Part 1 : Football Transfer Market Data EDA

This notebook covers data cleaning, dimensionality reduction, feature engineering and Exploratory Data Analysis (EDA) with the aid of visuals.

Importing Dependencies

In [1]:
```
# Import the required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import plotly.express as px
from sklearn.preprocessing import LabelEncoder
from datetime import datetime
import os

import warnings# Import this library to suppress the warnings
warnings.filterwarnings('ignore')   # The object 'warnings' is used to call the met
```

Loading the datasets

In [2]:
```
# List of file names
file_names = ["appearances", "club_games", "clubs", "competitions", "game_events",

# List to store DataFrames
dataframes = []

# Loop through each file name
for file_name in file_names:

    # Read the CSV file into a DataFrame
    df = pd.read_csv(f"{file_name}.csv")

    # Assign the DataFrame to the variable name
    globals()[f"{file_name}_df"] = df

    # Print name and shape of the DataFrame
    print(f"{file_name}_df:", df.shape)

    # Append the DataFrame to the list
    dataframes.append(df)

# Print a message indicating that data import is complete
print('Data imported successfully!')
```

```
appearances_df: (1485697, 13)
club_games_df: (128586, 11)
clubs_df: (426, 16)
competitions_df: (43, 10)
game_events_df: (652010, 10)
game_lineups_df: (86822, 9)
games_df: (64293, 23)
player_valuations_df: (440663, 9)
players_df: (30302, 23)
Data imported successfully!
```

