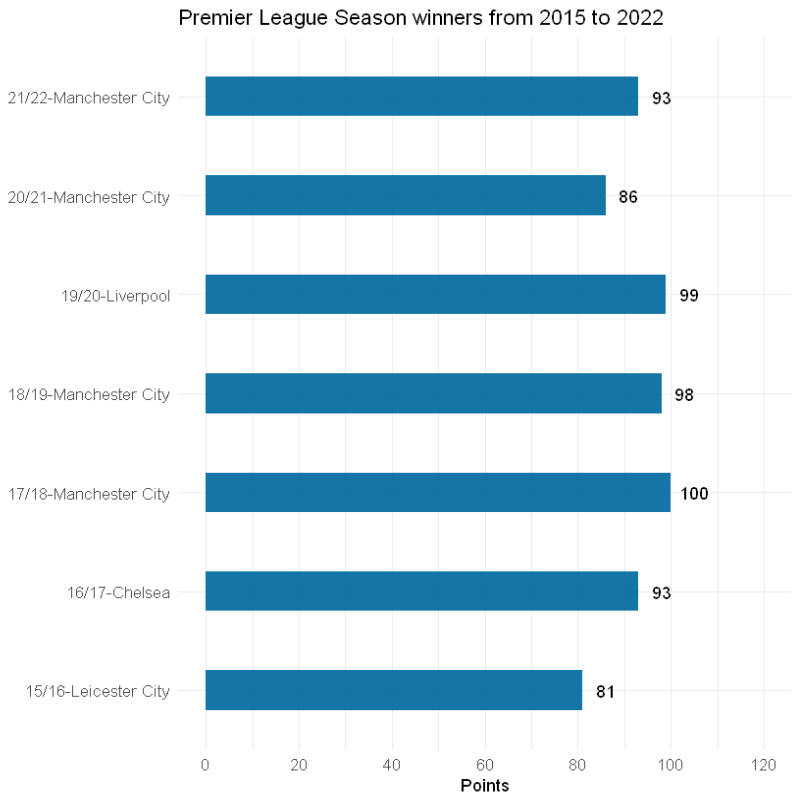


Figure 29: Premier League Season winners between 2015 to 2022

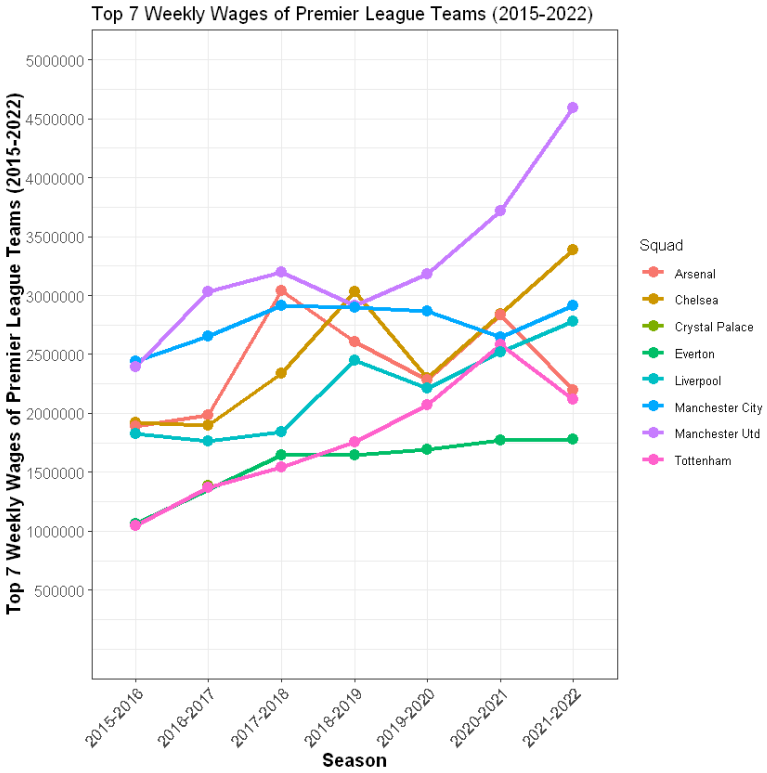


Figure 30: Premier League Season winners between 2015 to 2022

Manchester City secured the title in the seasons 2017-2018, 2018-2019, 2020-2021, and 2021-2022, consistently achieving point totals exceeding 90. Notably, Premier League winners have often achieved higher point totals, with one season's champion even reaching the remarkable milestone of 100 points. Liverpool, Chelsea, and Leicester City have each claimed a title, indicating a broader range of competition in the Premier League for the coveted top spot.

The data presented in figure 30 provides interesting insights into the dynamics of football clubs, particularly in the context of their wage expenditure and championship victories. Despite being the highest spender on wages, Manchester United's performance in terms of winning championships has not reflected this financial investment. On the other hand, Leicester City, while not ranking among the top 7 clubs in wage expenditure, has managed to secure championships. This contrast highlights the complex and sometimes unpredictable nature of success in football, where financial resources do not always directly correlate with on-field achievements.

6.3 Investigating the correlation between wages and wins

To examine the potential impact of player wages on team performance, the correlation between these two variables was assessed. Scatterplots will be used as they are useful for identifying the strength and direction of correlation of two variables. Annual wages were set as the independent variable while number of wins is the dependent variable.

