

# README

## Loading the Data:

The pre-processing of the EPL 20/21 dataset was first initiated by importing it into the Microsoft Power BI tool. There were two options for accomplishing this. Manually by selecting the file type from the "Get Data" menu and then importing it, or by creating a blank query and importing it using M-Language. In this report, the former was utilized; nonetheless, a screenshot of the M language input is included.

I also used external source to get the clubs logo for the visualizations. I will also be illustrating how I achieved this;

- Starting up Microsoft Power BI application:



Fig 28. Power BI starting up

- Click on the "Get Data" button in the ribbon at the top of the Power BI interface.

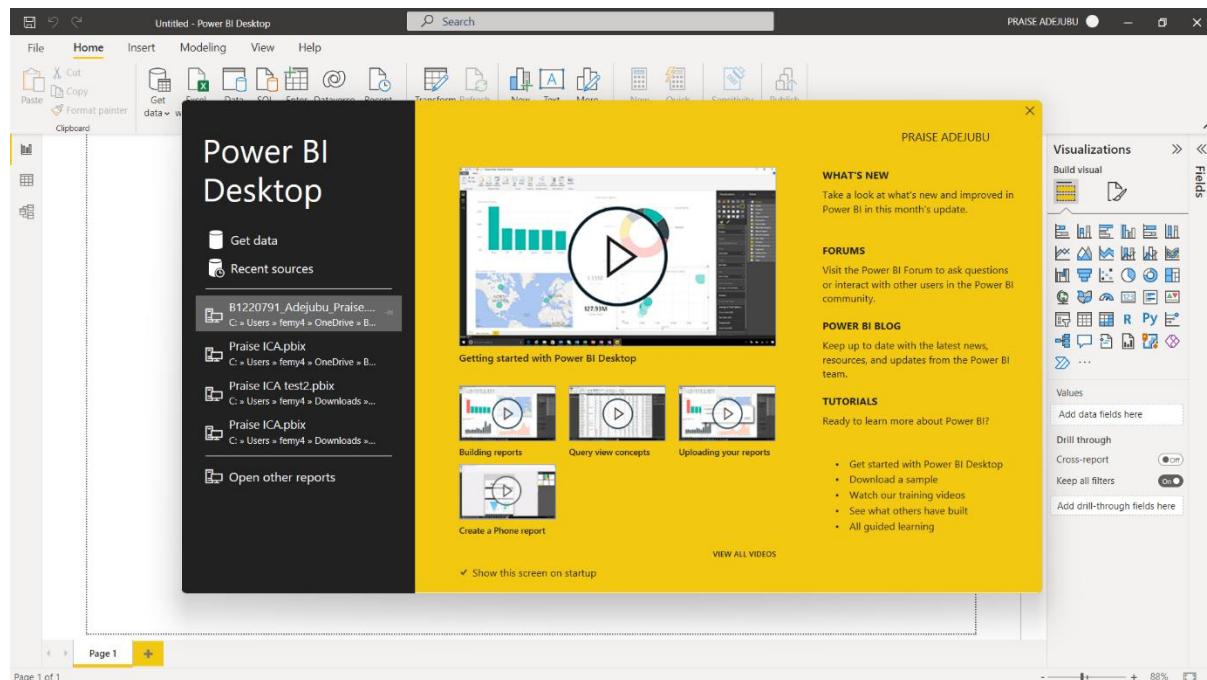


Fig 29. Power BI Start page

- In the "Get Data" window, select "File" from the list of data sources on the left side.
- From the list of file types, select "CSV" and click "Connect."

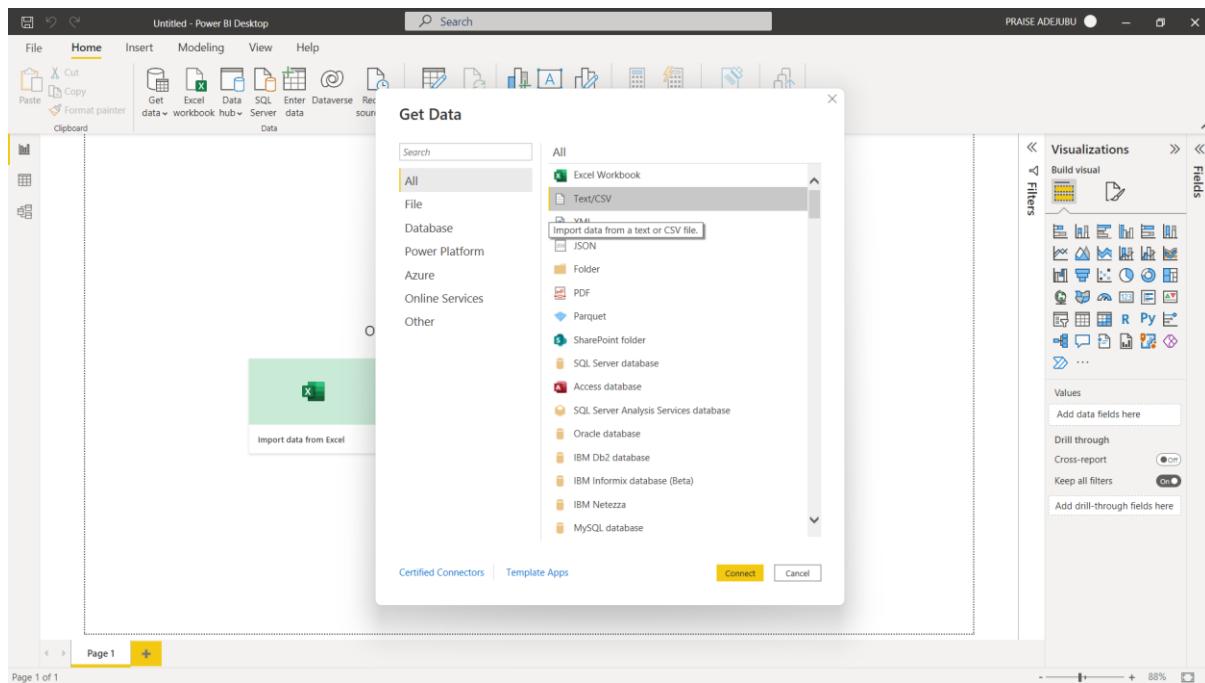


Fig 30. Power BI modelling process

- In the "Import Data" window, navigate to the location of the CSV file on your computer and select it.