Exploring the characteristics of the data

In [3]:
```
# printing the top 5 entries, info, number of unique values and describe for each d

for i, df in enumerate(dataframes):
    print("\n**********************************************************************
    print(file_names[i], df.shape)
    print("**********************************************************************
    print("\nhead:\n")
    print(df.head())
    print("\nGeneral information:\n")
    print(df.info())
    print("\nNumber of uniques values:\n")
    print(df.nunique())
    print("\nStatistical description:\n")
    print(df.describe())
```

```
********************************************************************************
***
appearances (1485697, 13)
********************************************************************************
***

head:

    appearance_id  game_id  player_id  player_club_id  player_current_club_id  \
0   2231978_38004  2231978      38004             853                     235
1   2233748_79232  2233748      79232            8841                    2698
2   2234413_42792  2234413      42792            6251                     465
3   2234418_73333  2234418      73333            1274                    6646
4  2234421_122011  2234421     122011             195                    3008

         date       player_name competition_id  yellow_cards  red_cards  \
0  2012-07-03  Aurélien Joachim            CLQ             0          0
1  2012-07-05    Ruslan Abyshov            ELQ             0          0
2  2012-07-05       Sander Puri            ELQ             0          0
3  2012-07-05   Vegar Hedenstad            ELQ             0          0
4  2012-07-05  Markus Henriksen            ELQ             0          0

   goals  assists  minutes_played
0      2        0              90
1      0        0              90
2      0        0              45
3      0        0              90
4      0        1              90

General information:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1485697 entries, 0 to 1485696
Data columns (total 13 columns):
 #   Column                Non-Null Count    Dtype
---  ------                --------------    -----
 0   appearance_id         1485697 non-null  object
 1   game_id               1485697 non-null  int64
 2   player_id             1485697 non-null  int64
 3   player_club_id        1485697 non-null  int64
 4   player_current_club_id  1485697 non-null  int64
 5   date                  1485697 non-null  object
 6   player_name           1485373 non-null  object
 7   competition_id        1485697 non-null  object
 8   yellow_cards          1485697 non-null  int64
 9   red_cards             1485697 non-null  int64
 10  goals                 1485697 non-null  int64
 11  assists               1485697 non-null  int64
 12  minutes_played        1485697 non-null  int64
dtypes: int64(9), object(4)
memory usage: 147.4+ MB
None

Number of uniques values:

appearance_id             1485697
game_id                     58417
player_id                   23740
player_club_id               1043
player_current_club_id        425
date                         3291
player_name                 23223
yellow_cards                    5
```

```
red_cards                     2
goals                         7
assists                       7
minutes_played              120
dtype: int64

Statistical description:

            game_id     player_id  player_club_id  player_current_club_id  \
count  1.485697e+06  1.485697e+06    1.485697e+06            1.485697e+06
mean   2.998498e+06  1.769102e+05    2.858144e+03            3.626989e+03
std    5.445426e+05  1.623144e+05    7.024775e+03            9.271304e+03
min    2.211607e+06  1.000000e+01    1.000000e+00           -1.000000e+00
25%    2.512848e+06  5.297900e+04    2.890000e+02            3.360000e+02
50%    2.899907e+06  1.264220e+05    8.550000e+02            9.310000e+02
75%    3.433282e+06  2.574480e+05    2.441000e+03            2.687000e+03
max    4.196249e+06  1.166093e+06    8.367800e+04            8.367800e+04

       yellow_cards     red_cards         goals       assists  minutes_played
count  1.485697e+06  1.485697e+06  1.485697e+06  1.485697e+06    1.485697e+06
mean   1.491495e-01  3.837929e-03  9.638843e-02  7.434019e-02    6.972077e+01
std    3.677364e-01  6.183205e-02  3.319122e-01  2.834616e-01    2.973195e+01
min    0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00    1.000000e+00
25%    0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00    5.300000e+01
50%    0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00    9.000000e+01
75%    0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00    9.000000e+01
max    2.000000e+00  1.000000e+00  6.000000e+00  6.000000e+00    1.200000e+02

********************************************************************************
***
club_games (128586, 11)
********************************************************************************
***

head:

   game_id  club_id  own_goals  own_position      own_manager_name  \
0  2221751      431        1.0           NaN         Lutz Göttling
1  2221755       83        3.0           NaN      Ralph Hasenhüttl
2  2222597     3725        2.0           2.0  Stanislav Cherchesov
3  2222627     2696        0.0          11.0        Andrey Kobelev
4  2222658     2410        0.0           2.0        Leonid Slutski

   opponent_id  opponent_goals  opponent_position opponent_manager_name  \
0           60             2.0                NaN     Christian Streich
1         4795             0.0                NaN            Tomas Oral
2          232             1.0                5.0            Unai Emery
3         4128             2.0               10.0         Rustem Khuzin
4          121             2.0               13.0          Dan Petrescu

  hosting  is_win
0    Home       0
1    Home       1
2    Home       1
3    Home       0
4    Home       0

General information:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 128586 entries, 0 to 128585
Data columns (total 11 columns):
 #   Column                        Non-Null Count   Dtype
---  ------                        --------------   -----
```

```
 0   game_id                128586 non-null  int64
 1   club_id                128586 non-null  int64
 2   own_goals              128582 non-null  float64
 3   own_position           90362 non-null   float64
 4   own_manager_name       127108 non-null  object
 5   opponent_id            128586 non-null  int64
 6   opponent_goals         128582 non-null  float64
 7   opponent_position      90362 non-null   float64
 8   opponent_manager_name  127108 non-null  object
 9   hosting                128586 non-null  object
 10  is_win                 128586 non-null  int64
dtypes: float64(4), int64(4), object(3)
memory usage: 10.8+ MB
None

Number of uniques values:

game_id                  64293
club_id                   2669
own_goals                   18
own_position                21
own_manager_name          5473
opponent_id               2669
opponent_goals              18
opponent_position           21
opponent_manager_name     5473
hosting                      2
is_win                       2
dtype: int64

Statistical description:

            game_id        club_id      own_goals  own_position  \
count  1.285860e+05  128586.000000  128582.000000  90362.000000
mean   3.001949e+06    4655.724223       1.456191      9.366991
std    5.478810e+05   10806.479172       1.401471      5.312555
min    2.211607e+06       1.000000       0.000000      1.000000
25%    2.512848e+06     352.000000       0.000000      5.000000
50%    2.903799e+06     995.000000       1.000000      9.000000
75%    3.433394e+06    3057.000000       2.000000     14.000000
max    4.196249e+06  112755.000000      19.000000     21.000000

         opponent_id  opponent_goals  opponent_position         is_win
count  128586.000000   128582.000000       90362.000000  128586.000000
mean     4655.724223        1.456191           9.366991       0.392539
std     10806.479172        1.401471           5.312555       0.488317
min         1.000000        0.000000           1.000000       0.000000
25%       352.000000        0.000000           5.000000       0.000000
50%       995.000000        1.000000           9.000000       0.000000
75%      3057.000000        2.000000          14.000000       1.000000
max    112755.000000       19.000000          21.000000       1.000000

**********************************************************************************
***
clubs (426, 16)
**********************************************************************************
***

head:

   club_id          club_code                name domestic_competition_id  \
0      105    sv-darmstadt-98    SV Darmstadt 98                       L1
                                    Yekaterinburg                      RU1
2      114  besiktas-istanbul        Besiktas JK                       TR1
```

```
3       12             as-rom             AS Roma                      IT1
4      148  tottenham-hotspur   Tottenham Hotspur                      GB1

   total_market_value  squad_size  average_age  foreigners_number  \
0                 NaN          29         26.1                 11
1                 NaN          25         27.6                 13
2                 NaN          32         26.7                 16
3                 NaN          26         26.7                 17
4                 NaN          29         25.6                 21

   foreigners_percentage  national_team_players  \
0                   37.9                      1
1                   52.0                      4
2                   50.0                     14
3                   65.4                     17
4                   72.4                     21

                    stadium_name  stadium_seats net_transfer_record  \
0  Merck-Stadion am Böllenfalltor          17500             €-1.60m
1             Yekaterinburg Arena          23000              €-770k
2                  Beşiktaş Park          42590            €-14.50m
3               Olimpico di Roma          73261            +€65.20m
4      Tottenham Hotspur Stadium          62062           €-126.40m

   coach_name  last_season                                              url
0         NaN         2023  https://www.transfermarkt.co.uk/sv-darmstadt-9...
1         NaN         2023  https://www.transfermarkt.co.uk/ural-ekaterinb...
2         NaN         2023  https://www.transfermarkt.co.uk/besiktas-istan...
3         NaN         2023  https://www.transfermarkt.co.uk/as-rom/startse...
4         NaN         2023  https://www.transfermarkt.co.uk/tottenham-hots...

General information:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 426 entries, 0 to 425
Data columns (total 16 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   club_id                  426 non-null    int64
 1   club_code                426 non-null    object
 2   name                     426 non-null    object
 3   domestic_competition_id  426 non-null    object
 4   total_market_value       0 non-null      float64
 5   squad_size               426 non-null    int64
 6   average_age              388 non-null    float64
 7   foreigners_number        426 non-null    int64
 8   foreigners_percentage    379 non-null    float64
 9   national_team_players    426 non-null    int64
 10  stadium_name             426 non-null    object
 11  stadium_seats            426 non-null    int64
 12  net_transfer_record      426 non-null    object
 13  coach_name               0 non-null      float64
 14  last_season              426 non-null    int64
 15  url                      426 non-null    object
dtypes: float64(4), int64(6), object(6)
memory usage: 53.4+ KB
None

Number of uniques values:

club_id                   426
club_code                 426
domestic_competition_id    14
```

```
total_market_value           0
squad_size                  31
average_age                 68
foreigners_number           28
foreigners_percentage      173
national_team_players       23
stadium_name               409
stadium_seats              379
net_transfer_record        273
coach_name                   0
last_season                 12
url                        426
dtype: int64

Statistical description:

            club_id  total_market_value  squad_size  average_age  \
count    426.000000                 0.0  426.000000   388.000000
mean    5314.525822                 NaN   24.352113    25.320619
std    11752.013818                 NaN    8.796949     1.525676
min        3.000000                 NaN    0.000000    18.300000
25%      421.000000                 NaN   24.000000    24.300000
50%     1139.500000                 NaN   27.000000    25.300000
75%     3415.750000                 NaN   29.000000    26.300000
max    83678.000000                 NaN   41.000000    29.000000

       foreigners_number  foreigners_percentage  national_team_players  \
count         426.000000             379.000000             426.000000
mean           10.946009              45.152507               4.793427
std             6.714998              19.944304               5.036297
min             0.000000               2.400000               0.000000
25%             6.000000              30.800000               1.000000
50%            12.000000              46.700000               3.500000
75%            16.000000              58.200000               7.000000
max            31.000000             100.000000              22.000000

       stadium_seats  coach_name  last_season
count     426.000000         0.0   426.000000
mean    24300.420188         NaN  2020.830986
std     17146.038353         NaN     3.207283
min         0.000000         NaN  2012.000000
25%     11006.000000         NaN  2019.000000
50%     20046.000000         NaN  2023.000000
75%     32798.000000         NaN  2023.000000
max     81365.000000         NaN  2023.000000

********************************************************************************
***
competitions (43, 10)
********************************************************************************
***

head:

  competition_id               competition_code  \
0            CIT                     italy-cup
1           NLSC           johan-cruijff-schaal
2            GRP                kypello-elladas
3           POSU  supertaca-candido-de-oliveira
4           RUSS              russian-super-cup

                           name            sub_type          type  \
0                     italy-cup      domestic_cup  domestic_cup
1           johan-cruijff-schaal  domestic_super_cup         other
```

```
2                kypello-elladas        domestic_cup  domestic_cup
3  supertaca-candido-de-oliveira  domestic_super_cup         other
4              russian-super-cup  domestic_super_cup         other

   country_id country_name domestic_league_code confederation  \
0          75        Italy                  IT1        europa
1         122  Netherlands                  NL1        europa
2          56       Greece                  GR1        europa
3         136     Portugal                  PO1        europa
4         141       Russia                  RU1        europa

                                                 url
0  https://www.transfermarkt.co.uk/italy-cup/star...
1  https://www.transfermarkt.co.uk/johan-cruijff-...
2  https://www.transfermarkt.co.uk/kypello-ellada...
3  https://www.transfermarkt.co.uk/supertaca-cand...
4  https://www.transfermarkt.co.uk/russian-super-...

General information:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 43 entries, 0 to 42
Data columns (total 10 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   competition_id        43 non-null     object
 1   competition_code      43 non-null     object
 2   name                  43 non-null     object
 3   sub_type              43 non-null     object
 4   type                  43 non-null     object
 5   country_id            43 non-null     int64
 6   country_name          36 non-null     object
 7   domestic_league_code  36 non-null     object
 8   confederation         43 non-null     object
 9   url                   43 non-null     object
dtypes: int64(1), object(9)
memory usage: 3.5+ KB
None

Number of uniques values:

competition_id          43
competition_code        42
name                    42
sub_type                11
type                     4
country_id              15
country_name            14
domestic_league_code    14
confederation            1
url                     43
dtype: int64

Statistical description:

       country_id
count   43.000000
mean    97.093023
std     69.766896
min     -1.000000
25%     39.500000
50%    122.000000
max    190.000000
```

```
********************************************************************************
***
game_events (652010, 10)
********************************************************************************
***

head:

                     game_event_id        date  game_id  minute  \
0  2f41da30c471492e7d4a984951671677  05/08/2012  2211607      77
1  a72f7186d132775f234d3e2f7bc0ed5b  05/08/2012  2211607      77
2  b2d721eaed4692a5c59a92323689ef18  05/08/2012  2211607       3
3  aef768899cedac0c9a650980219075a2  05/08/2012  2211607      53
4  5d6d9533023057b6619ecd145a038bbe  05/08/2012  2211607      74

            type  club_id  player_id  \
0          Cards      610       4425
1          Cards      383      33210
2          Goals      383      36500
3          Goals      383      36500
4  Substitutions      383      36500

                                       description  player_in_id  \
0              1. Yellow card  , Mass confrontation           NaN
1              1. Yellow card  , Mass confrontation           NaN
2  , Header, 1. Tournament Goal Assist: , Corner,...           NaN
3  , Right-footed shot, 2. Tournament Goal Assist...           NaN
4                                  , Not reported       49499.0

   player_assist_id
0               NaN
1               NaN
2           56416.0
3          146258.0
4               NaN

General information:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 652010 entries, 0 to 652009
Data columns (total 10 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   game_event_id     652010 non-null  object
 1   date              652010 non-null  object
 2   game_id           652010 non-null  int64
 3   minute            652010 non-null  int64
 4   type              652010 non-null  object
 5   club_id           652010 non-null  int64
 6   player_id         652010 non-null  int64
 7   description       317778 non-null  object
 8   player_in_id      413164 non-null  float64
 9   player_assist_id  29158 non-null   float64
dtypes: float64(2), int64(4), object(4)
memory usage: 49.7+ MB
None

Number of uniques values:

game_event_id       652010
date                  3358
```



```
minute                121
```

```
type                    4
club_id              2574
player_id           51478
description          4570
player_in_id        50149
player_assist_id     9439
dtype: int64

Statistical description:

            game_id         minute        club_id     player_id  player_in_id  \
count  6.520100e+05  652010.000000  652010.000000  6.520100e+05  4.131640e+05
mean   3.027107e+06      63.491683    4548.496678  1.962196e+05  2.386084e+05
std    5.918583e+05      21.908195   10796.108584  1.834003e+05  2.043951e+05
min    2.211607e+06      -1.000000       1.000000  1.000000e+01  1.000000e+01
25%    2.481046e+06      51.000000     336.000000  5.641600e+04  7.309700e+04
50%    2.942767e+06      68.000000     984.000000  1.352290e+05  1.863680e+05
75%    3.553565e+06      80.000000    3015.000000  2.862970e+05  3.441070e+05
max    4.196249e+06     120.000000  112755.000000  1.195049e+06  1.195054e+06

       player_assist_id
count      2.915800e+04
mean       1.142005e+05
std        1.575255e+05
min        1.000000e+01
25%        2.943400e+04
50%        5.728750e+04
75%        1.252508e+05
max        1.189238e+06

********************************************************************************
***
game_lineups (86822, 9)
********************************************************************************
***

head:

                    game_lineups_id  game_id  club_id             type number  \
0  baa0d6827dab4fab07d8dc1604e720a7  3606208      338  starting_lineup     15
1  1715ec342bf902522b69322a036a3f29  3606208      338  starting_lineup     71
2  d7c22834cb4c20efeddc527511feabad  3606208      338  starting_lineup     34
3  113289469cede9376049b78521f7b382  3606208      338  starting_lineup     25
4  e6abe553801b09bc623c3deb96acba17  3606208      338  starting_lineup     16

   player_id        player_name  team_captain      position
0     264372   Viktor Tsygankov             0  Right Winger
1      91931        Denys Boyko             0    Goalkeeper
2     505463   Oleksandr Syrota             0   Centre-Back
3     659089      Ilya Zabarnyi             0   Centre-Back
4     404842  Vitaliy Mykolenko             0     Left-Back

General information:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 86822 entries, 0 to 86821
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   game_lineups_id  86822 non-null  object
 1   game_id          86822 non-null  int64
 2   club_id          86822 non-null  int64
 3   type             86822 non-null  object
 4   number           86822 non-null  object
```

```
 5   player_id        86822 non-null  int64
 6   player_name      86822 non-null  object
 7   team_captain     86822 non-null  int64
 8   position         86822 non-null  object
dtypes: int64(4), object(5)
memory usage: 6.0+ MB
None

Number of uniques values:

game_lineups_id    86822
game_id             2144
club_id              972
type                   2
number               100
player_id          21942
player_name        21660
team_captain           2
position              17
dtype: int64

Statistical description:

            game_id        club_id     player_id  team_captain
count  8.682200e+04   86822.000000  8.682200e+04  86822.000000
mean   4.113804e+06    9505.120476  4.509931e+05      0.045611
std    4.844309e+04   19245.549845  2.733798e+05      0.208640
min    3.606208e+06       1.000000  3.159000e+03      0.000000
25%    4.094631e+06     418.000000  2.376585e+05      0.000000
50%    4.113131e+06    1148.000000  4.066400e+05      0.000000
75%    4.146925e+06    6673.000000  6.269540e+05      0.000000
max    4.196249e+06  112755.000000  1.195054e+06      1.000000

********************************************************************************
***
games (64293, 23)
********************************************************************************
***

head:

   game_id competition_id  season         round        date  home_club_id  \
0  2222597            RU1    2012   6. Matchday  2012-08-25          3725
1  2222627            RU1    2012   5. Matchday  2012-08-20          2696
2  2222658            RU1    2012  10. Matchday  2012-09-30          2410
3  2222664            RU1    2012   8. Matchday  2012-09-15           932
4  2222683            RU1    2012  12. Matchday  2012-10-22          2696

   away_club_id  home_club_goals  away_club_goals  home_club_position  ...  \
0           232              2.0              1.0                 2.0  ...
1          4128              0.0              2.0                11.0  ...
2           121              0.0              2.0                 2.0  ...
3          2698              1.0              0.0                 5.0  ...
4         12438              0.0              1.0                11.0  ...

        stadium attendance               referee  \
0  Akhmat-Arena    21700.0  Vladislav Bezborodov
1     Metallurg    11400.0         Sergey Ivanov
2  Arena Khimki    12000.0        Sergey Karasev
3     RZD Arena    11408.0        Sergey Karasev
4     Metallurg     7534.0     Timur Arslanbekov

                                                 url  home_club_formation  \
0  https://www.transfermarkt.co.uk/terek-grozny_s...                  NaN
```

```
1  https://www.transfermarkt.co.uk/krylya-sovetov...                  NaN
2  https://www.transfermarkt.co.uk/cska-moscow_di...                  NaN
3  https://www.transfermarkt.co.uk/lokomotiv-mosc...                  NaN
4  https://www.transfermarkt.co.uk/krylya-sovetov...                  NaN

  away_club_formation         home_club_name                   away_club_name  \
0                 NaN          Akhmat Grozny                   Spartak Moscow
1                 NaN  Krylya Sovetov Samara                       Amkar Perm
2                 NaN            CSKA Moscow                    Dynamo Moscow
3                 NaN       Lokomotiv Moscow                      Rubin Kazan
4                 NaN  Krylya Sovetov Samara  Volga Nizhniy Novgorod (- 2016)

  aggregate competition_type
0       2:1  domestic_league
1       0:2  domestic_league
2       0:2  domestic_league
3       1:0  domestic_league
4       0:1  domestic_league

[5 rows x 23 columns]

General information:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 64293 entries, 0 to 64292
Data columns (total 23 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   game_id                 64293 non-null  int64
 1   competition_id          64293 non-null  object
 2   season                  64293 non-null  int64
 3   round                   64293 non-null  object
 4   date                    64293 non-null  object
 5   home_club_id            64293 non-null  int64
 6   away_club_id            64293 non-null  int64
 7   home_club_goals         64291 non-null  float64
 8   away_club_goals         64291 non-null  float64
 9   home_club_position      45181 non-null  float64
 10  away_club_position      45181 non-null  float64
 11  home_club_manager_name  63554 non-null  object
 12  away_club_manager_name  63554 non-null  object
 13  stadium                 64078 non-null  object
 14  attendance              54810 non-null  float64
 15  referee                 63697 non-null  object
 16  url                     64293 non-null  object
 17  home_club_formation     2091 non-null   object
 18  away_club_formation     2111 non-null   object
 19  home_club_name          53256 non-null  object
 20  away_club_name          54407 non-null  object
 21  aggregate               64291 non-null  object
 22  competition_type        64293 non-null  object
dtypes: float64(5), int64(4), object(14)
memory usage: 11.3+ MB
None

Number of uniques values:

game_id                   64293
competition_id               43
season                       12
round                       116
date                       3362
away_club_id               2122
```

```
home_club_goals              16
away_club_goals              18
home_club_position           21
away_club_position           21
home_club_manager_name     4731
away_club_manager_name     4491
stadium                    2253
attendance                26875
referee                    2316
url                       64293
home_club_formation          39
away_club_formation          37
home_club_name              426
away_club_name              426
aggregate                   112
competition_type              4
dtype: int64

Statistical description:

            game_id        season   home_club_id   away_club_id  \
count  6.429300e+04  64293.000000   64293.000000   64293.000000
mean   3.001949e+06   2017.172849    4887.516028    4423.932419
std    5.478832e+05      3.304216   11355.003131   10223.416646
min    2.211607e+06   2012.000000       1.000000       2.000000
25%    2.512848e+06   2014.000000     354.000000     347.000000
50%    2.903799e+06   2017.000000     995.000000     989.000000
75%    3.433394e+06   2020.000000    3205.000000    3026.000000
max    4.196249e+06   2023.000000  112751.000000  112755.000000

       home_club_goals  away_club_goals  home_club_position  \
count     64291.000000     64291.000000        45181.000000
mean          1.592276         1.320107            9.275691
std           1.427064         1.361882            5.300881
min           0.000000         0.000000            1.000000
25%           1.000000         0.000000            5.000000
50%           1.000000         1.000000            9.000000
75%           2.000000         2.000000           14.000000
max          15.000000        19.000000           21.000000

       away_club_position    attendance
count        45181.000000  54810.000000
mean             9.458290  18030.075369
std              5.322696  17734.316215
min              1.000000      1.000000
25%              5.000000   4301.000000
50%              9.000000  12135.000000
75%             14.000000  26107.500000
max             21.000000  99354.000000

********************************************************************************
***
player_valuations (440663, 9)
********************************************************************************
***

head:

   player_id  last_season             datetime        date    dateweek  \
0       3132         2013  2003-12-09 00:00:00  2003-12-09  2003-12-08
1       6893         2012  2003-12-15 00:00:00  2003-12-15  2003-12-15
2         10         2015  2004-10-04 00:00:00  2004-10-04  2004-10-04
3         28         2015  2004-10-04 00:00:00  2004-10-04  2004-10-04
4         65         2015  2004-10-04 00:00:00  2004-10-04  2004-10-04
```

```
   market_value_in_eur  n  current_club_id player_club_domestic_competition_id
0               400000  1              126                                 TR1
1               900000  1              984                                 GB1
2              7000000  1              398                                 IT1
3              1500000  1               16                                  L1
4              8000000  1             1091                                 GR1

General information:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 440663 entries, 0 to 440662
Data columns (total 9 columns):
 #   Column                              Non-Null Count   Dtype
---  ------                              --------------   -----
 0   player_id                           440663 non-null  int64
 1   last_season                         440663 non-null  int64
 2   datetime                            440663 non-null  object
 3   date                                440663 non-null  object
 4   dateweek                            440663 non-null  object
 5   market_value_in_eur                 440663 non-null  int64
 6   n                                   440663 non-null  int64
 7   current_club_id                     440663 non-null  int64
 8   player_club_domestic_competition_id  440663 non-null  object
dtypes: int64(5), object(4)
memory usage: 30.3+ MB
None

Number of uniques values:

player_id                             28794
last_season                              12
datetime                               4817
date                                   4817
dateweek                                972
market_value_in_eur                     423
n                                         1
current_club_id                         424
player_club_domestic_competition_id      14
dtype: int64

Statistical description:

          player_id    last_season  market_value_in_eur         n  \
count  4.406630e+05  440663.000000         4.406630e+05  440663.0
mean   1.964113e+05    2018.762887         2.357557e+06       1.0
std    1.793622e+05       3.624305         6.603356e+06       0.0
min    1.000000e+01    2012.000000         1.000000e+04       1.0
25%    5.532200e+04    2016.000000         2.000000e+05       1.0
50%    1.407480e+05    2019.000000         5.000000e+05       1.0
75%    2.896450e+05    2022.000000         1.600000e+06       1.0
max    1.166093e+06    2023.000000         2.000000e+08       1.0

       current_club_id
count    440663.000000
mean       4041.891491
std        9508.375247
min           3.000000
25%         368.000000
50%        1010.000000
75%        2944.000000
max       83678.000000
```