Figures 31 and 32 shows that there is a positive correlation between the annual wages and wins for both La Liga and Premier League from seasons 2015 to 2022. For La Liga, a tighter clustering of points can be observed close to the origin of the graph. This indicates that while there is a positive relationship, the correlation is not very strong. To objectively assess the strength, the correlation coefficient was found to be 0.73, which still indicates strong correlation as it is closer to 1.

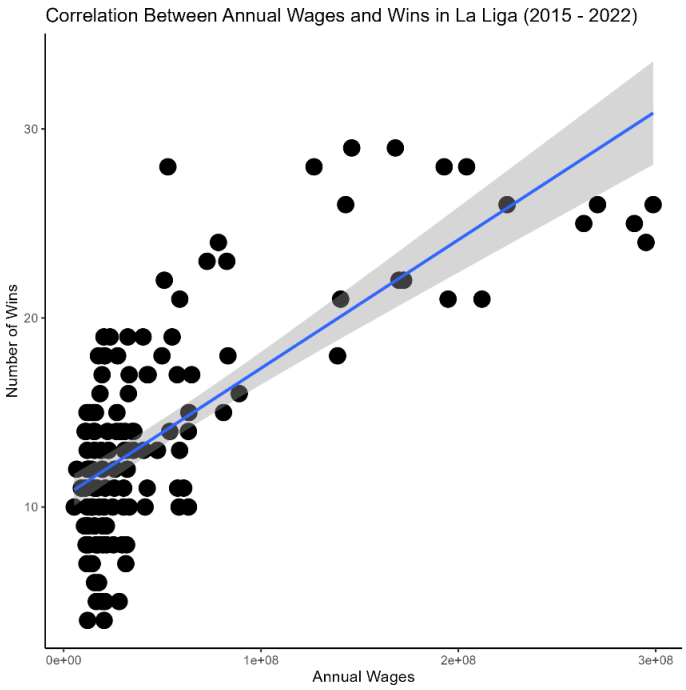


Figure 31: LaLiga player wages appear to be positively correlated with annual wages

In the Premier League, the correlation between annual wages and wins is positive, but weaker. The points in the scatterplot are less tightly clustered compared to the La Liga wages and wins correlation. This is confirmed by its correlation coefficient of 0.68, which is slightly lower than La Liga (Figure 32).

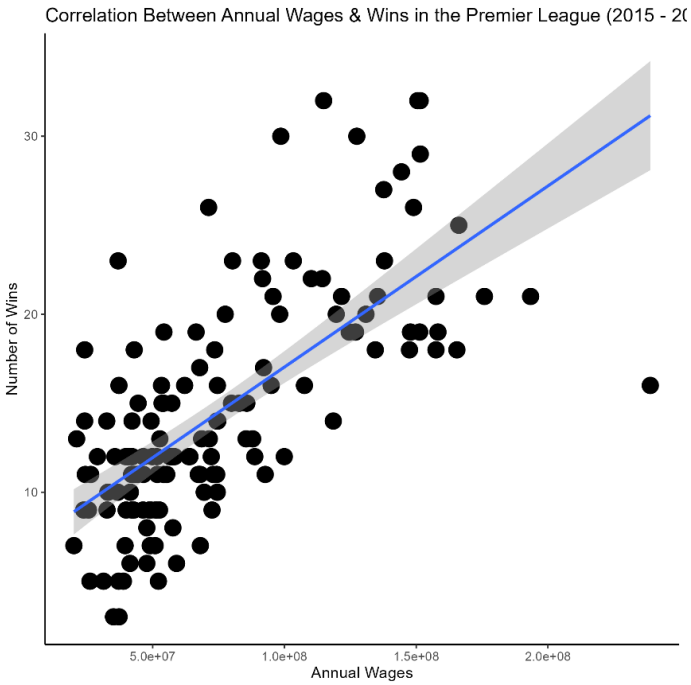


Figure 32: Premier League player wages appear to be positively correlated with annual wages

Now that the correlation between player wages and wins have been confirmed, we can know perform some further analyses to understand how player compensation impacts team performance.

6.4 Does attendance affect wins?

A positive correlation between attendance and wins suggests that as a team performs better, more fans are inclined to attend matches. In both La Liga and Premier League, a positive correlation is observed between these two variables. However, the correlation is stronger in La Liga matches. This difference may be due to a difference in marketing strategies, historical success, etc. These factors were not analysed further in this report.

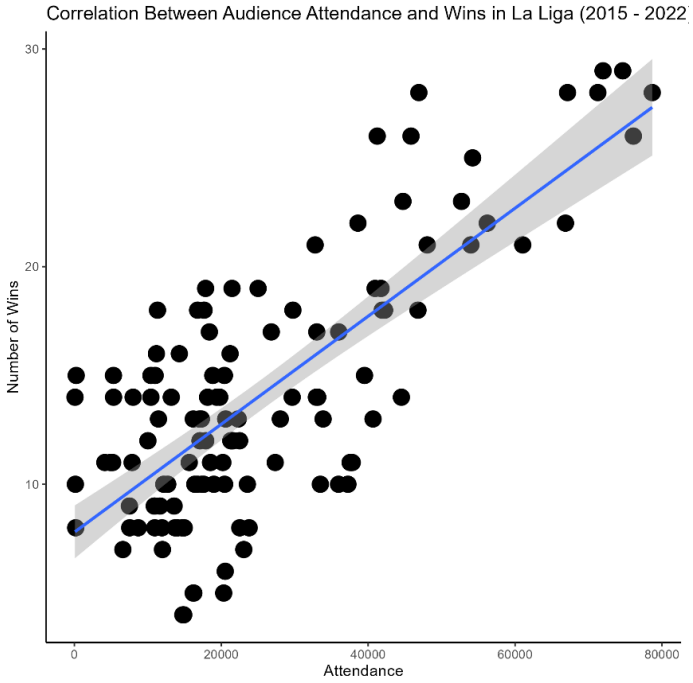


Figure 33: Correlation Between the Attendance and Wins in LaLiga.