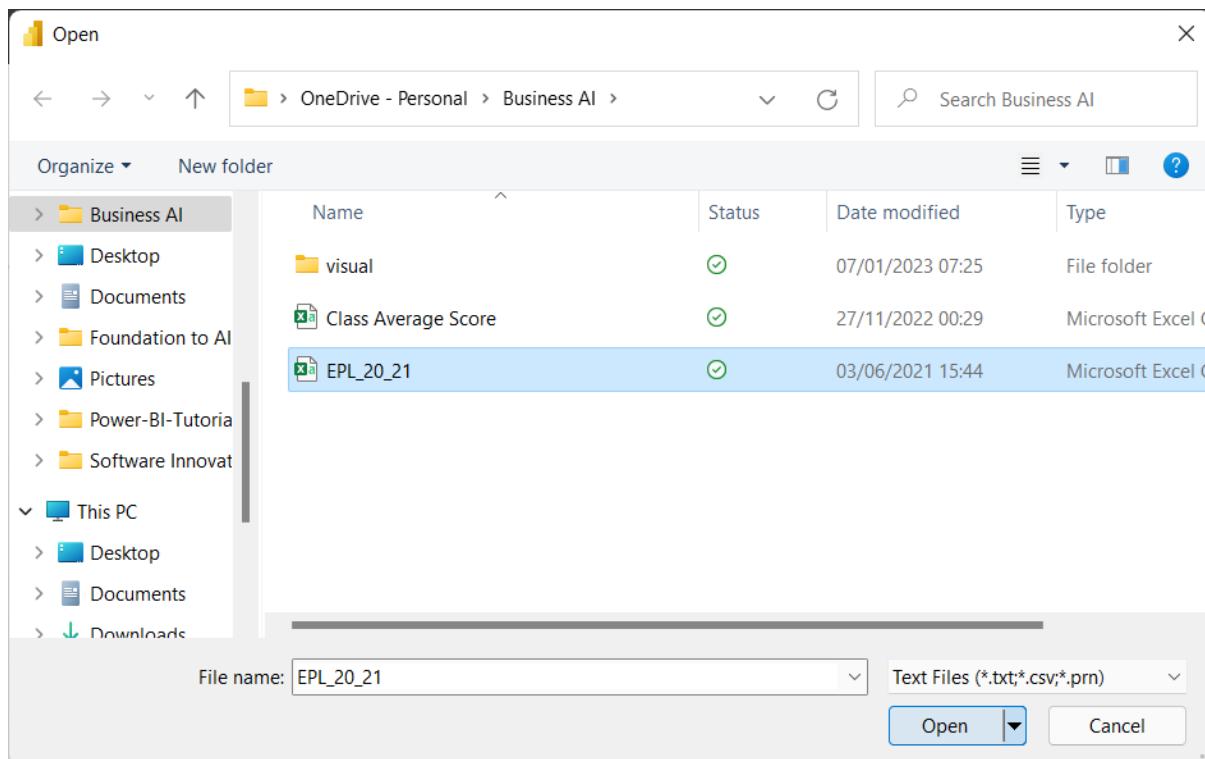


Fig 31. Power BI modelling process

Click "Load" to import the data into Power BI.

When loading the data in, we need to attention to some key info on the dialog box:

- i. the data type is calibrated based on the first 200 rows by default; however, this may be adjusted to use the complete dataset or no data type at all.
- ii. The delimiter for a csv file is commas. This can vary also depending on the file. When importing a text file separated by "Tab", the delimiter would be changed accordingly.
- iii. File origin/encoding. The encoding defaulted to 1252: Western European (Windows).

Fig 32. Power BI modelling process

The data will be imported as a table.

Fig 32. Power BI modelling process

Now for using M Language to import the dataset into power BI;

- Click on the "Get Data" button in the ribbon at the top of the Power BI interface and from the list of file types, select "Blank Query".

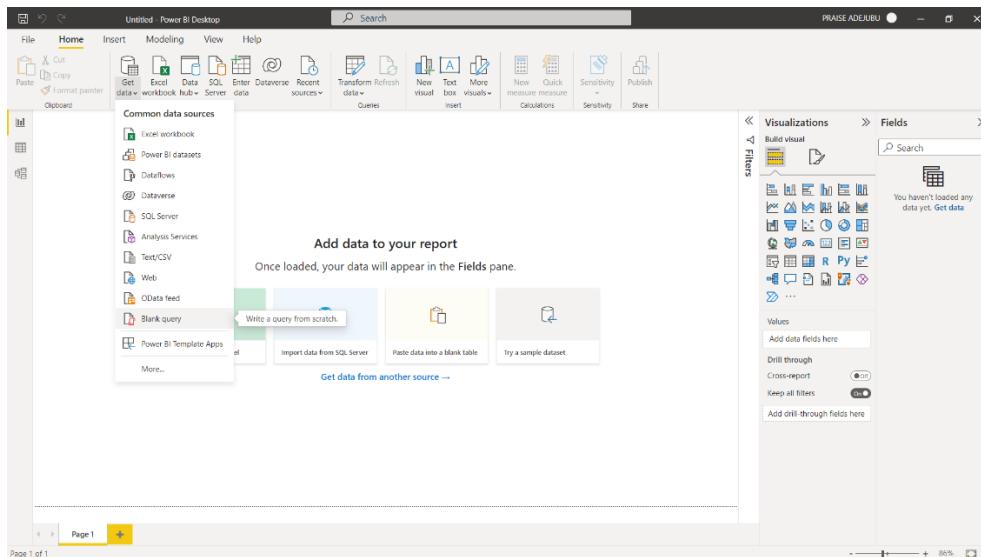


Fig 33. Power BI modelling process

- Next click on Advanced Query:

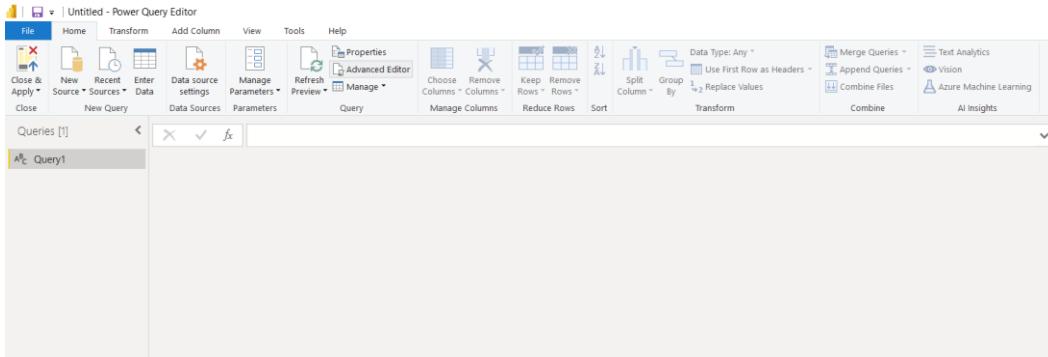


Fig 34. Power BI modelling process

- Type in the source destination as follows:



Fig 35. Power BI modelling process

Now for creating the clubs logo table, I sourced the data from dream league soccer kits website, <https://dreamleaguesoccerkits.com/2021/05/premier-league-club-logo/> :

- Click on the “Enter Data” on the Home tab:

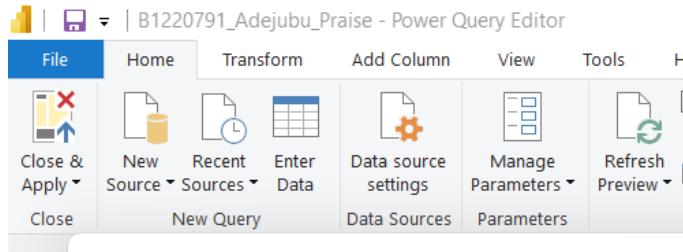


Fig 36. Power BI modelling process

- Go to the URL links and copy them:

### Premier League 2020-2021 Logo URLs

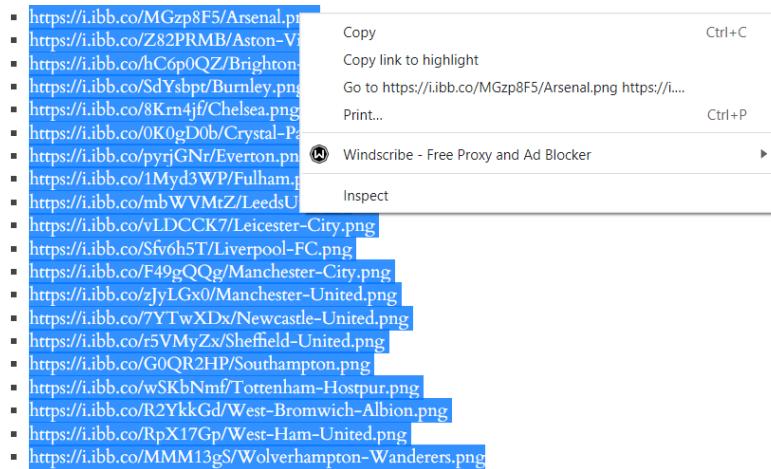


Fig 37. Power BI modelling process

- Paste them in the empty column Table and rename it to Logos, then click “Okay”.