..........**********

```
***
players (30302, 23)
*******************************************************************************
***

head:

   player_id first_name   last_name              name  last_season  \
0        598       Timo  Hildebrand   Timo Hildebrand         2014
1        670     Martin      Petrov     Martin Petrov         2012
2       1323     Martin     Amedick    Martin Amedick         2012
3       3195   Jermaine     Pennant  Jermaine Pennant         2013
4       3259     Damien        Duff       Damien Duff         2013

   current_club_id       player_code country_of_birth city_of_birth  \
0               24   timo-hildebrand          Germany         Worms
1              714     martin-petrov         Bulgaria        Vratsa
2               24    martin-amedick          Germany     Paderborn
3              512  jermaine-pennant          England    Nottingham
4              931       damien-duff          Ireland    Ballyboden

  country_of_citizenship  ...   foot height_in_cm market_value_in_eur  \
0                Germany  ...    NaN          NaN                 NaN
1               Bulgaria  ...    NaN          NaN                 NaN
2                Germany  ...    NaN          NaN                 NaN
3                England  ...  right        173.0                 NaN
4                Ireland  ...   left        177.0                 NaN

  highest_market_value_in_eur  contract_expiration_date  agent_name  \
0                  10000000.0                      NaN         NaN
1                  12000000.0                      NaN         IFM
2                   2750000.0                      NaN         NaN
3                  10500000.0                      NaN  Andrew Sky
4                  17000000.0                      NaN         NaN

                                        image_url  \
0  https://img.a.transfermarkt.technology/portrai...
1  https://img.a.transfermarkt.technology/portrai...
2  https://img.a.transfermarkt.technology/portrai...
3  https://img.a.transfermarkt.technology/portrai...
4  https://img.a.transfermarkt.technology/portrai...

                                             url  \
0  https://www.transfermarkt.co.uk/timo-hildebran...
1  https://www.transfermarkt.co.uk/martin-petrov/...
2  https://www.transfermarkt.co.uk/martin-amedick...
3  https://www.transfermarkt.co.uk/jermaine-penna...
4  https://www.transfermarkt.co.uk/damien-duff/pr...

  current_club_domestic_competition_id       current_club_name
0                                   L1     Eintracht Frankfurt
1                                  ES1  RCD Espanyol Barcelona
2                                   L1     Eintracht Frankfurt
3                                  GB1              Stoke City
4                                  GB1               Fulham FC

[5 rows x 23 columns]

General information:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30302 entries, 0 to 30301
```



```
 #   Column                                Non-Null Count  Dtype
```

```
---  ------                               --------------  -----
 0   player_id                            30302 non-null  int64
 1   first_name                           28337 non-null  object
 2   last_name                            30302 non-null  object
 3   name                                 30302 non-null  object
 4   last_season                          30302 non-null  int64
 5   current_club_id                      30302 non-null  int64
 6   player_code                          30302 non-null  object
 7   country_of_birth                     27613 non-null  object
 8   city_of_birth                        28099 non-null  object
 9   country_of_citizenship               29759 non-null  object
 10  date_of_birth                        30255 non-null  object
 11  sub_position                         30130 non-null  object
 12  position                             30302 non-null  object
 13  foot                                 27913 non-null  object
 14  height_in_cm                         28204 non-null  float64
 15  market_value_in_eur                  19383 non-null  float64
 16  highest_market_value_in_eur          28981 non-null  float64
 17  contract_expiration_date             18835 non-null  object
 18  agent_name                           14941 non-null  object
 19  image_url                            30302 non-null  object
 20  url                                  30302 non-null  object
 21  current_club_domestic_competition_id  30302 non-null  object
 22  current_club_name                    30302 non-null  object
dtypes: float64(3), int64(3), object(17)
memory usage: 5.3+ MB
None

Number of uniques values:

player_id                               30302
first_name                               6551
last_name                               22310
name                                    29662
last_season                                12
current_club_id                           424
player_code                             29628
country_of_birth                          184
city_of_birth                            8184
country_of_citizenship                    180
date_of_birth                            8892
sub_position                               13
position                                    5
foot                                        3
height_in_cm                               50
market_value_in_eur                       125
highest_market_value_in_eur               204
contract_expiration_date                   98
agent_name                               2622
image_url                               24683
url                                     30302
current_club_domestic_competition_id       14
current_club_name                         424
dtype: int64

Statistical description:

          player_id   last_season  current_club_id  height_in_cm  \
count  3.030200e+04  30302.000000     30302.000000  28204.000000
mean   3.112814e+05   2018.768926      4366.055574    182.234577
std    2.502577e+05      3.654540     10056.373140      6.833916
min    1.000000e+01   2012.000000         3.000000     18.000000
[illegible]  [illegible]  [illegible]  403.000000    178.000000
50%    2.378345e+05   2019.000000      1071.000000    182.000000
```

```
75%    4.655942e+05   2022.000000      3008.000000    187.000000
max    1.186012e+06   2023.000000     83678.000000    207.000000

       market_value_in_eur  highest_market_value_in_eur
count         1.938300e+04                 2.898100e+04
mean          2.234721e+06                 3.571396e+06
std           7.340683e+06                 9.352255e+06
min           1.000000e+04                 1.000000e+04
25%           1.750000e+05                 2.500000e+05
50%           3.500000e+05                 7.500000e+05
75%           1.000000e+06                 2.700000e+06
max           1.800000e+08                 2.000000e+08
```