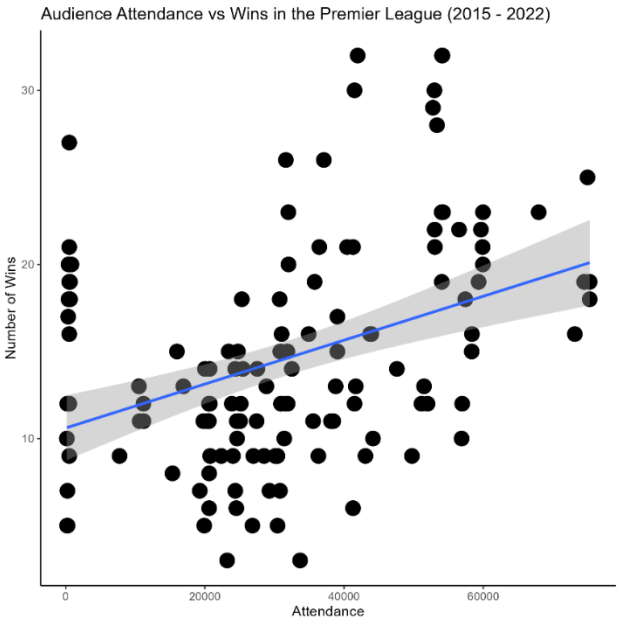


Figure 34: Correlation Between the Attendance and Wins in Premier League

6.5 Fantasy Tables

Ggplot2 library is used to visualise the wrangled data. Ggplot2 is a popular R package for creating data visualizations, particularly for creating elegant and customised graphics. It's a part of the larger "tidyverse" ecosystem in R, which emphasises a consistent and intuitive approach to data manipulation and visualisation.

6.5.1 Top seven highest scored (fantasy points) premier league teams 2015-2022

Figure 35: Accumulated Fantasy Points (Premier League)

It can be inferred that players of Manchester City (MCFC) team have scored high during various years considering the time range 2015-2022. Liverpool (LPOOL) team members are having the highest points, and this was achieved in the season 2021-2022.

6.5.2 Total fantasy points of premier league teams by player position

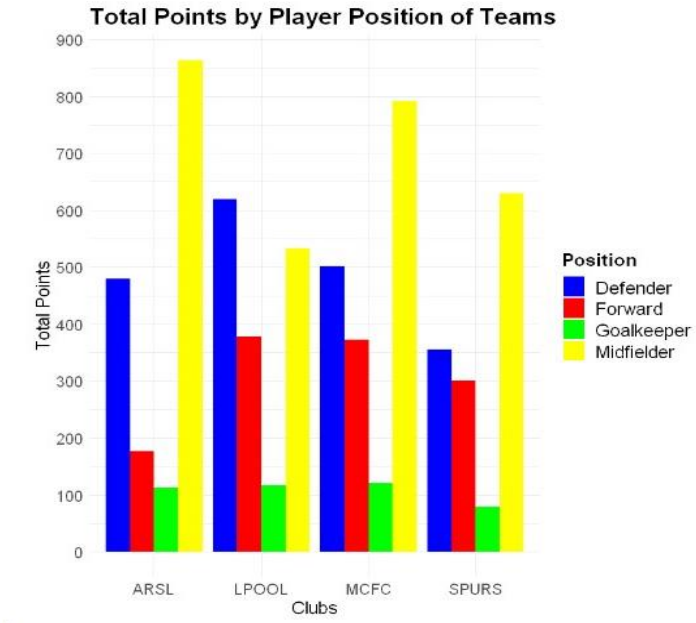


Figure 36: Fantasy Points by Position

It can be inferred that midfielder of each team has scored more goals leading to highest fantasy points (goal scored by midfielder=5 points). Defender has contributed more to the Liverpool team than the midfielder (goal scored by goalkeeper or defender=6 points). Arsenal midfielder scored the highest points, but defender and forward were not able to score more points (goal scored by forward=4 points)

6.5.3 Top seven Highest scored (Fantasy Points) LaLiga Teams 2015-2022

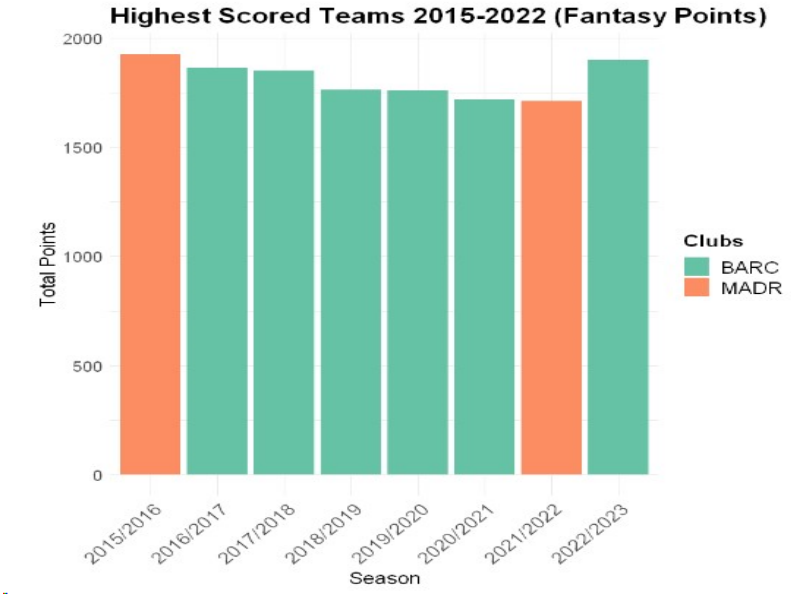


Figure 37: Fantasy Points by Year

It can be inferred that players of Real Madrid (MADR) have scored high almost all seasons' years considering the time range 2015-2022. MADR team members are having the highest points in this plot, and this was achieved in the season 2015-2016.