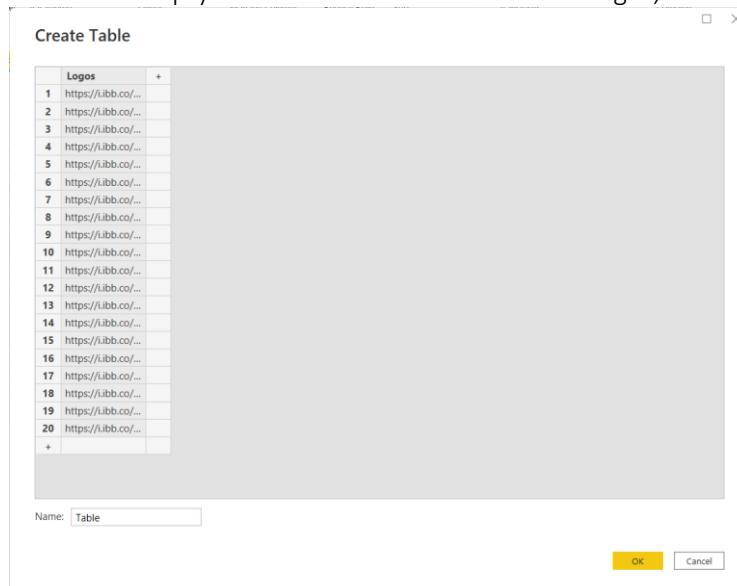


Fig 38. Power BI modelling process

The new table comes in the queries.

The screenshot shows the Power Query Editor interface with the following details:

- File**: B1220791\_Adejuju,Praise - Power Query Editor
- Home**: Transform, Add Column, View, Tools, Help
- Data Sources**: Close & Apply, New Source, Recent Sources, Enter Data, Data source settings, Manage Parameters, Refresh Preview, Advanced Editor, Properties, Choose Columns, Remove Columns, Manage Columns, Keep Rows, Remove Rows, Sort, Transform, Data Type: Text, Use First Row as Headers, Merge Queries, Append Queries, Combine files, Text Analytics, Azure Machine Learning, AI Insights
- Queries [7]**: Player information, Match statistics, booking, Performance stat, Club Logo, Clubs, Table (selected).
- Table View**: A list of 20 URLs for club logos, starting with <https://libb.co/MGzpBFS/Arsenal.png>.
- Query Settings** pane on the right shows:
  - PROPERTIES**: Name: Table, All Properties
  - APPLIED STEPS**: Source, Changed Type
- Bottom Status Bar**: 1 COLUMN, 20 ROWS, Column profiling based on top 1000 rows, PREVIEW DOWNLOADED AT 13:03

Fig 39. Power BI modelling process

## Data Cleaning:

The next step was to clean up the pre-process the data. This will was done in Power query, to do this,

💡 I clicked on the “Transform data”, on the home tab:

The screenshot shows the Power Query Editor interface with the following details:

- File**: Untitled - Power Query Editor
- Home**: Transform, Add Column, View, Tools, Help
- Data Sources**: Close & Apply, New Source, Recent Sources, Enter Data, Data source settings, Manage Parameters, Refresh Preview, Advanced Editor, Properties, Choose Columns, Remove Columns, Manage Columns, Keep Rows, Remove Rows, Sort, Transform, Data Type: Text, Use First Row as Headers, Merge Queries, Append Queries, Combine files, Text Analytics, Azure Machine Learning, AI Insights
- Queries [1]**: EPL\_20\_21 (selected).
- Table View**: A table with columns: Club, Nationality, Position, Age, Matches, Starts, Mins.
- Query Settings** pane on the right shows:
  - PROPERTIES**: Name: EPL\_20\_21, All Properties
  - APPLIED STEPS**: Source, Promoted Headers, Changed Type
- Bottom Status Bar**: 18 COLUMNS, 532 ROWS, Column profiling based on top 1000 rows, PREVIEW DOWNLOADED AT 13:25

Fig 40. Power BI modelling process

The dataset was cleansed once the previous step of importing it into Power BI was completed, allowing us to use it for our analysis. The cleaning procedure followed the thinking process below.

**Promoted Headers:** The program recognized the next row on the data as the main headings, and thus Transfers the First row to be the header of the table. In this case it happened automatically given that the original csv file was imported with column 1, column 2... as the headings. This step can be done manually:

- After the table is imported, select the “Use First Row as Header” option on the home tab:

A <sub>1</sub> Column1	A <sub>1</sub> Column2	A <sub>1</sub> Column3	A <sub>1</sub> Column4	A <sub>1</sub> Column5	A <sub>1</sub> Column6	A <sub>1</sub> Column7
1 Name	Club	Nationality	Position	Age	Matches	Starts
2 Mason Mount	Chelesa	ENG	MF,FW	21	36	32
3 Edouard Mendy	Chelesa	SEN	GK	28	31	31
4 Timo Werner	Chelesa	GER	FW	24	35	29
5 Ben Chilwell	Chelesa	ENG	DF	23	27	27
6 Reece James	Chelesa	ENG	DF	20	32	25
7 César Azpilicueta	Chelesa	ESP	DF	30	26	24
8 N'Golo Kanté	Chelesa	FRA	MF	29	30	24
9 Jorginho	Chelesa	ITA	MF	28	28	23
10 Thiazen Silva	Chelesa	RRΔ	DF	25	23	23

Fig 41. Power BI modelling process

**Changed Type:** The next cleansing was to change the data types, this is to correctly recognize what each column type it is. This step was also done automatically, but It can be done alternately using this M formula:

```
= Table.TransformColumnTypes(#"Promoted Headers",{{"Name", type text}, {"Club", type text}, {"Nationality", type text}, {"Position", type text}, {"Age", Int64.Type}, {"Matches", Int64.Type}, {"Starts", Int64.Type}, {"Mins", Int64.Type}, {"Goals", Int64.Type}, {"Assists", Int64.Type}, {"Passes_Attempted", Int64.Type}, {"Perc_Passes_Completed", type number}, {"Penalty_Goals", Int64.Type}, {"Penalty_Attempted", Int64.Type}, {"xG", type number}, {"xA", type number}, {"Yellow_Cards", Int64.Type}, {"Red_Cards", Int64.Type}})
```

- This text above should be inputted to the “formula bar” to transform the columns type.

Fig 42. Power BI modelling process

**Checking for Nulls, Blanks from the dataset:** I checked for empty values in all columns but there were none, all columns were 100 percent.

Fig 43. Power BI modelling process

- If there are null values (empty cells) in the dataset, you can replace them with a default value using the "Replace Values" function. To do this, click on the column header with the null values, then select "Replace Values" from the ribbon at the top of the Power BI interface.

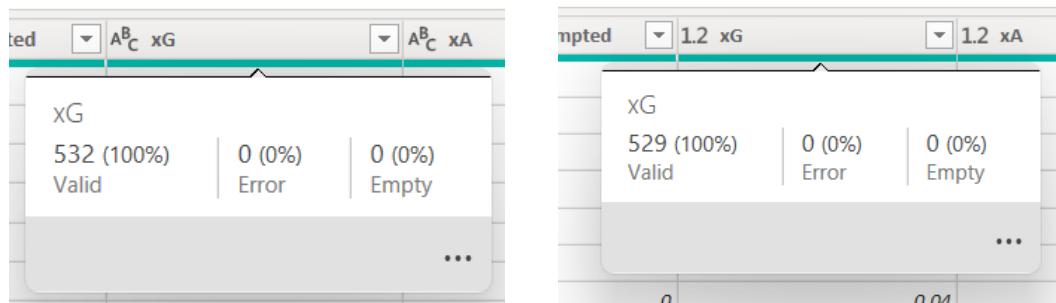
**Remove unnecessary columns:** If there are columns in the dataset that you do not need for your analysis, you can remove them to reduce the size of the dataset and make it easier to work with. To do this, click on the column header, then select "Remove Columns" from the ribbon at the top of the Power BI interface.