In [4]:

```
# Initial enquiry into missing values

for i, df in enumerate(dataframes):
    print("\n**********************************************************************
    print(file_names[i], df.shape)
    print("**********************************************************************
    print(df.isnull().sum())
    print(f"\nThe total number of missing values in {file_names[i]} dataframe are:
    print(f"\nThat is equal to ", round(((df.isna().sum().sum()) / (df.size)) * 100
```

```
********************************************************************************
***
appearances (1485697, 13)
********************************************************************************
***

appearance_id               0
game_id                     0
player_id                   0
player_club_id              0
player_current_club_id      0
date                        0
player_name               324
competition_id              0
yellow_cards                0
red_cards                   0
goals                       0
assists                     0
minutes_played              0
dtype: int64

The total number of missing values in appearances dataframe are:  324 out of 19314
061.

That is equal to  0.0 percent.

********************************************************************************
***
club_games (128586, 11)
********************************************************************************
***

game_id                     0
club_id                     0
own_goals                   4
own_position            38224
own_manager_name         1478
opponent_id                 0
opponent_goals              4
opponent_position       38224
opponent_manager_name    1478
hosting                     0
is_win                      0
dtype: int64

The total number of missing values in club_games dataframe are:  79412 out of 1414
446.

That is equal to  5.61 percent.

********************************************************************************
***
clubs (426, 16)
********************************************************************************
***

club_id                     0
club_code                   0
name                        0
domestic_competition_id     0
total_market_value        426
squad_size                  0
foreigners_number           0
```

```
foreigners_percentage       47
national_team_players        0
stadium_name                 0
stadium_seats                0
net_transfer_record          0
coach_name                 426
last_season                  0
url                          0
dtype: int64

The total number of missing values in clubs dataframe are:  937 out of 6816.

That is equal to  13.75 percent.

********************************************************************************
***
competitions (43, 10)
********************************************************************************
***

competition_id          0
competition_code        0
name                    0
sub_type                0
type                    0
country_id              0
country_name            7
domestic_league_code    7
confederation           0
url                     0
dtype: int64

The total number of missing values in competitions dataframe are:  14 out of 430.

That is equal to  3.26 percent.

********************************************************************************
***
game_events (652010, 10)
********************************************************************************
***

game_event_id            0
date                     0
game_id                  0
minute                   0
type                     0
club_id                  0
player_id                0
description         334232
player_in_id        238846
player_assist_id    622852
dtype: int64

The total number of missing values in game_events dataframe are:  1195930 out of 6
520100.

That is equal to  18.34 percent.

********************************************************************************
***
game_lineups (86822, 9)
********************************************************************************
```

```
game_lineups_id    0
game_id            0
club_id            0
type               0
number             0
player_id          0
player_name        0
team_captain       0
position           0
dtype: int64

The total number of missing values in game_lineups dataframe are:  0 out of 78139
8.

That is equal to  0.0 percent.

********************************************************************************
***
games (64293, 23)
********************************************************************************
***

game_id                      0
competition_id               0
season                       0
round                        0
date                         0
home_club_id                 0
away_club_id                 0
home_club_goals              2
away_club_goals              2
home_club_position       19112
away_club_position       19112
home_club_manager_name     739
away_club_manager_name     739
stadium                    215
attendance                9483
referee                    596
url                          0
home_club_formation      62202
away_club_formation      62182
home_club_name           11037
away_club_name            9886
aggregate                    2
competition_type             0
dtype: int64

The total number of missing values in games dataframe are:  195309 out of 1478739.

That is equal to  13.21 percent.

********************************************************************************
***
player_valuations (440663, 9)
********************************************************************************
***

player_id                          0
last_season                        0
datetime                           0
date                               0
                                   0
market_value_in_eur                0
```

```
n                                       0
current_club_id                         0
player_club_domestic_competition_id     0
dtype: int64

The total number of missing values in player_valuations dataframe are:  0 out of 3
965967.

That is equal to  0.0 percent.

********************************************************************************
***
players (30302, 23)
********************************************************************************
***

player_id                                 0
first_name                             1965
last_name                                 0
name                                      0
last_season                               0
current_club_id                           0
player_code                               0
country_of_birth                       2689
city_of_birth                          2203
country_of_citizenship                  543
date_of_birth                            47
sub_position                            172
position                                  0
foot                                   2389
height_in_cm                           2098
market_value_in_eur                   10919
highest_market_value_in_eur            1321
contract_expiration_date              11467
agent_name                            15361
image_url                                 0
url                                       0
current_club_domestic_competition_id      0
current_club_name                         0
dtype: int64

The total number of missing values in players dataframe are:  51174 out of 696946.

That is equal to  7.34 percent.
```

As we can see above, some dataframes contain obsevations with as much as 18% missing values. As a general rule of thumb, if a feature contains more than 50% missing values, it may be justified in removing that column. If an observation contains ober 20% missing values it may be justified in removing that row. All depending on the use missing values, use case and ultimate project requirements. At a glance, many of the missing values above are from features that will not be required in evaluating player market values. They will be retained for now but will need to be addressed following selecting and merging the required datasets for modelling.

In [5]:
```
# Removing columns with over 50% of mssing values

for df in dataframes:
    df.dropna(thresh = df.shape[0] * 0.5, axis = 1, inplace=True)
```



```
for i, df in enumerate(dataframes):
```

```
    print("\n**********************************************************************
    print(file_names[i], df.shape)
    print("**********************************************************************
    print(df.isnull().sum())
    print(f"\nThe total number of missing values in {file_names[i]} dataframe are:
    print(f"\nThat is equal to ", round(((df.isna().sum().sum()) / (df.size)) * 100
```

```
********************************************************************************
***
appearances (1485697, 13)
********************************************************************************
***

appearance_id               0
game_id                     0
player_id                   0
player_club_id              0
player_current_club_id      0
date                        0
player_name               324
competition_id              0
yellow_cards                0
red_cards                   0
goals                       0
assists                     0
minutes_played              0
dtype: int64

The total number of missing values in appearances dataframe are:  324 out of 19314
061.

That is equal to  0.0 percent.

********************************************************************************
***
club_games (128586, 11)
********************************************************************************
***

game_id                     0
club_id                     0
own_goals                   4
own_position            38224
own_manager_name         1478
opponent_id                 0
opponent_goals              4
opponent_position       38224
opponent_manager_name    1478
hosting                     0
is_win                      0
dtype: int64

The total number of missing values in club_games dataframe are:  79412 out of 1414
446.

That is equal to  5.61 percent.

********************************************************************************
***
clubs (426, 14)
********************************************************************************
***

club_id                     0
club_code                   0
name                        0
domestic_competition_id     0
squad_size                  0
average_age                38
foreigners_percentage      47
```

```
national_team_players       0
stadium_name                0
stadium_seats               0
net_transfer_record         0
last_season                 0
url                         0
dtype: int64

The total number of missing values in clubs dataframe are:  85 out of 5964.

That is equal to  1.43 percent.

********************************************************************************
***
competitions (43, 10)
********************************************************************************
***

competition_id          0
competition_code        0
name                    0
sub_type                0
type                    0
country_id              0
country_name            7
domestic_league_code    7
confederation           0
url                     0
dtype: int64

The total number of missing values in competitions dataframe are:  14 out of 430.

That is equal to  3.26 percent.

********************************************************************************
***
game_events (652010, 8)
********************************************************************************
***

game_event_id         0
date                  0
game_id               0
minute                0
type                  0
club_id               0
player_id             0
player_in_id     238846
dtype: int64

The total number of missing values in game_events dataframe are:  238846 out of 52
16080.

That is equal to  4.58 percent.

********************************************************************************
***
game_lineups (86822, 9)
********************************************************************************
***

game_lineups_id    0
club_id            0
```

```
type               0
number             0
player_id          0
player_name        0
team_captain       0
position           0
dtype: int64

The total number of missing values in game_lineups dataframe are:  0 out of 78139
8.

That is equal to  0.0 percent.

********************************************************************************
***
games (64293, 21)
********************************************************************************
***

game_id                       0
competition_id                0
season                        0
round                         0
date                          0
home_club_id                  0
away_club_id                  0
home_club_goals               2
away_club_goals               2
home_club_position        19112
away_club_position        19112
home_club_manager_name      739
away_club_manager_name      739
stadium                     215
attendance                 9483
referee                     596
url                           0
home_club_name            11037
away_club_name             9886
aggregate                     2
competition_type              0
dtype: int64

The total number of missing values in games dataframe are:  70925 out of 1350153.

That is equal to  5.25 percent.

********************************************************************************
***
player_valuations (440663, 9)
********************************************************************************
***

player_id                             0
last_season                           0
datetime                              0
date                                  0
dateweek                              0
market_value_in_eur                   0
n                                     0
current_club_id                       0
player_club_domestic_competition_id   0
dtype: int64

The total number of missing values in player_valuations dataframe are:  0 out of 3
```

```
965967.

That is equal to  0.0 percent.

********************************************************************************
***
players (30302, 22)
********************************************************************************
***

player_id                                  0
first_name                              1965
last_name                                  0
name                                       0
last_season                                0
current_club_id                            0
player_code                                0
country_of_birth                        2689
city_of_birth                           2203
country_of_citizenship                   543
date_of_birth                             47
sub_position                             172
position                                   0
foot                                    2389
height_in_cm                            2098
market_value_in_eur                    10919
highest_market_value_in_eur             1321
contract_expiration_date               11467
image_url                                  0
url                                        0
current_club_domestic_competition_id       0
current_club_name                          0
dtype: int64

The total number of missing values in players dataframe are:  35813 out of 666644.

That is equal to  5.37 percent.
```