6.5.4 Total fantasy points of LaLiga teams by player position

It can be inferred that forwarder of each team has scored more goals leading to highest fantasy points (goal scored by midfielder=5 points). Midfielders has the second highest points, and they scored more goals than defender (goal they scored by goalkeeper or defender=6 points). Barcelona and Real Madrid is having the same trend of point distribution with respect to the position.

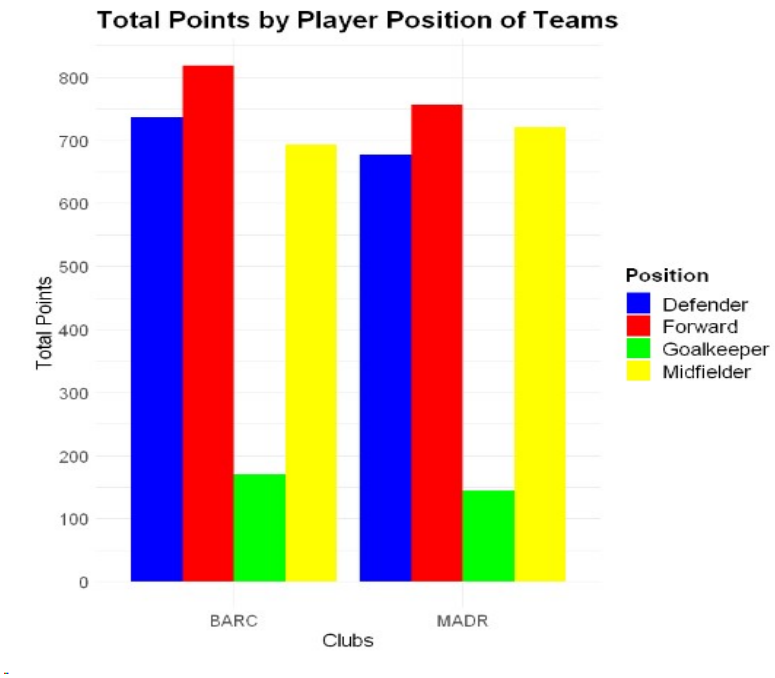


Figure 38: Total Fantasy Points in LaLiga by Player Position

6.6 Top seven teams according to the league points for the period of 2015 to 2022.

6.6.1 LaLiga

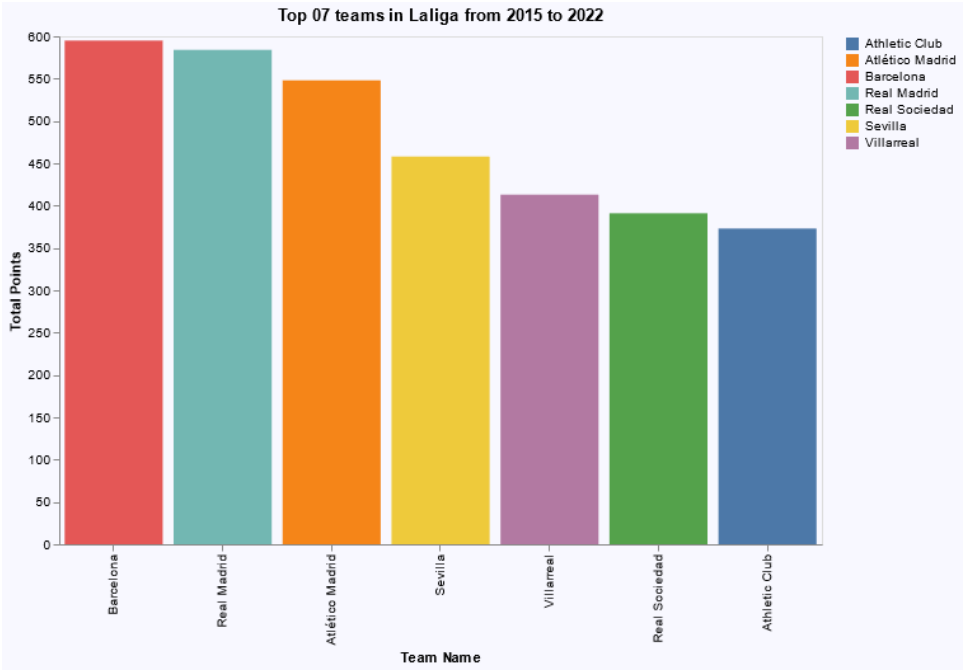


Figure 39: Top Seven LaLiga Teams

In the Spanish football league LaLiga from 2015 to 2022 (seven seasons), club Barcelona accumulated 595 points, while Real Madrid accumulated 584 points. Atletico Madrid is third on the list, and these

are the teams that have won trophies (Barcelona: 3 times, Real Madrid: 3 times, Atletico Madrid: 3 times) within the period indicated above.