**Check and Remove errors from the dataset:** I noticed the “xG” and the “Perc\_Passes\_Completed” columns had wrong values, in that; Research online explains that “xG”, Expected Goals is a metric designed to measure the probability of a shot resulting in a goal. An xG model uses historical information from thousands of shots with similar characteristics to estimate the likelihood of a goal on a scale between 0 and 1, [5] as explained in <https://statsbomb.com/soccer-metrics/expected-goals-xg-explained/>. So any number above 1 is an error in the xG column.

Also in the “Perc\_Passes\_Completed” column, there was a -1 value, which is impossible to rate a pass a -1.

- So the next action was to filter the data using M Language:

```
= Table.SelectRows(#"Changed Type", each ([xG] <> 1.16) and ([Perc_Passes_Completed] <> -1))
```



- This can also be done manually, as shown in the illustrations below:

The image contains two side-by-side screenshots of the Power BI Advanced Editor interface.

**Left Screenshot:** Shows the 'Goals' column being filtered. A context menu is open over the column header, with the 'Number Filters' option selected. A dropdown menu shows values from 0.44 to 0.62, with '1.16' highlighted. Below the dropdown are 'OK' and 'Cancel' buttons.

**Right Screenshot:** Shows the 'Perc\_Passes\_Completed' column being filtered. A context menu is open over the column header, with the 'Number Filters' option selected. A dropdown menu shows values ranging from -1 to 56.2, with several values checked (e.g., 0, 28.6, 43.2, 46.5, 48.2, 49.1, 49.9, 50, 50.1, 50.7, 51.3, 52.9, 54.5, 55.4, 56.2). Below the dropdown are 'OK' and 'Cancel' buttons.

- Next step was to manually change the percentage column by adding "%" to the value and then changing the column type to percentage. The M formula for this is:

```
= Table.TransformColumns(#"Filtered Rows1", {"Perc_Passes_Completed", each Text.From(_, "en-GB") & "%", type text})
```

A screenshot of the Power BI Advanced Editor's formula bar. The formula is displayed as:

```
= Table.TransformColumns(#"Filtered Rows1", {"Perc_Passes_Completed", each Text.From(_, "en-GB") & "%", type text}})
```

- And then to change the column type, type this M formula into the Formular bar and press enter:

```
= Table.TransformColumnTypes(#"Added Suffix", {"Perc_Passes_Completed", Percentage.Type})
```

A screenshot of the Power BI Advanced Editor's formula bar. The formula is displayed as:

```
= Table.TransformColumnTypes(#"Added Suffix", {"Perc_Passes_Completed", Percentage.Type})
```

This can be done manually by firstly selecting the drop menu on “Format” from the Add column bar, them scrolling down to “Add Suffix”.

- Next go to the “Perc\_Passes\_Completed” column Header and click on the left side symbol icon, then choose “Percentage from the list.

**Checking for Duplicate rows:** I checked for any rows that were repeated in the table datand found none. This means all data on the table are unique and almost ready for analyzing. The steps I used are illustrated below;

- Firstly select the first column “Name”:

The screenshot shows the Power Query Editor interface with the 'EPL\_20\_21' query selected. The 'Name' column is highlighted in blue. The 'APPLIED STEPS' pane on the right shows a single step named 'Changed Type'.

- Then hold ‘Shift’ key on the keyboard and select the last column “Red\_Card”:

The screenshot shows the Power Query Editor interface with the 'Red\_Card' column highlighted in blue. The 'APPLIED STEPS' pane on the right shows a single step named 'Changed Type'.

Fig 45. Power BI modelling process

- Next 'Right click' the mouse and select "Remove Duplicates" from options:

The screenshot shows the Microsoft Power Query Editor interface. A context menu is open over a table in the query editor, specifically over the column headers. The menu path 'Remove Duplicates' is highlighted. Other options visible in the menu include 'Remove Columns', 'Remove Other Columns', 'Add Column From Examples...', 'Remove Errors', 'Replace Values...', 'Fill', 'Change Type', 'Merge Columns', 'Group By...', 'Unpivot Columns', 'Unpivot Only Selected Columns', and 'Move'. The main Power Query ribbon is at the top, and the status bar at the bottom indicates '18 COLUMNS, 532 ROWS' and 'Column profiling based on top 1000 rows'.

If there are duplicate rows in the dataset, these steps would remove them and the number of Rows would reduce in number.

## BI DATA MODELLING VIA STAR SCHEMA - FACTS AND DIMENSIONS

### Adding Index column:

The next step was to introduce a column that would help with my relationship throughout my model. A common shared data that would help in the relationship chart.

- The M Language formula for this is: = Table.AddIndexColumn(#"Removed Duplicates", "Index", 1, 1, Int64.Type)

The screenshot shows the M Language formula bar in the Power Query Editor. The formula = Table.AddIndexColumn(#"Removed Duplicates", "Index", 1, 1, Int64.Type) is displayed. The formula bar has a preview pane on the right showing the current state of the table.

- Alternatively, this can also be done: goto “IndexColumn”, on the “Add Column” bar, select “From 1”.

- A new column “Index”, will appear at the end of the table, as shown below:

	1.2	Penalty_Attempted	12_XG	12_XA	Yellow_Cards	Red_Cards	Index
1	1	0	0.21	0.24	2	0	1
2	0	0	0	0	2	0	2
3	0	0	0.41	0.21	2	0	3
4	0	0	0.1	0.11	3	0	4
5	0	0	0.06	0.12	3	0	5
6	0	0	0.03	0.11	5	1	6
7	0	0	0.04	0.05	7	0	7
8	7	9	0.31	0.09	2	0	8
9	0	0	0.05	0.02	5	1	9
10	0	0	0.08	0	3	0	10
11	0	0	0.05	0.09	4	0	11
12	0	0	0.06	0.02	0	0	12
13	0	0	0.28	0.14	2	0	13
14	0	0	0.87	0.09	2	0	14
15	0	0	0.01	0.02	2	1	15
16	0	0	0.15	0.28	3	0	16
17	0	0	0.56	0.07	0	0	17
18	0	0	0.16	0.11	2	0	18
19	0	0	0.12	0.26	0	0	19
20	0	0	0.58	0.09	1	0	20
21	0	0	0	0	1	0	21
22	0	0	0.01	0.04	0	0	22
23	0	0	0	0	0	0	23
24	0	0	0	0	0	0	24
25	0	0	0	0	0	0	25
26	0	0	0	0	0	0	26
27	0	0	0.06	0.16	0	0	27
28	0	0	0	0.01	3	0	28
29	0	0	0.07	0	4	0	29
30	1	1	0.98	0.98	4	0	30

**Reordering column:** Manually drag the Index column to the start of the table, just before the “Name” column.

The screenshot shows the Power Query Editor interface with the 'EPL\_20\_21' query selected. The table has 19 columns and 529 rows. The 'Index' column is highlighted in yellow at the top of the table. The 'APPLIED STEPS' pane on the right shows the 'Reordered Columns' step, indicating that the 'Index' column was moved to the beginning of the table.