In [6]:
```python
# Checking for duplicate entries and displaying the results using ptint()

for i, df in enumerate(dataframes):
    duplicateRows = df[df.duplicated()]
    print("\n**************************************************************************
    print(f"\nThe total number of duplicate rows in {file_names[i]} dataframe is {d
    print(f"\nThat is equal to ", round(((duplicateRows.value_counts().sum()) / (df
```

```
**********************************************************************************
***

The total number of duplicate rows in appearances dataframe is 0 out of 19314061.

That is equal to  0.0 percent.

**********************************************************************************
***

The total number of duplicate rows in club_games dataframe is 0 out of 1414446.

That is equal to  0.0 percent.

**********************************************************************************
***

The total number of duplicate rows in clubs dataframe is 0 out of 5964.

That is equal to  0.0 percent.

**********************************************************************************
***

The total number of duplicate rows in competitions dataframe is 0 out of 430.

That is equal to  0.0 percent.

**********************************************************************************
***

The total number of duplicate rows in game_events dataframe is 0 out of 5216080.

That is equal to  0.0 percent.

**********************************************************************************
***

The total number of duplicate rows in game_lineups dataframe is 0 out of 781398.

That is equal to  0.0 percent.

**********************************************************************************
***

The total number of duplicate rows in games dataframe is 0 out of 1350153.

That is equal to  0.0 percent.

**********************************************************************************
***

The total number of duplicate rows in player_valuations dataframe is 0 out of 3965
967.

That is equal to  0.0 percent.

**********************************************************************************
***

The total number of duplicate rows in players dataframe is 0 out of 666644.
```

As of now, there are now duplicate entries within any of the datasets

Displaying a Profile Report for each dataset

In [7]:
```
from ydata_profiling import ProfileReport #importing profile report attribute
```

In [ ]:
```
ProfileReport(appearances_df)# generating a profile report of the dataset
```

In [ ]:
```
ProfileReport(club_games_df)
```

In [ ]:
```
ProfileReport(clubs_df)
```

In [ ]:
```
ProfileReport(competitions_df)
```

In [ ]:
```
ProfileReport(game_events_df)
```

In [ ]:
```
ProfileReport(game_lineups_df)
```

In [ ]:
```
ProfileReport(games_df)
```

In [ ]:
```
ProfileReport(player_valuations_df)
```

In [8]:
```
ProfileReport(players_df)
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]
Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]
Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
```

Overview

Dataset statistics

Number of variables	22
Number of observations	30302
Missing cells	35813
Missing cells (%)	5.4%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	5.1 MiB
Average record size in memory	176.0 B

Variable types

Numeric	6
Text	10
DateTime	2
Categorical	4

Alerts

`player_id` is highly overall correlated with `last_season`	**High correlation**

Out[8]:

After evaluating the findings of the profile reports it is evident that there are two outliers within the players height column in the players_df with a value of 18cm. The mean and std deviation for the column is 182.5cm and 6.9cm respectively. The value is more than likey missing a digit.

In [9]:
```
# locating the outliers
players_df[players_df['height_in_cm'] == 18]
```

Out[9]:

	player_id	first_name	last_name	name	last_season	current_club_id	player_code	c
6846	592398	Aïssa	Boudechicha	Aïssa Boudechicha	2021	40	aissa-boudechicha	
8716	628490	Genar	Fornés	Genar Fornés	2021	2687	genar-fornes	

2 rows × 22 columns

Options for handling outliers include removing the entry, imputing the height with the mean or mode, or replacing it with the correct value sourced from reliable source. I have decided to replace the error value for height with correct value sourced from transfermarkt (www.transfermarkt.co.uk, n.d.) (www.transfermarkt.co.uk, n.d.).

In [10]:
```
players_df[players_df['name'] == 'Genar Fornés']
```

Out[10]:

	player_id	first_name	last_name	name	last_season	current_club_id	player_code	country_
8716	628490	Genar	Fornés	Genar Fornés	2021	2687	genar-fornes	

1 rows × 22 columns

In [11]:
```
# Replacing the incorrect height with the correct value
players_df.loc[players_df['name'] == 'Genar Fornés', 'height_in_cm'] = 180

players_df.loc[players_df['name'] == 'Aïssa Boudechicha', 'height_in_cm'] = 180

# Both players are the same height so alternatively the code below could be used to

#players_df['height_in_cm'].replace(18, 180, inplace=True)
```

In [12]:
```
# Confirming new height
players_df[players_df['name'] == 'Genar Fornés']
```

Out[12]:

	player_id	first_name	last_name	name	last_season	current_club_id	player_code	country_
8716	628490	Genar	Fornés	Genar Fornés	2021	2687	genar-fornes	

1 rows × 22 columns

In [13]:
```
# Confirming new height
players_df[players_df['name'] == 'Aïssa Boudechicha']
```

Out[13]:

	player_id	first_name	last_name	name	last_season	current_club_id	player_code	c
6846	592398	Aïssa	Boudechicha	Aïssa Boudechicha	2021	40	aissa-boudechicha	

1 rows × 22 columns

In [14]:
```
# Confirming that the outliers values for height are corrected
players_df[players_df['height_in_cm'] == 18]
```

Out[14]:

	player_id	first_name	last_name	name	last_season	current_club_id	player_code	country_of_bir

0 rows × 22 columns

Exploring the player 'market_value_in_eur' feature...

..With the hope of merging values from the 'player_valuations_df' and 'players_df' to replace missing values

In [15]:
```
player_valuations_df.nunique()
```

Out[15]:
```
player_id                          28794
last_season                           12
datetime                            4817
date                                4817
dateweek                             972
market_value_in_eur                  423
n                                      1
current_club_id                      424
player_club_domestic_competition_id   14
dtype: int64
```

In [16]:
```
player_valuations_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 440663 entries, 0 to 440662
Data columns (total 9 columns):
 #   Column                             Non-Null Count   Dtype 
---  ------                             --------------   ----- 
 0   player_id                          440663 non-null  int64 
 1   last_season                        440663 non-null  int64 
 2   datetime                           440663 non-null  object
 3   date                               440663 non-null  object
 4   dateweek                           440663 non-null  object
 5   market_value_in_eur                440663 non-null  int64 
 6   n                                  440663 non-null  int64 
 7   current_club_id                    440663 non-null  int64 
 8   player_club_domestic_competition_id  440663 non-null  object
dtypes: int64(5), object(4)
memory usage: 30.3+ MB
```

In [17]:
```
players_df.nunique()
```

Out[17]:
```
player_id                              30302
first_name                              6551
last_name                              22310
name                                   29662
last_season                               12
current_club_id                          424
player_code                            29628
country_of_birth                         184
city_of_birth                           8184
country_of_citizenship                   180
date_of_birth                           8892
sub_position                              13
position                                   5
foot                                       3
height_in_cm                              49
market_value_in_eur                      125
highest_market_value_in_eur              204
contract_expiration_date                  98
image_url                              24683
url                                    30302
current_club_domestic_competition_id      14
current_club_name                        424
dtype: int64
```

In [18]:
```
players_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30302 entries, 0 to 30301
Data columns (total 22 columns):
 #   Column                                 Non-Null Count  Dtype
---  ------                                 --------------  -----
 0   player_id                              30302 non-null  int64
 1   first_name                             28337 non-null  object
 2   last_name                              30302 non-null  object
 3   name                                   30302 non-null  object
 4   last_season                            30302 non-null  int64
 5   current_club_id                        30302 non-null  int64
 6   player_code                            30302 non-null  object
 7   country_of_birth                       27613 non-null  object
 8   city_of_birth                          28099 non-null  object
 9   country_of_citizenship                 29759 non-null  object
 10  date_of_birth                          30255 non-null  object
 11  sub_position                           30130 non-null  object
 12  position                               30302 non-null  object
 13  foot                                   27913 non-null  object
 14  height_in_cm                           28204 non-null  float64
 15  market_value_in_eur                    19383 non-null  float64
 16  highest_market_value_in_eur            28981 non-null  float64
 17  contract_expiration_date               18835 non-null  object
 18  image_url                              30302 non-null  object
 19  url                                    30302 non-null  object
 20  current_club_domestic_competition_id   30302 non-null  object
 21  current_club_name                      30302 non-null  object
dtypes: float64(3), int64(3), object(16)
memory usage: 5.1+ MB
```

In [19]:
```
market_values_A = player_valuations_df[player_valuations_df['last_season'] == 2023]
```

In [20]:
```
market_values_A.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 101275 entries, 186 to 440661
Data columns (total 9 columns):
 #   Column                                Non-Null Count   Dtype
---  ------                                --------------   -----
 0   player_id                             101275 non-null  int64
 1   last_season                           101275 non-null  int64
 2   datetime                              101275 non-null  object
 3   date                                  101275 non-null  object
 4   dateweek                              101275 non-null  object
 5   market_value_in_eur                   101275 non-null  int64
 6   n                                     101275 non-null  int64
 7   current_club_id                       101275 non-null  int64
 8   player_club_domestic_competition_id   101275 non-null  object
dtypes: int64(5), object(4)
memory usage: 7.7+ MB
```