6.6.2 Premier League

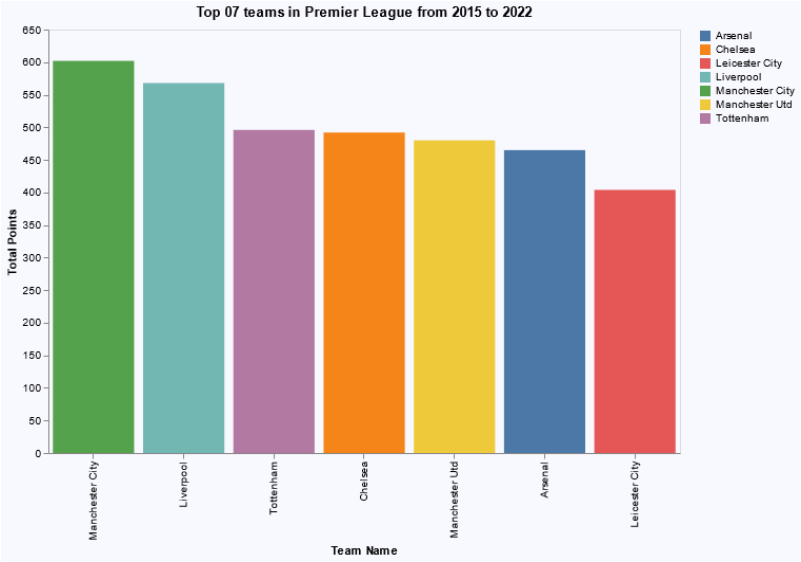


Figure 40: Top Seven Premier League Teams

Manchester City collected 602 league points while winning four titles in the English Premier League between 2015 and 2022. Team Liverpool and Tottenham are second and third on the list, however neither has won a championship within the period indicated. Chelsea won the championship in the 2016-2017 season, whereas Leicester City won it in the 2015-2016 season. However, they are ranked fourth and seventh in the cumulative point list, respectively.

6.7 Pounds Per League Point

6.7.1 LaLiga

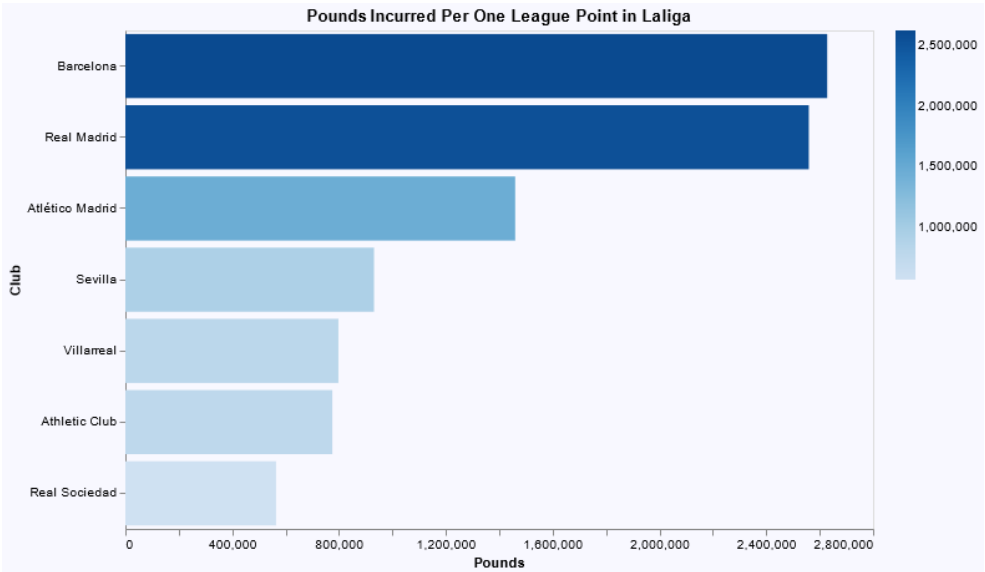


Figure 41: Pounds Per League Point - LaLiga

The above bar graph depicts how much each team spent to gain a single LaLiga league point from 2015 to 2022. As with the cumulative points graph, Barcelona and Real Madrid are at the top of the list, with Atletico Madrid coming in third. However, all the other clubs spent substantially less than the top three.