**Rename columns:** Now the “Index” column is not descriptive or easy to understand, I renamed it to “Player\_ID”, to make the data easier to work with. To do this, ‘Right click’ on the “Index” column header, then select "Rename" from the list.

- This can easily be done using M language also with the formula: =  
Table.RenameColumns(#"Added Index",{{"Index", "Players\_ID"}}).

The screenshot shows the Power Query Editor interface with the 'B1220791\_Adejibu\_Praise' query selected. The table has 19 columns and 529 rows. The 'Player\_ID' column is highlighted in yellow at the top of the table. The 'APPLIED STEPS' pane on the right shows the 'Renamed Columns' step, indicating that the 'Index' column was renamed to 'Player\_ID'.

These were the steps I used in cleaning the data to be ready for analysing and by following these steps, I effectively cleaned this dataset for modelling and analysis in Power BI.

## Creating Dimension tables

I duplicated the table severally and removed unwanted columns to create my Fact and Dimension tables.

**Duplicating Tables:** To create the dimension tables, after completing the pre-processing, I duplicated the main table now and the steps are shown below:

- ‘Right’ click on the Table “EPL\_20\_21!” in the “Queries” Station, by the far left, and select “Duplicate” from the list drop down.

The screenshot shows the Power Query Editor interface. On the left, the 'Queries [1]' pane lists 'EPL\_20\_21'. A context menu is open over this query, with 'Duplicate' highlighted. The main workspace displays a table with 529 rows and 9 columns, including columns for Name, Club, Nationality, Position, Age, and Matches. The 'APPLIED STEPS' pane on the right shows the history of transformations applied to the query.

This screenshot shows the same Power Query Editor interface as above, but with a different view. The 'Queries [1]' pane now shows 'EPL\_20\_21' with its properties expanded. The context menu over the query again highlights 'Duplicate'. The main workspace and applied steps pane are visible.

**Removing unwanted Columns for Dimension tables:** This is how I removed columns from tables is illustrated using M Language: I used this formula: = Table.RemoveColumns(#"Reordered Columns", {"Matches", "Starts", "Mins", "Goals", "Assists", "Passes\_Attempted", "Perc\_Passes\_Completed", "Penalty\_Goals", "Penalty\_Attempted", "xG", "xA", "Yellow\_Cards", "Red\_Cards"})

```
= Table.RemoveColumns(#"Reordered Columns", {"Matches", "Starts", "Mins", "Goals", "Assists", "Passes_Attempted", "Perc_Passes_Completed", "Penalty_Goals", "Penalty_Attempted", "xG", "xA", "Yellow_Cards", "Red_Cards"})
```

- Adding any column into the list above to remove them. To do this, just select the columns wanted or the once not wanted (by holding down 'Ctrl' key and 'Right click' on one of them, select "Remove Other Columns or Remove Columns in whichever preference.

The screenshot shows the Power Query Editor interface with a dimension table named 'Players\_ID' loaded. A context menu is open over the 'Age' column, specifically at the row for player number 29. The menu options include Copy, Remove Columns, Remove Other Columns (which is highlighted in grey), Add Column From Examples..., Remove Duplicates, Remove Errors, Replace Values..., Fill, Change Type, Merge Columns, Group By..., Unpivot Columns, Unpivot Other Columns, Unpivot Only Selected Columns, and Move. The 'Remove Other Columns' option is selected, indicating it will be used to remove the 'Age' column from the table.

- I repeated this step for all four Dimension tables and the one bridge table.

The screenshot shows the Power BI 'Queries' pane, which lists six tables: 'Performance stat', 'Player information', 'Match statistics', 'booking', 'Club Logo', and 'Clubs'. The 'Performance stat' table is currently selected, indicated by a yellow highlight.

There is one table that I created using external data to get the Logos of each clubs. After loading the url's into the table, I just added an "index" table to it and renamed it "Players ID" in order to share data with the "Clubs" data.

1	https://i.ibb.co/MGzp8f5/Arsenal.png	1
2	https://i.ibb.co/z82PRMB/Aston-Villa.png	2
3	https://i.ibb.co/hC6p0QZ/Brighton-Hove-Albion.png	3
4	https://i.ibb.co/SdYsbpt/Burnley.png	4
5	https://i.ibb.co/8Krnqjf/Chesterfield.png	5
6	https://i.ibb.co/OK0gDOb/Crystal-Palace.png	6
7	https://i.ibb.co/pyrjGNr/Everton.png	7
8	https://i.ibb.co/1Myd3WP/Fulham.png	8
9	https://i.ibb.co/mbWVMTz/LeedsUtd.png	9
10	https://i.ibb.co/vLDCCk7/Leicester-City.png	10
11	https://i.ibb.co/Sfv6h5T/Liverpool-FC.png	11
12	https://i.ibb.co/F49gQQg/Manchester-City.png	12
13	https://i.ibb.co/zLyLGx0/Manchester-United.png	13
14	https://i.ibb.co/7TwxDx/Newcastle-United.png	14
15	https://i.ibb.co/r5VMy2x/Sheffield-United.png	15
16	https://i.ibb.co/G0QR2HP/Southampton.png	16
17	https://i.ibb.co/w5KbNmff/Tottenham-Hotspur.png	17
18	https://i.ibb.co/R2YkkGd/West-Bromwich-Albion.png	18
19	https://i.ibb.co/RpX17Gp/West-Ham-United.png	19
20	https://i.ibb.co/MMM13g5/Wolverhampton-Wanderers....	20

The next and final process was to create a separate “Clubs” table to Bridge/link the “Club logos” table data to the rest of the model. I achieved this using these steps:

= Table.SelectColumns(#"Reordered Columns", {"Club"})

The screenshot shows a Power BI desktop interface with a table containing 24 rows, all labeled "Chelsea". A context menu is open over the first row, specifically for the "Club" column. The menu includes options like Copy, Remove, Remove Other Columns, Duplicate Column, Add Column From Examples..., Remove Duplicates (highlighted in grey), Remove Errors, Change Type, Transform, Replace Values..., Replace Errors..., Split Column, Group By..., Fill, Unpivot Columns, Unpivot Only Selected Columns, Rename..., Move, Drill Down, and Add as New Query.

I duplicated the original table and removed all columns except the “Club” column as shown in the column above.

- Next, I sorted the “club” column in Ascending order, using M Language: =  
Table.Sort(#"Renamed Columns1",{{"Club", Order.Ascending}})

Queries [6]

Player information  
Match statistics  
booking  
Performance stat  
Club Logo  
**Clubs**

`= Table.Distinct(#"Removed Other Columns")`

Club

1 Chelsea  
2 Manchester City  
3 Manchester United  
4 Liverpool FC  
5 Leicester City  
6 West Ham United  
7 Tottenham Hotspur  
8 Arsenal  
9 Leeds United  
10 Everton  
11 Aston Villa  
12 Newcastle United  
13 Wolverhampton Wanderers  
14 Crystal Palace  
15 Southampton  
16 Brighton  
17 Burnley  
18 Fulham  
19 West Bromwich Albion  
20 Sheffield United

PROPERTIES

Name: Clubs

APPLIED STEPS

Source  
Promoted Headers  
Changed Type  
Filtered Rows  
Filtered Rows1  
Added Suffix  
Changed Type1  
Filtered Rows2  
Removed Duplicates  
Added Index  
Renamed Columns  
Reordered Columns  
Removed Other Columns  
Removed Duplicates1  
Added Index1  
Renamed Columns1  
Sorted Rows  
Removed Columns  
Added Index2  
Renamed Columns2