In [21]:
```
market_values_A.head()
```

Out[21]:

	player_id	last_season	datetime	date	dateweek	market_value_in_eur	n	current_club_id	p
186	3333	2023	2004-10-04 00:00:00	2004-10-04	2004-10-04	7500000	1	1237	
343	4391	2023	2004-10-04 00:00:00	2004-10-04	2004-10-04	1300000	1	383	
743	8246	2023	2004-10-04 00:00:00	2004-10-04	2004-10-04	50000	1	3	
782	9500	2023	2004-10-04 00:00:00	2004-10-04	2004-10-04	750000	1	903	
884	12029	2023	2004-10-04 00:00:00	2004-10-04	2004-10-04	1500000	1	1421	

In [22]:
```
pd.set_option('display.max_row', None) # to display max 100 rows
```

In [23]:
```
market_values_A['dateweek'].value_counts()
```

Out[23]:

dateweek	
2020-04-06	3523
2021-12-27	1967
2022-11-07	1816
2023-06-12	1815
2023-06-19	1769
2022-05-30	1686
2018-12-17	1601
2019-12-16	1432
2022-06-06	1391
2022-10-31	1381
2019-06-10	1294
2020-10-12	1263
2021-06-07	1235
2019-06-03	1198
2021-05-31	1168
2023-06-26	1152
2021-12-20	1068
2020-12-28	949
2018-05-28	925
2015-06-29	923
2019-12-09	905
2016-07-11	900
2018-06-04	868
2022-01-03	854
2021-01-04	816
2017-06-26	791
2023-06-05	765
2021-03-15	752
2017-06-05	739
2021-10-11	725
2018-01-01	657
2021-02-08	651
2021-05-17	632
2019-06-24	607
2021-10-04	582
2022-09-12	579
2022-06-27	576
2022-06-13	575
2022-10-24	572
2021-12-13	559
2023-03-13	552
2021-05-24	550
2020-07-27	537
2020-12-21	532
2020-10-05	521
2022-06-20	514
2022-03-28	504
2021-06-28	503
2023-03-20	492
2022-12-19	471
2016-02-01	468
2022-12-05	451
2017-12-25	450
2019-12-23	449
2019-12-30	434
2020-08-03	430
2021-06-21	428
2017-01-16	421
2019-09-09	417
2018-12-24	403
2017-06-19	394
2017-12-18	386

2020-08-24	367
2023-03-27	361
2017-05-29	361
2022-09-19	345
2016-07-18	343
2019-01-07	340
2016-08-01	332
2020-08-17	332
2020-01-06	316
2022-11-21	312
2018-06-11	310
2014-02-10	306
2016-02-22	304
2021-03-08	304
2013-01-07	302
2017-01-23	297
2018-01-08	293
2017-02-20	292
2015-01-05	288
2022-04-11	284
2022-05-16	281
2022-12-12	279
2019-06-17	272
2023-05-22	266
2022-12-26	266
2016-05-30	264
2020-07-20	264
2017-01-02	263
2018-01-22	262
2018-12-31	262
2016-01-04	258
2023-04-03	257
2015-06-01	254
2017-02-06	250
2020-10-19	250
2016-02-08	246
2018-02-26	246
2021-01-11	243
2014-08-11	240
2020-03-09	238
2021-01-25	236
2023-05-29	234
2020-06-29	233
2018-08-06	233
2021-06-14	231
2019-03-04	223
2016-12-19	221
2019-02-18	220
2021-11-01	220
2013-06-24	219
2015-02-09	219
2018-06-25	216
2018-01-15	209
2021-12-06	209
2014-01-13	209
2015-02-02	202
2021-01-18	199
2013-06-17	199
2022-03-14	196
2022-09-05	196
2014-06-30	192
2022-01-10	190
2018-03-19	187

2023-02-27	182
2019-02-25	180
2020-02-03	180
2015-01-12	177
2020-12-14	174
2020-02-10	173
2020-08-31	173
2018-02-05	172
2018-12-10	172
2021-03-22	170
2018-06-18	170
2018-05-21	169
2013-06-10	167
2019-09-23	167
2022-11-14	166
2011-06-27	165
2014-07-14	165
2015-09-21	164
2020-11-23	163
2015-06-15	163
2017-10-23	160
2015-02-16	160
2018-10-22	160
2019-01-21	158
2022-09-26	157
2014-08-04	156
2020-03-30	156
2015-12-14	153
2019-09-30	153
2019-01-14	151
2015-08-31	148
2018-10-15	148
2019-05-20	147
2014-01-20	146
2018-10-29	146
2020-11-09	145
2021-11-22	144
2013-05-27	143
2017-10-09	141
2016-02-15	141
2017-01-30	141
2016-01-18	141
2022-05-09	140
2020-02-24	139
2023-07-03	137
2020-07-06	136
2012-06-18	136
2014-07-07	135
2018-09-24	133
2019-10-07	133
2021-04-05	132
2022-05-23	132
2013-07-01	131
2023-09-18	130
2020-09-14	130
2016-08-08	130
2015-02-23	129
2018-12-03	128
2012-01-30	127
2016-06-27	126
2015-01-19	125
2017-01-09	124
2017-03-20	121

2013-10-07	121
2014-01-06	120
2014-06-23	120
2017-08-14	120
2016-12-26	120
2016-10-24	119
2015-10-12	119
2016-11-28	118
2011-01-31	117
2012-12-31	117
2016-05-23	117
2018-02-12	116
2012-07-02	115
2022-11-28	114
2016-08-29	114
2011-12-19	114
2015-06-22	113
2013-07-22	112
2017-06-12	112
2021-02-22	111
2017-02-27	111
2019-09-16	109
2017-10-16	109
2013-01-14	108
2019-10-14	108
2017-03-27	106
2016-10-10	104
2018-07-02	102
2017-12-04	101
2018-10-08	101
2018-04-02	101
2016-06-13	100
2018-07-09	100
2014-07-21	100
2017-09-04	100
2012-01-09	99
2014-09-29	99
2016-07-25	98
2010-12-27	97
2014-03-24	97
2018-09-10	95
2012-06-25	95
2019-03-11	93
2016-09-26	93
2016-03-14	93
2019-07-01	90
2021-09-13	90
2012-08-06	90
2023-04-10	89
2020-06-22	88
2018-09-03	86
2014-01-27	85
2015-11-23	85
2018-05-14	85
2015-06-08	85
2011-01-10	84
2016-10-31	84
2014-07-28	83
2011-12-26	83
2016-04-18	83
2018-02-19	83
2015-12-28	83
2015-01-26	82

2011-08-08	82
2020-08-10	81
2017-09-18	81
2011-05-30	81
2014-12-22	80
2021-04-12	80
2019-12-02	77
2019-02-11	76
2020-09-07	76
2012-02-06	76
2014-11-10	76
2021-03-29	76
2016-06-20	76
2015-11-16	76
2012-05-21	76
2020-11-02	75
2018-08-20	74
2022-01-17	74
2012-01-23	73
2012-06-04	73
2011-01-17	72
2023-09-11	72
2013-01-28	71
2022-10-03	71
2013-06-03	71
2020-11-30	71
2017-05-08	70
2014-11-17	70
2020-10-26	69
2013-12-30	69
2010-07-26	69
2018-03-26	69
2017-02-13	69
2019-11-25	68
2019-05-27	67
2015-08-24	67
2017-07-10	66
2018-07-16	66
2016-04-04	66
2019-08-19	66
2010-08-23	66
2021-10-18	65
2014-11-03	64
2018-11-26	64
2018-04-23	63
2018-10-01	63
2012-07-16	63
2011-06-13	63
2017-10-02	61
2019-03-18	61
2012-01-02	60
2010-10-18	60
2012-09-17	60
2022-03-07	60
2013-12-23	60
2020-07-13	60
2014-02-03	60
2020-01-13	60
2011-05-23	59
2016-10-17	59
2019-08-05	59
2020-12-07	58
2015-03-02	58

2022-04-04	57
2013-09-16	57
2014-08-25	57
2013-02-25	56
2014-10-06	56
2011-08-29	56
2013-01-21	55
2012-05-28	54
2021-07-12	54
2015-11-09	54
2016-01-11	53
2015-04-06	52
2015-07-13	52
2015-05-04	52
2013-02-04	52
2016-03-28	52
2010-07-12	52
2019-07-22	52
2018-08-13	52
2013-07-08	51
2014-12-15	50
2021-10-25	50
2011-06-20	50
2021-07-05	50
2012-06-11	50
2023-07-24	49
2015-07-27	48
2014-10-20	48
2011-07-25	48
2014-02-17	47
2015-10-19	47
2009-08-03	46
2015-09-14	46
2023-07-10	46
2013-08-05	46
2019-02-04	46
2014-08-18	46
2017-11-13	45
2011-07-18	45
2015-11-02	45
2023-01-02	45
2022-02-21	44
2018-03-12	44
2021-08-30	44
2018-09-17	44
2013-11-04	44
2010-01-11	44
2019-08-12	44
2012-12-24	42
2010-01-25	42
2015-07-20	42
2021-07-26	42
2011-07-04	41
2020-11-16	41
2010-08-16	41
2012-07-23	41
2021-11-08	41
2021-11-29	40
2021-09-20	40
2010-08-09	40
2021-08-02	40
2019-04-15	39
2017-11-20	39

2013-05-06	39
2015-11-30	38
2012-01-16	38
2011-10-17	38
2022-08-01	38
2019-10-28	38
2015-05-25	38
2021-09-06	38
2017-04-17	38
2012-11-19	38
2008-02-04	37
2022-10-10	37
2008-06-30	37
2012-11-05	37
2014-03-03	36
2017-07-03	36
2011-07-11	36
2013-09-23	36
2017-07-31	35
2011-02-07	35
2021-08-09	35
2012-12-03	35
2012-03-26	35
2018-05-07	35
2015-03-23	35
2016-04-11	34
2009-08-31	34
2019-11-11	34
2012-07-30	34
2015-12-21	34
2016-06-06	34
2015-09-28	34
2010-01-18	34
2016-10-03	33
2017-03-13	33
2020-02-17	33
2010-04-05	33
2013-08-26	32
2022-04-18	32
2022-07-25	31
2012-03-05	31
2014-03-31	31
2009-01-19	31
2019-07-29	31
2021-04-19	31
2012-02-27	30
2013-02-11	30
2016-09-12	30
2011-12-05	30
2015-12-07	29
2012-08-20	29
2010-12-06	29
2012-10-22	29
2015-03-16	29
2011-09-19	29
2015-08-10	29
2009-12-07	29
2010-05-17	28
2009-09-21	28
2015-09-07	28
2017-05-01	28
2014-03-10	28
2009-01-05	28