6.7.2 Premier League

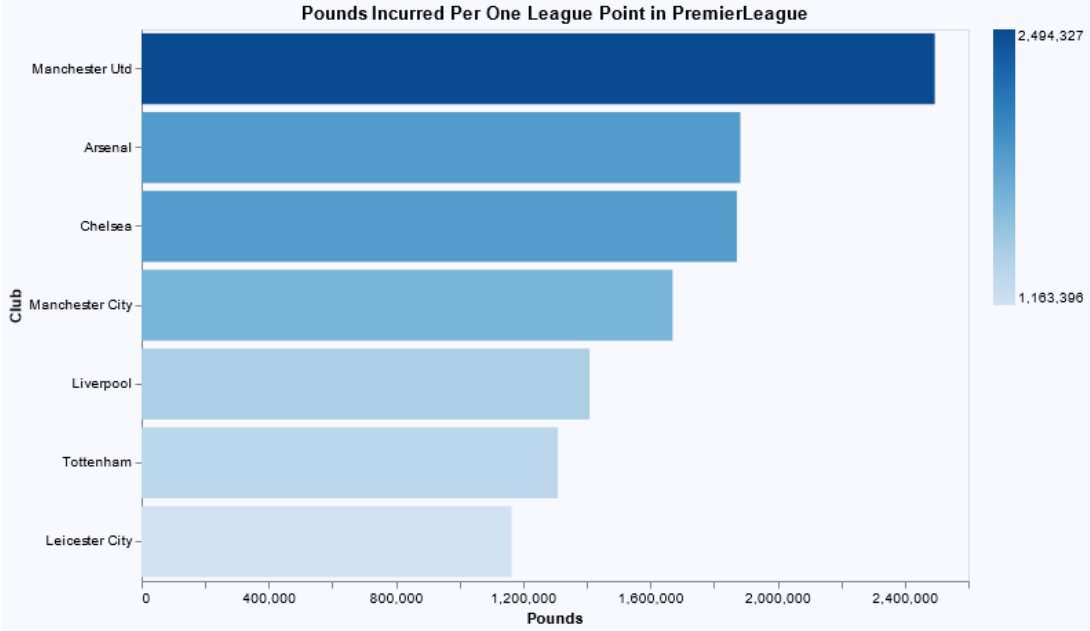


Figure 42: Pounds Per League Point - Premier League

When considering the English Premier League, Manchester United has spent more than 2.4 million pounds per league point. Arsenal and Chelsea are second and third, respectively, with four-time winner Manchester City in fourth.

6.8 Pounds Per Fantasy Point

6.8.1 LaLiga

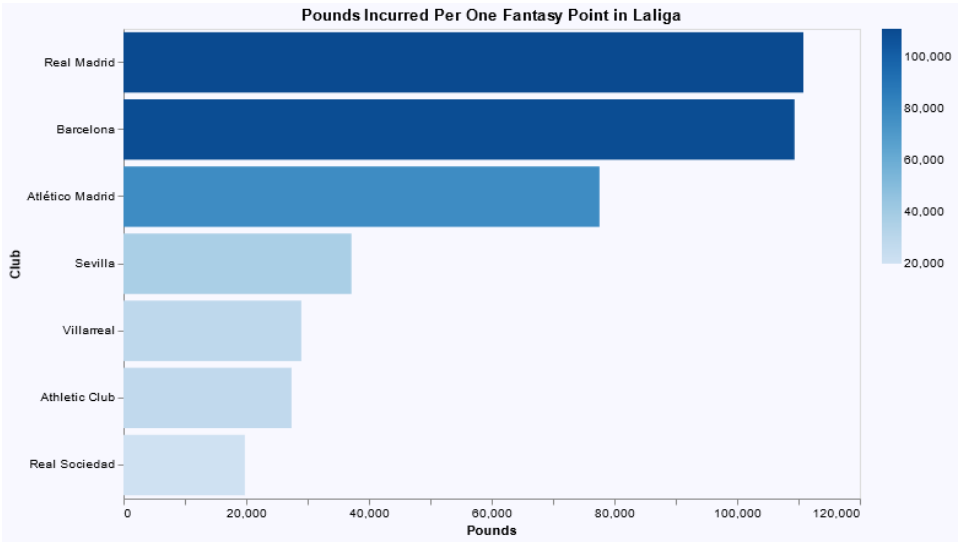


Figure 43: Pounds Per Fantasy Point - LaLiga

When it comes to fantasy points in LaLiga, Real Madrid has spent the most for one fantasy point, with Barcelona coming in second. Both teams have spent more than 100,000 pounds to get a single fantasy point. Atletico Madrid has spent around 80,000 per fantasy point, whereas the other teams have spent substantially less than these three giants.

6.8.2 Premier League

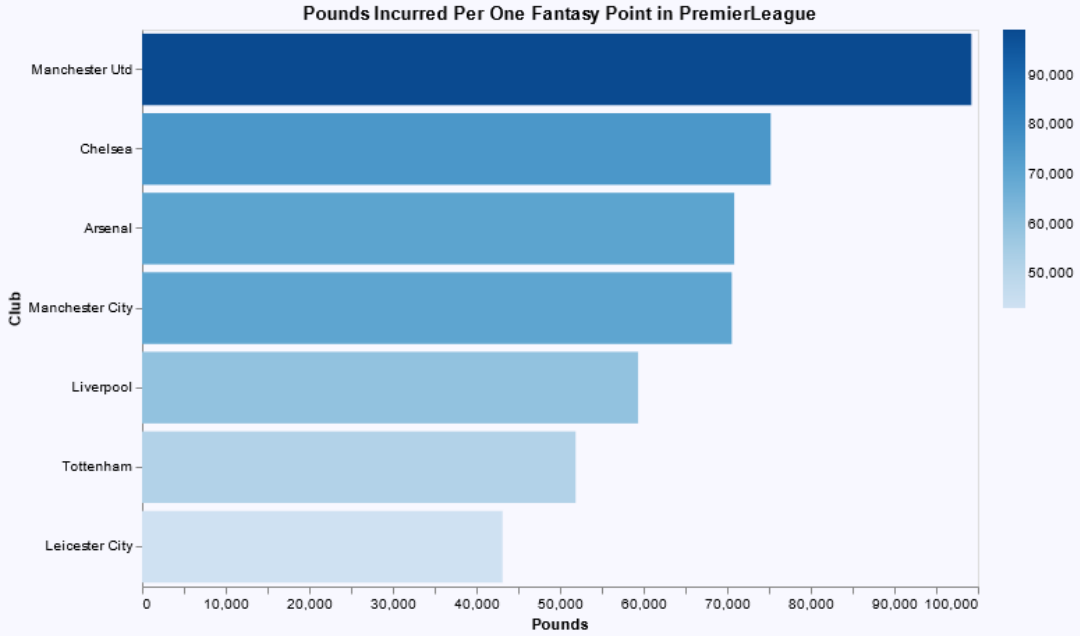


Figure 44: Pounds Per League Point - Premier League

Manchester United football club is at the top of the Premier League expenditure list for the most money to acquire a single fantasy point. Chelsea and Arsenal are in second and third place, respectively, with Manchester City in fourth place. Manchester United has spent more than 100,000 pounds per fantasy point, far more than any other team in the league.