PREVIEW DOWNLOADED AT 15:45

- Finally, adding an “Index” column and renaming it “Player\_ID” with these M Formulas:  
#"Added Index2" = Table.AddIndexColumn(#"Removed Columns", "Index", 1, 1, Int64.Type),  
#"Renamed Columns2" = Table.RenameColumns(#"Added Index2",{{"Index", "Player\_ID"}})

`= Table.RenameColumns(#"Added Index2",{{"Index", "Player_ID"}})`

	Club	Player_ID
1	Arsenal	1
2	Aston Villa	2
3	Brighton	3
4	Burnley	4
5	Chelsea	5
6	Crystal Palace	6
7	Everton	7
8	Fulham	8
9	Leeds United	9
10	Leicester City	10
11	Liverpool FC	11
12	Manchester City	12
13	Manchester United	13
14	Newcastle United	14
15	Sheffield United	15
16	Southampton	16
17	Tottenham Hotspur	17
18	West Bromwich Albion	18
19	West Ham United	19
20	Wolverhampton Wanderers	20

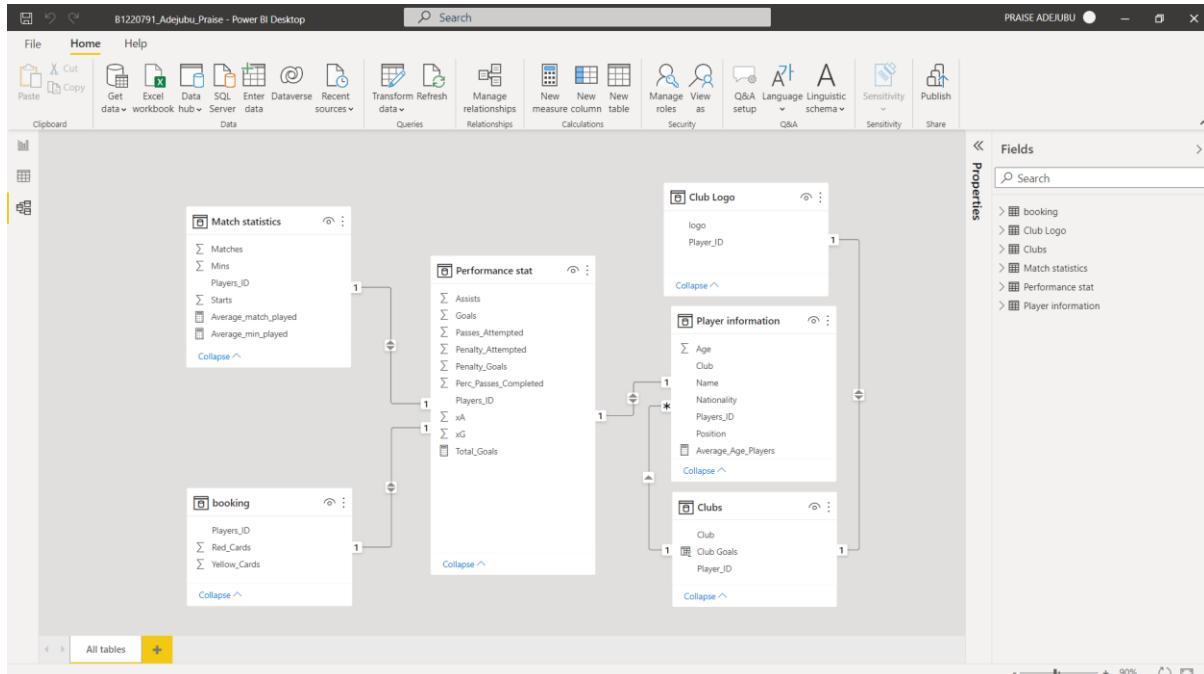
This created a way for my model to share a relationship, using the “Player\_ID” table.

The screenshot shows the Power Query Editor interface. A query named "Performance stat" is open, displaying a preview of 529 rows. The query uses Table.SelectColumns to extract specific columns: "Goals", "Assists", "Passes Attempted", "%Perc\_Passes\_Completed", "Penalty\_Goals", and "Penalty\_Attended". The preview table has 9 columns and 30 rows. The "APPLIED STEPS" pane on the right lists various transformations applied to the data, such as Promoted Headers, Changed Type, Filtered Rows, and Removed Other Columns.

- To analyze the data for this project, relationships must be established between the tables in the model. To create these relationships, we will select "Manage relationships" from the home tab of the data model section. This will open a dialog box, and we will select "New" to create a new relationship.

The screenshot shows the Power BI Desktop interface with the "Home" tab selected. The ribbon includes File, Home, Help, Data, Queries, Relationships, Calculations, Security, Q&A, Sensitivity, and Publish. Below the ribbon, four data tables are listed: "Match statistics", "Performance stat", "booking", and "Player information". The "Properties" pane on the right shows the details for the "Player information" table, including fields like Age, Club, Name, Nationality, Player\_ID, Position, and Average\_Age\_Players.

Once all the relationships have been created, we will have a Star schema model where one fact table (Performance stat) is connected to several dimension tables (Player Information, Match statistics, Bookings, Club Logo, and Clubs). The Clubs table will serve as a bridge table between the Club Logo table and other tables.



### C. DAX and M Language:

Several measures were created using DAX to calculate the rankings for each metric in the visualizations. The following DAX measures and column were created:

	DAX Description	DAX Formula
1	DAX calculated column "Club Goals" This is to calculate the total number of goals for each club and create a column.	Club Goals = <code>SUMX(DISTINCT('Clubs'[Club]),CALCULATE(SUM('Performance stat'[Goals])))</code>
2	DAX calculated measure of "Average_match_played". This is to create a measure to get every players average number of matches played that can be called for visualization without taking computing space.	Average_match_played = <code>AVERAGE('Match statistics'[Matches])</code>

3	DAX calculated measure of "Average_min_played" This is to create a measure to get the average number of minutes played by each player.	Average_min_played = AVERAGE('Match statistics'[Mins])
4	DAX calculated measure of "Average_Age_Players". This measure calculates the average age of players across the clubs in the league	Average_Age_Players = AVERAGE('Player information'[Age])

Table. 3 DAX Formulas and descriptions:

M Language was used to clean and transform the data before creating the star schema. These are some of the M Language formulas used in the model:

	Description	M Language Formula sample in model
1	To filter errors from the data using M Language from the ([xG] and "Perc_Passes_Completed" table at the same time.	= Table.SelectRows(#"Changed Type", each ([xG] <> 1.16) and ([Perc_Passes_Completed] <> -1))
2	Adding suffix "%" to a "Perc_Passes_Completed" column	= Table.TransformColumns(#"Filtered Rows1", {"Perc_Passes_Completed", each Text.From(_,"en-GB") & "%", type text}})
3	Changing the "Perc_Passes_Completed" column type to percentage.	= Table.TransformColumnTypes(#"Added Suffix", {"Perc_Passes_Completed", Percentage.Type})
4	Creating an index column to the table. In this case the "Player_ID" column was introduced to the table as a relationship tool.	= Table.AddIndexColumn(#"Removed Duplicates", "Index", 1, 1, Int64.Type)
5	Renaming columns: Here the "Index" column is not descriptive or easy to understand, I renamed it to "Player_ID", to make the data easier to work with.	= Table.RenameColumns(#"Added Index", {"Index", "Players_ID"}).

Table. 3 DAX Formulas and descriptions:

## DASHBOARDS

The pages were organized by metric, with each page containing visualizations showing general and vital information's for the 2020/2021 English premier league season for each club.