2012-09-24	28
2017-05-22	28
2012-08-13	27
2008-10-13	27
2008-07-14	27
2009-12-21	27
2012-12-17	27
2009-06-29	27
2014-05-05	27
2019-04-22	27
2010-06-28	27
2018-04-09	27
2019-09-02	26
2019-11-18	26
2018-01-29	26
2008-12-29	26
2013-03-25	26
2016-12-05	26
2009-06-22	26
2022-08-29	26
2009-11-23	26
2009-05-18	25
2009-02-02	25
2009-06-08	25
2013-12-02	25
2014-06-16	25
2009-07-20	25
2010-08-02	25
2021-07-19	25
2018-07-30	25
2013-11-25	25
2021-02-15	25
2015-10-26	25
2009-02-23	24
2018-08-27	24
2011-08-22	24
2012-07-09	24
2009-10-19	24
2019-03-25	24
2010-11-15	24
2014-09-01	24
2010-02-15	24
2015-10-05	24
2009-11-30	24
2012-04-09	23
2012-09-03	23
2009-01-26	23
2014-09-08	23
2017-11-06	23
2014-12-01	23
2009-10-05	22
2016-11-14	22
2008-12-08	22
2007-07-23	22
2011-12-12	22
2010-01-04	22
2017-12-11	22
2014-05-12	22
2010-04-12	22
2012-02-20	22
2011-02-28	21
2011-09-26	21
2012-10-29	21

2010-11-22	21
2018-04-30	21
2013-03-04	21
2017-09-25	20
2009-08-17	20
2020-06-15	20
2008-12-15	20
2017-10-30	20
2004-10-04	20
2014-09-22	20
2012-08-27	20
2010-06-07	20
2021-09-27	20
2010-09-27	20
2011-08-15	20
2011-02-21	20
2007-08-27	19
2010-02-01	19
2019-04-29	19
2021-05-10	18
2009-06-15	18
2014-10-27	18
2014-09-15	18
2012-11-26	18
2010-06-21	18
2016-08-22	18
2008-10-20	18
2022-05-02	18
2018-11-05	18
2011-10-10	18
2013-08-12	18
2014-02-24	18
2016-01-25	18
2020-03-23	18
2017-07-24	17
2017-07-17	17
2010-04-19	17
2010-05-31	17
2013-04-15	17
2010-03-01	17
2010-03-08	17
2015-08-17	17
2016-08-15	17
2017-11-27	16
2010-09-20	16
2020-01-20	16
2009-07-06	16
2020-01-27	16
2009-06-01	16
2009-11-09	16
2013-03-11	16
2010-09-06	16
2014-12-29	16
2019-04-01	16
2011-11-21	16
2008-01-07	16
2011-09-05	16
2013-04-01	15
2011-11-14	15
2017-04-10	15
2023-01-09	15
2010-05-24	15
2022-07-04	15

2012-03-19	15
2015-03-09	15
2010-07-19	15
2012-11-12	15
2011-04-04	14
2007-12-24	14
2008-08-18	14
2008-09-01	14
2007-06-18	14
2013-10-21	14
2018-11-19	14
2010-03-29	14
2022-02-07	14
2022-07-18	14
2022-02-14	14
2015-04-13	14
2010-05-03	14
2010-08-30	13
2015-07-06	13
2010-02-22	13
2010-10-04	13
2011-03-14	13
2023-08-21	13
2021-03-01	13
2007-12-10	13
2008-09-08	13
2012-04-02	13
2011-11-07	12
2013-08-19	12
2008-09-22	12
2016-09-05	12
2008-12-01	12
2014-10-13	12
2012-12-10	12
2013-04-22	12
2008-03-17	12
2008-04-07	12
2012-04-30	12
2013-05-20	11
2011-09-12	11
2009-02-16	11
2019-07-15	11
2007-10-08	11
2011-05-09	11
2010-02-08	11
2011-01-24	11
2010-10-11	11
2012-02-13	11
2023-01-30	11
2008-05-05	11
2017-05-15	11
2008-09-15	11
2013-10-28	11
2013-11-11	11
2019-07-08	10
2011-06-06	10
2023-08-07	10
2012-09-10	10
2008-10-06	10
2007-09-17	10
2013-05-13	10
2015-03-30	10
2009-05-25	10

2011-01-03	10
2009-07-13	10
2013-10-14	10
2015-08-03	10
2014-04-14	10
2013-07-29	10
2009-11-16	10
2007-08-06	10
2007-08-20	10
2007-09-03	10
2007-11-19	10
2008-01-21	10
2008-02-18	10
2016-03-21	10
2009-01-12	10
2010-07-05	10
2008-08-04	9
2018-11-12	9
2007-10-22	9
2018-07-23	9
2012-10-15	9
2023-02-20	9
2019-01-28	9
2010-03-22	9
2010-11-29	9
2008-01-28	9
2016-11-21	9
2014-06-09	9
2016-11-07	9
2009-09-07	9
2015-05-11	9
2009-10-26	9
2023-07-17	9
2008-11-24	9
2016-07-04	9
2018-03-05	9
2012-10-01	9
2010-04-26	8
2007-12-17	8
2023-08-28	8
2008-03-31	8
2011-02-14	8
2008-06-02	8
2019-08-26	8
2010-12-20	8
2009-04-27	8
2020-05-11	8
2009-03-02	8
2011-10-31	8
2011-04-25	8
2023-05-15	8
2013-03-18	8
2012-10-08	8
2009-12-14	8
2023-07-31	8
2008-08-25	8
2008-07-28	8
2008-07-21	8
2012-03-12	8
2019-05-06	8
2007-10-15	7
2023-02-13	7
2011-10-24	7

2011-03-21	7
2012-04-23	7
2008-11-03	7
2008-10-27	7
2011-04-18	7
2008-07-07	7
2010-12-13	7
2014-03-17	7
2008-06-09	7
2014-12-08	7
2008-11-17	7
2009-09-28	7
2008-02-25	7
2016-05-02	7
2009-11-02	7
2009-10-12	7
2010-03-15	6
2009-04-20	6
2017-08-28	6
2006-10-02	6
2017-08-07	6
2023-01-23	6
2022-10-17	6
2021-08-16	6
2011-03-07	6
2007-04-23	6
2007-09-10	6
2009-12-28	6
2010-09-13	6
2010-11-01	6
2016-05-09	6
2013-02-18	6
2011-11-28	6
2007-12-31	6
2012-04-16	6
2008-03-10	6
2006-09-25	5
2005-12-05	5
2021-11-15	5
2023-08-14	5
2022-07-11	5
2014-04-21	5
2008-06-23	5
2009-05-11	5
2008-04-21	5
2023-05-08	5
2010-10-25	5
2008-01-14	5
2009-03-09	5
2021-04-26	5
2009-04-13	5
2020-03-16	5
2008-08-11	5
2007-07-30	5
2009-08-24	5
2009-08-10	5
2005-01-31	5
2008-11-10	5
2017-09-11	5
2011-05-02	4
2007-02-12	4
2021-02-01	4
2016-02-29	4

2007-01-15	4
2008-03-03	4
2011-04-11	4
2020-09-28	4
2005-10-10	4
2009-03-16	4
2007-08-13	4
2009-03-23	4
2011-03-28	4
2007-07-09	4
2007-11-05	4
2007-09-24	4
2005-01-10	4
2007-03-26	4
2020-09-21	4
2005-10-24	4
2010-11-08	3
2013-04-29	3
2005-09-12	3
2006-01-30	3
2023-04-17	3
2019-04-08	3
2005-03-28	3
2006-08-21	3
2013-09-02	3
2011-05-16	3
2009-02-09	3
2007-11-26	3
2007-07-02	3
2007-02-19	3
2019-10-21	3
2007-04-30	3
2007-05-07	3
2007-05-21	3
2007-05-28	3
2006-12-18	3
2008-05-12	3
2007-06-04	3
2006-09-18	3
2008-02-11	3
2005-11-21	2
2007-02-26	2
2007-03-05	2
2005-03-14	2
2006-01-23	2
2006-01-09	2
2023-02-06	2
2005-01-17	2
2007-03-12	2
2007-03-19	2
2007-04-02	2
2005-11-28	2
2005-08-01	2
2005-08-08	2
2005-07-04	2
2007-05-14	2
2008-12-22	2
2005-02-14	2
2005-07-18	2
2006-02-20	2
2022-08-08	2
2007-06-25	2
2005-08-29	2

2005-08-15	2
2022-01-31	2
2019-05-13	2
2006-02-27	2
2015-05-18	2
2016-09-19	2
2021-05-03	2
2006-10-09	2
2006-10-23	2
2008-04-14	2
2006-07-03	2
2006-06-12	2
2016-05-16	2
2006-12-25	2
2016-04-25	2
2008-05-19	2
2016-03-07	2
2008-06-16	2
2010-05-10	2
2007-01-22	2
2007-02-05	2
2006-05-01	2
2006-03-13	2
2006-04-10	2
2006-04-17	2
2005-03-07	2
2006-05-15	2
2006-05-22	2
2007-11-12	1
2004-10-18	1
2023-01-16	1
2009-04-06	1
2009-03-30	1
2004-12-27	1
2007-07-16	1
2019-11-04	1
2023-09-04	1
2008-09-29	1
2008-05-26	1
2004-12-13	1
2008-04-28	1
2005-02-07	1
2005-01-24	1
2004-12-06	1
2005-02-21	1
2023-03-06	1
2004-11-22	1
2011-08-01	1
2004-11-15	1
2004-11-01	1
2007-10-29	1
2006-09-11	1
2020-04-27	1
2007-01-08	1
2017-03-06	1
2010-06-14	1
2006-05-08	1
2015-04-20	1
2006-06-05	1
2007-01-29	1
2007-01-01	1
2006-02-06	1
2006-06-26	1

```
2006-07-17       1
2006-07-31       1
2006-08-14       1
2006-08-28       1
2006-02-13       1
2014-11-24       1
2020-05-04       1
2005-09-19       1
2005-04-18       1
2005-04-25       1
2005-06-13       1
2005-07-11       1
2009-05-04       1
2007-06-11       1
2014-04-28       1
2004-10-11       1
2006-09-04       1
2005-10-17       1
2005-11-14       1
2005-12-19       1
2005-12-26       1
2006-01-16       1
2005-10-03       1
Name: count, dtype: int64
```