7 Conclusion

Since football is the most popular sport in the world, the money involved in the game is enormous. So, the main idea behind this analysis was to determine how the major football teams performed over the last seven seasons and whether they obtained the projected return on investment.

When it comes to performance, the Barcelona and Real Madrid teams have dominated LaLiga, winning three titles each in the last seven years. Only Atletico Madrid has won a championship, and other teams failing to even come close. On the other hand, Manchester City football team has dominated the Premier League. They have won four championships in the last seven seasons. Aside from that, Liverpool, Chelsea, and Leicester City have each won the title once. However, when total goal count and points are considered, the Premier League is more competitive than LaLiga. The top two teams in LaLiga have a significant advantage over the other teams. But, in the Premier League, everyone has fought hard, and the margins are minimal. As a result, the Premier League is more competitive and interesting as a competition.

When comparing two leagues, it's clear that some clubs spent a lot of money to acquire the results they needed, while others got the results, they needed without spending a lot of money. Manchester United, for example, has spent a lot of money on players yet has yet to win a single championship. They are also at the top of both graphs for spending the most money to acquire league and fantasy points. Manchester City, on the other side, has won four titles in the last seven seasons despite not spending a lot of money on players. Real Madrid and Barcelona teams have spent a lot on players to acquire points and they have able to win the championship three times each within last seven seasons. So, they have got the result they want after investing a huge amount of money.

Another interesting finding is that there is a positive relationship between home ground attendance and home team wins. When there are more supporters, the team is more likely to win. This could be a physiological factor, and the cheering crowd could boost the player energy.

Although we examined the performance and the money invested, there are additional elements that influence the return on investment. One of the factors affecting investment and return is the popularity of the teams. There are some teams that are underperforming but have an enormous fan following. As a result, they have more income than the other performing teams, and they prefer to spend more on players regardless of their performance. Manchester United Football Club is a perfect example of this. Despite their average success over the last seven seasons, they have a huge global fan base. As a result, they make a lot of money from commercials, sponsorships, and other sources. This is a broad topic that requires a large quantity of data from several aspects to evaluate, which we were unable to do in this analysis due to the time constrains we have.

8 References

Gumpo. *statbunker*. Retrieved October 05, 2023, from https://www.statbunker.com/

Netlify, F. *Julia*. Retrieved October 05, 2023, from https://julialang.org/

SPORTS REFERENCE and STATHEAD trademarks. (2000, January 1). *fbref*. Retrieved 10 05, 2023, from https://fbref.com/en/#site_menu_link

tack Exchange Inc;. *stackoverflow*. Retrieved October 05, 2023, from https://stackoverflow.com/

9 Appendix

Here are some graphs that may provide additional insight into LaLiga and the Premier League.