The Power BI dashboard for this analysis included the following pages:



Fig. Power Dashboard page 1

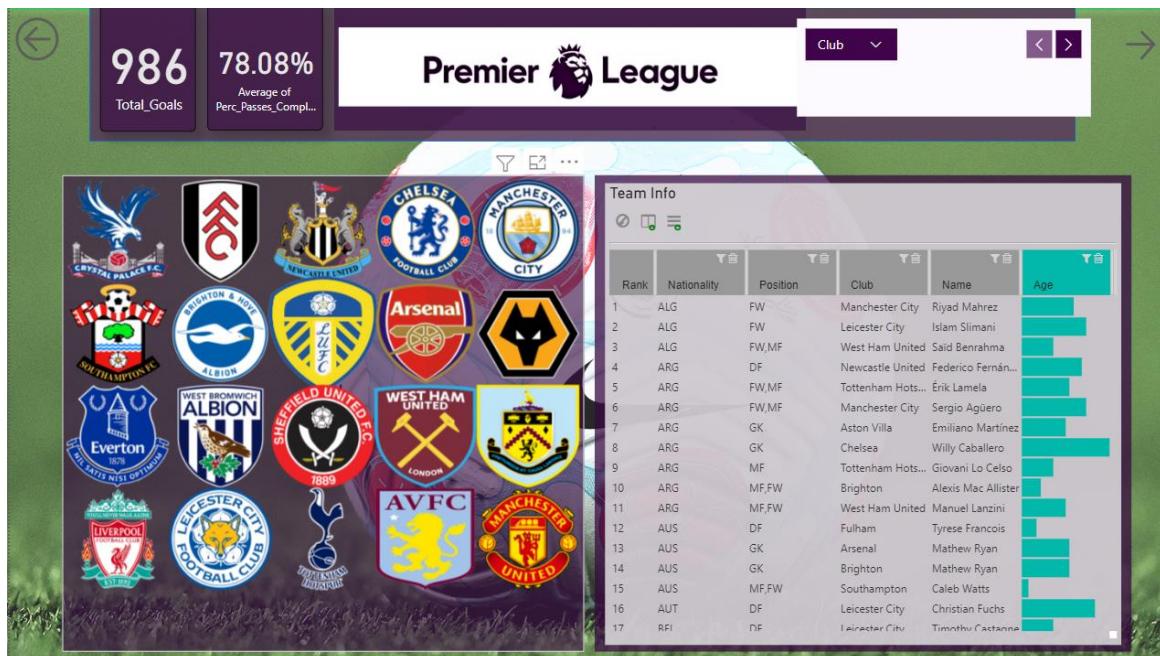


Fig. Power Dashboard page 2

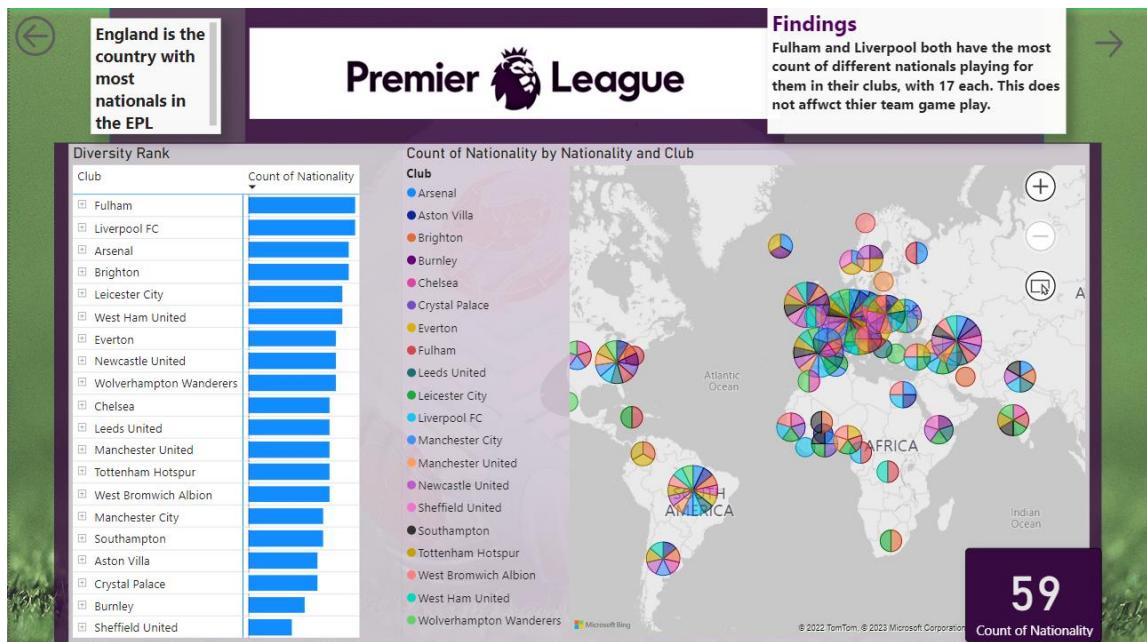


Fig. Power Dashboard page 3

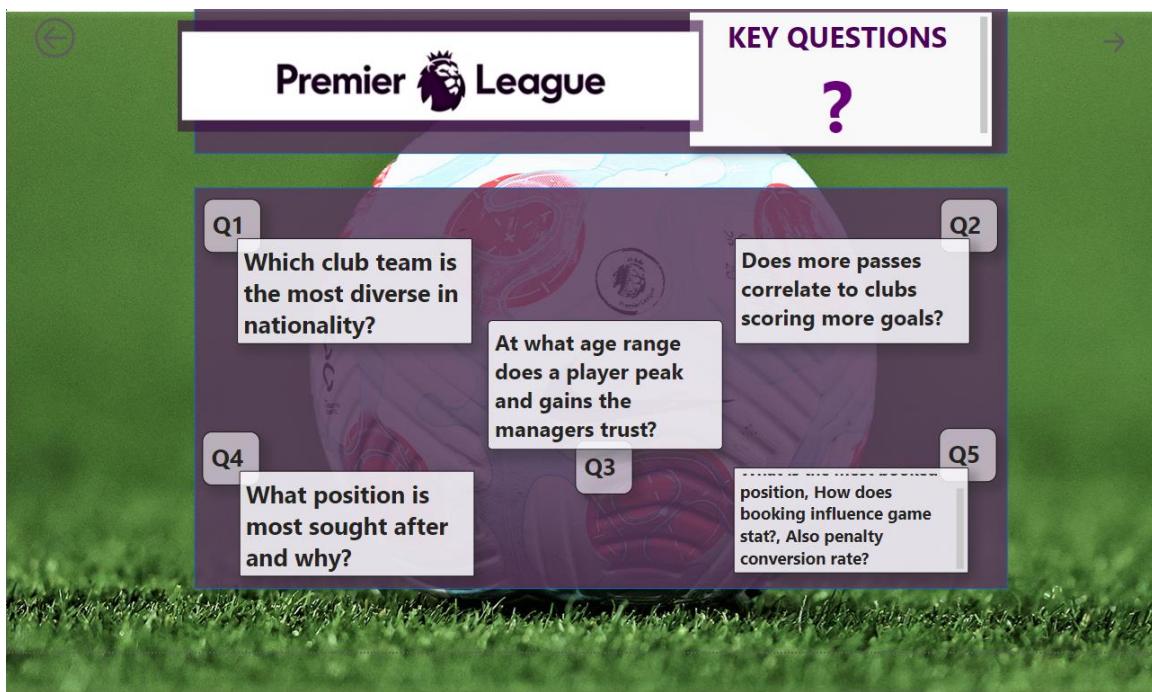


Fig. Power Dashboard page 4



Fig. Power Dashboard page 5