9.1 Team standings in each league

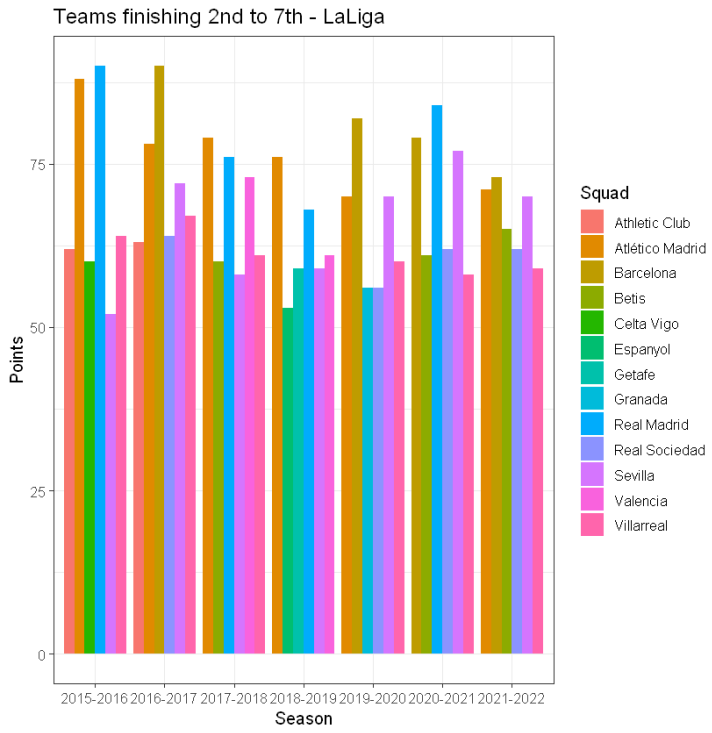


Figure 45: Team Standings in LaLiga

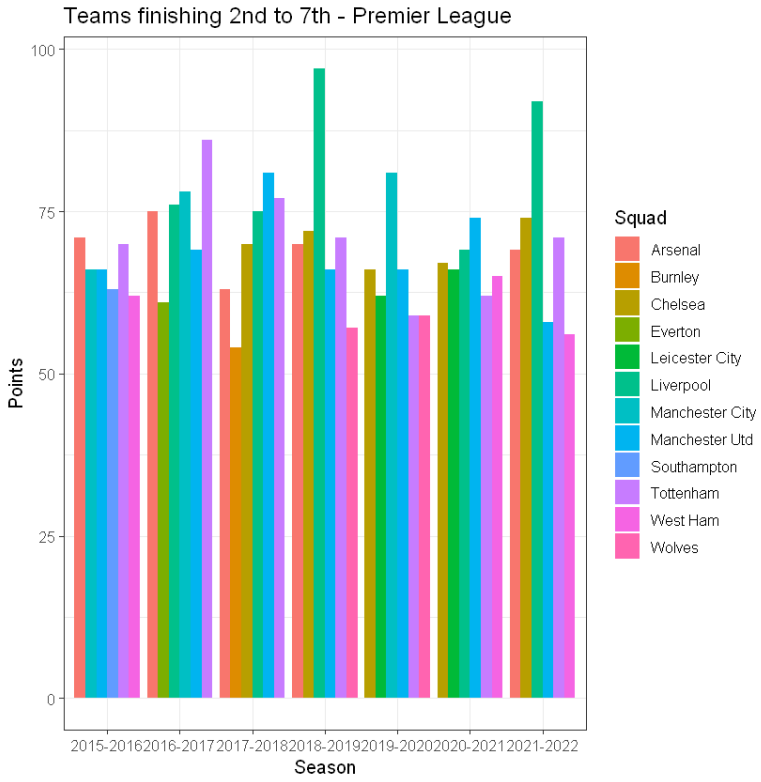


Figure 46: Team Standings in Premier League

9.2 Mean and Median goals per season

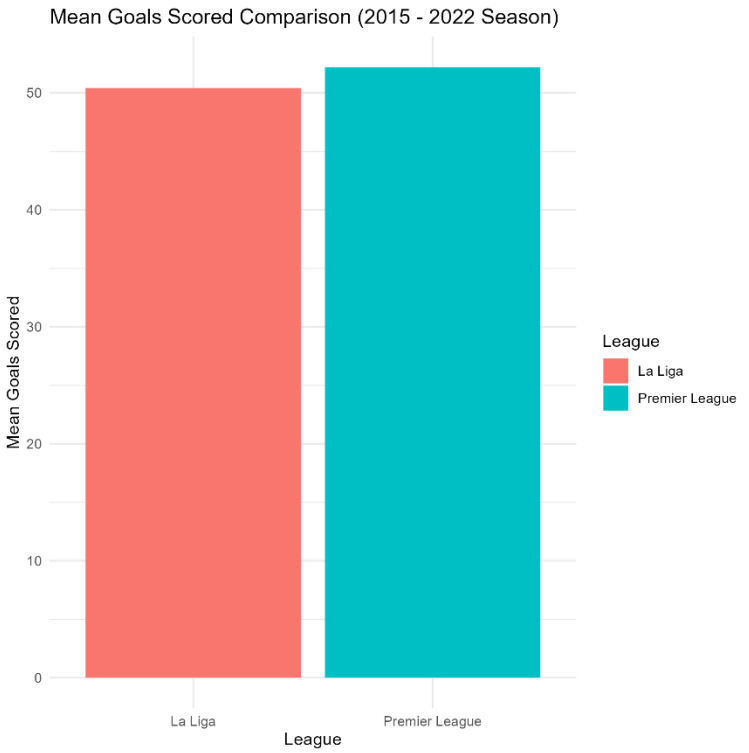


Figure 47: Mean Goals Per Season

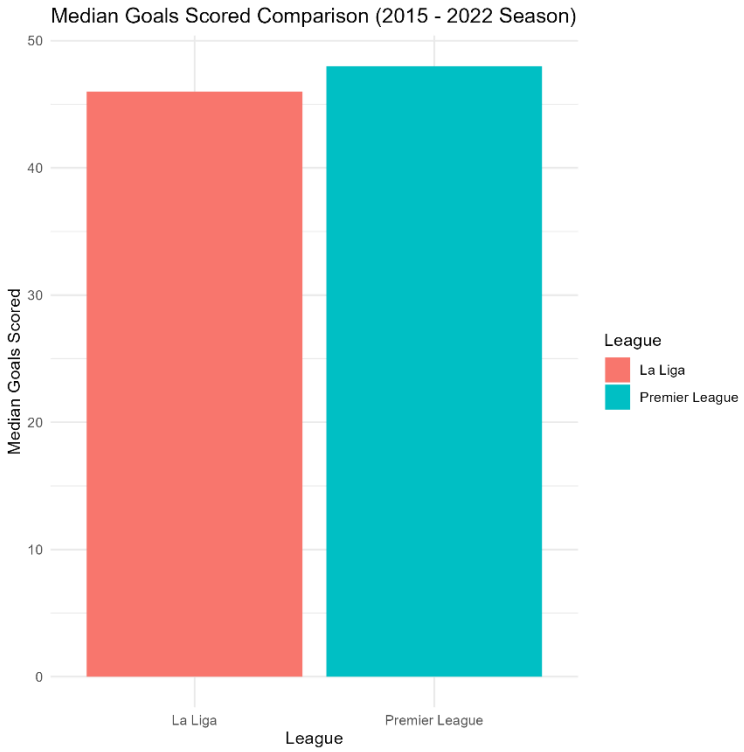


Figure 48: Median Goals Per Season