Fig. Power Dashboard page 6

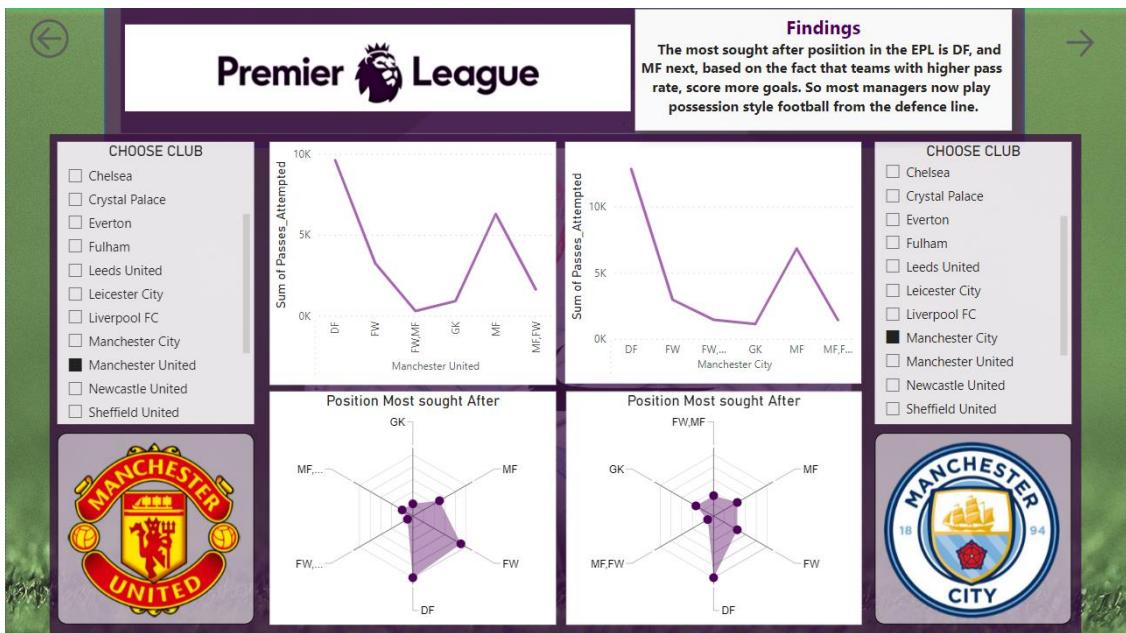


Fig. Power Dashboard page 7

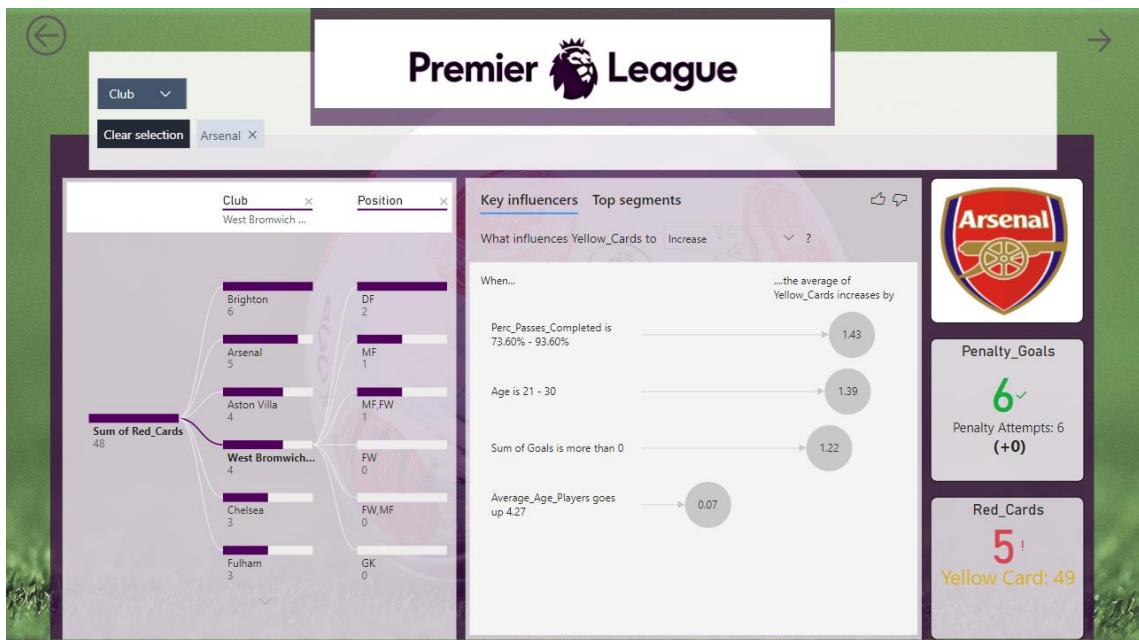


Fig. Power Dashboard page 8

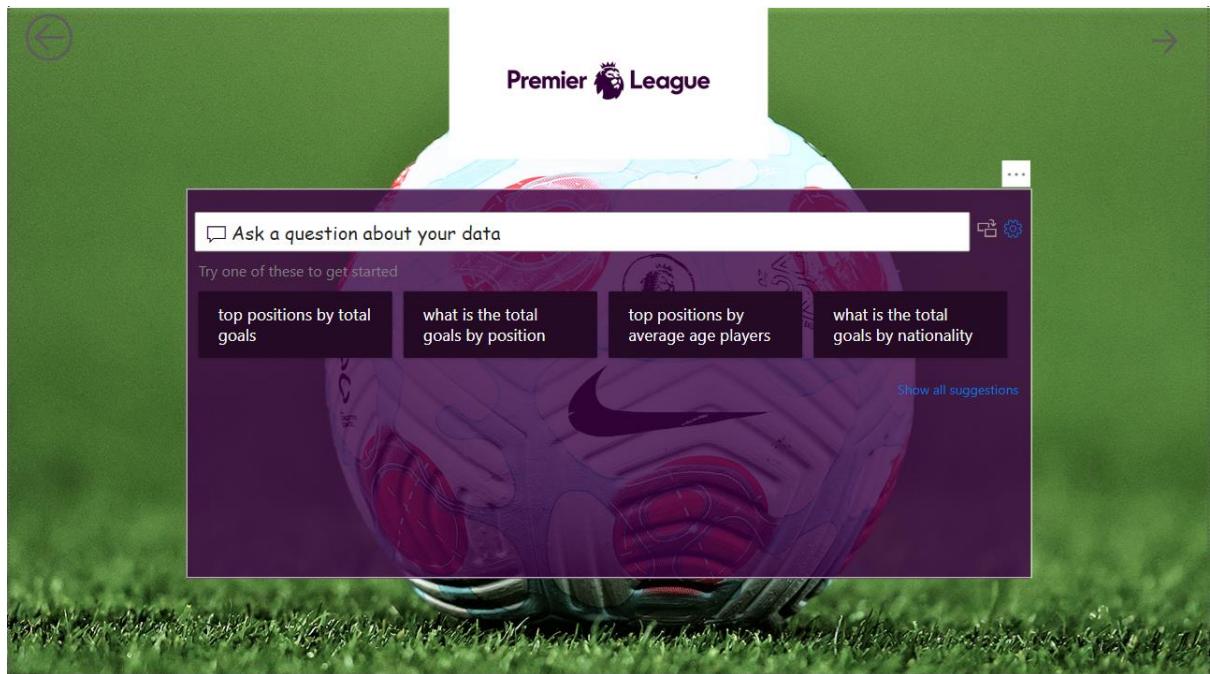


Fig. Power Dashboard page 9



Fig. Power Dashboard page 10

Note: All descriptions of Dashboards are made in the “Findings based on evaluation” Section.

Thank you.