In [24]:
```
sorted_dateweek_counts = market_values_A['dateweek'].value_counts().sort_index()
```

In [25]:
```
sorted_dateweek_counts
```

Out[25]:

dateweek	
2004-10-04	20
2004-10-11	1
2004-10-18	1
2004-11-01	1
2004-11-15	1
2004-11-22	1
2004-12-06	1
2004-12-13	1
2004-12-27	1
2005-01-10	4
2005-01-17	2
2005-01-24	1
2005-01-31	5
2005-02-07	1
2005-02-14	2
2005-02-21	1
2005-03-07	2
2005-03-14	2
2005-03-28	3
2005-04-18	1
2005-04-25	1
2005-06-13	1
2005-07-04	2
2005-07-11	1
2005-07-18	2
2005-07-25	2
2005-08-01	2
2005-08-08	2
2005-08-15	2
2005-08-29	2
2005-09-12	3
2005-09-19	1
2005-10-03	1
2005-10-10	4
2005-10-17	1
2005-10-24	4
2005-11-14	1
2005-11-21	2
2005-11-28	2
2005-12-05	5
2005-12-19	1
2005-12-26	1
2006-01-09	2
2006-01-16	1
2006-01-23	2
2006-01-30	3
2006-02-06	1
2006-02-13	1
2006-02-20	2
2006-02-27	2
2006-03-13	2
2006-04-10	2
2006-04-17	2
2006-05-01	2
2006-05-08	1
2006-05-15	2
2006-05-22	2
2006-05-29	4
2006-06-05	1
2006-06-12	2
2006-06-26	1
2006-07-17	1

2006-07-31	1
2006-08-14	1
2006-08-21	3
2006-08-28	1
2006-09-04	1
2006-09-11	1
2006-09-18	3
2006-09-25	5
2006-10-02	6
2006-10-09	2
2006-10-16	7
2006-10-23	2
2006-10-30	1
2006-12-18	3
2006-12-25	2
2007-01-01	1
2007-01-08	1
2007-01-15	4
2007-01-22	2
2007-01-29	1
2007-02-05	2
2007-02-12	4
2007-02-19	3
2007-02-26	2
2007-03-05	2
2007-03-12	2
2007-03-19	2
2007-03-26	4
2007-04-02	2
2007-04-23	6
2007-04-30	3
2007-05-07	3
2007-05-14	2
2007-05-21	3
2007-05-28	3
2007-06-04	3
2007-06-11	1
2007-06-18	14
2007-06-25	2
2007-07-02	3
2007-07-09	4
2007-07-16	1
2007-07-23	22
2007-07-30	5
2007-08-06	10
2007-08-13	4
2007-08-20	10
2007-08-27	19
2007-09-03	10
2007-09-10	6
2007-09-17	10
2007-09-24	4
2007-10-01	21
2007-10-08	11
2007-10-15	7
2007-10-22	9
2007-10-29	1
2007-11-05	4
2007-11-12	1
2007-11-19	10
2007-11-26	3
2007-12-10	13
2007-12-24	14

```
2007-12-31       6
2008-01-07      16
2008-01-14       5
2008-01-21      10
2008-01-28       9
2008-02-04      37
2008-02-11       3
2008-02-18      10
2008-02-25       7
2008-03-03       4
2008-03-10       6
2008-03-17      12
2008-03-31       8
2008-04-07      12
2008-04-14       2
2008-04-21       5
2008-04-28       1
2008-05-05      11
2008-05-12       3
2008-05-19       2
2008-05-26       1
2008-06-02       8
2008-06-09       7
2008-06-16       2
2008-06-23       5
2008-06-30      37
2008-07-07       7
2008-07-14      27
2008-07-21       8
2008-07-28       8
2008-08-04       9
2008-08-11       5
2008-08-18      14
2008-08-25       8
2008-09-01      14
2008-09-08      13
2008-09-15      11
2008-09-22      12
2008-09-29       1
2008-10-06      10
2008-10-13      27
2008-10-20      18
2008-10-27       7
2008-11-03       7
2008-11-10       5
2008-11-17       7
2008-11-24       9
2008-12-01      12
2008-12-08      22
2008-12-15      20
2008-12-22       2
2008-12-29      26
2009-01-05      28
2009-01-12      10
2009-01-19      31
2009-01-26      23
2009-02-02      25
2009-02-09       3
2009-02-16      11
2009-02-23      24
2009-03-02       8
2009-03-09       5
2009-03-23       4
```

2009-03-30	1
2009-04-06	1
2009-04-13	5
2009-04-20	6
2009-04-27	8
2009-05-04	1
2009-05-11	5
2009-05-18	25
2009-05-25	10
2009-06-01	16
2009-06-08	25
2009-06-15	18
2009-06-22	26
2009-06-29	27
2009-07-06	16
2009-07-13	10
2009-07-20	25
2009-08-03	46
2009-08-10	5
2009-08-17	20
2009-08-24	5
2009-08-31	34
2009-09-07	9
2009-09-21	28
2009-09-28	7
2009-10-05	22
2009-10-12	7
2009-10-19	24
2009-10-26	9
2009-11-02	7
2009-11-09	16
2009-11-16	10
2009-11-23	26
2009-11-30	24
2009-12-07	29
2009-12-14	8
2009-12-21	27
2009-12-28	6
2010-01-04	22
2010-01-11	44
2010-01-18	34
2010-01-25	42
2010-02-01	19
2010-02-08	11
2010-02-15	24
2010-02-22	13
2010-03-01	17
2010-03-08	17
2010-03-15	6
2010-03-22	9
2010-03-29	14
2010-04-05	33
2010-04-12	22
2010-04-19	17
2010-04-26	8
2010-05-03	14
2010-05-10	2
2010-05-17	28
2010-05-24	15
2010-05-31	17
2010-06-07	20
2010-06-14	1
2010-06-28	27

```
2010-07-05      10
2010-07-12      52
2010-07-19      15
2010-07-26      69
2010-08-02      25
2010-08-09      40
2010-08-16      41
2010-08-23      66
2010-08-30      13
2010-09-06      16
2010-09-13       6
2010-09-20      16
2010-09-27      20
2010-10-04      13
2010-10-11      11
2010-10-18      60
2010-10-25       5
2010-11-01       6
2010-11-08       3
2010-11-15      24
2010-11-22      21
2010-11-29       9
2010-12-06      29
2010-12-13       7
2010-12-20       8
2010-12-27      97
2011-01-03      10
2011-01-10      84
2011-01-17      72
2011-01-24      11
2011-01-31     117
2011-02-07      35
2011-02-14       8
2011-02-21      20
2011-02-28      21
2011-03-07       6
2011-03-14      13
2011-03-21       7
2011-03-28       4
2011-04-04      14
2011-04-11       4
2011-04-18       7
2011-04-25       8
2011-05-02       4
2011-05-09      11
2011-05-16       3
2011-05-23      59
2011-05-30      81
2011-06-06      10
2011-06-13      63
2011-06-20      50
2011-06-27     165
2011-07-04      41
2011-07-11      36
2011-07-18      45
2011-07-25      48
2011-08-01       1
2011-08-08      82
2011-08-15      20
2011-08-22      24
2011-08-29      56
2011-09-05      16
2011-09-19      29
```

2011-09-26	21
2011-10-10	18
2011-10-17	38
2011-10-24	7
2011-10-31	8
2011-11-07	12
2011-11-14	15
2011-11-21	16
2011-11-28	6
2011-12-05	30
2011-12-12	22
2011-12-19	114
2011-12-26	83
2012-01-02	60
2012-01-09	99
2012-01-16	38
2012-01-23	73
2012-01-30	127
2012-02-06	76
2012-02-13	11
2012-02-20	22
2012-02-27	30
2012-03-05	31
2012-03-12	8
2012-03-19	15
2012-03-26	35
2012-04-02	13
2012-04-09	23
2012-04-16	6
2012-04-23	7
2012-04-30	12
2012-05-21	76
2012-05-28	54
2012-06-04	73
2012-06-11	50
2012-06-18	136
2012-06-25	95
2012-07-02	115
2012-07-09	24
2012-07-16	63
2012-07-23	41
2012-07-30	34
2012-08-06	90
2012-08-13	27
2012-08-20	29
2012-08-27	20
2012-09-03	23
2012-09-10	10
2012-09-17	60
2012-09-24	28
2012-10-01	9
2012-10-08	8
2012-10-15	9
2012-10-22	29
2012-10-29	21
2012-11-05	37
2012-11-12	15
2012-11-19	38
2012-11-26	18
2012-12-03	35
2012-12-10	12
2012-12-17	27
2012-12-31	117

2013-01-07	302
2013-01-14	108
2013-01-21	55
2013-01-28	71
2013-02-04	52
2013-02-11	30
2013-02-18	6
2013-02-25	56
2013-03-04	21
2013-03-11	16
2013-03-18	8
2013-03-25	26
2013-04-01	15
2013-04-08	28
2013-04-15	17
2013-04-22	12
2013-04-29	3
2013-05-06	39
2013-05-13	10
2013-05-20	11
2013-05-27	143
2013-06-03	71
2013-06-10	167
2013-06-17	199
2013-06-24	219
2013-07-01	131
2013-07-08	51
2013-07-22	112
2013-07-29	10
2013-08-05	46
2013-08-12	18
2013-08-19	12
2013-08-26	32
2013-09-02	3
2013-09-09	58
2013-09-16	57
2013-09-23	36
2013-10-07	121
2013-10-14	10
2013-10-21	14
2013-10-28	11
2013-11-04	44
2013-11-11	11
2013-11-25	25
2013-12-02	25
2013-12-23	60
2013-12-30	69
2014-01-06	120
2014-01-13	209
2014-01-20	146
2014-01-27	85
2014-02-03	60
2014-02-10	306
2014-02-17	47
2014-02-24	18
2014-03-03	36
2014-03-10	28
2014-03-17	7
2014-03-24	97
2014-03-31	31
2014-04-14	10
2014-04-21	5
2014-05-05	27

2014-05-12	22
2014-06-09	9
2014-06-16	25
2014-06-23	120
2014-06-30	192
2014-07-07	135
2014-07-14	165
2014-07-21	100
2014-07-28	83
2014-08-04	156
2014-08-11	240
2014-08-18	46
2014-08-25	57
2014-09-01	24
2014-09-08	23
2014-09-15	18
2014-09-22	20
2014-09-29	99
2014-10-06	56
2014-10-13	12
2014-10-20	48
2014-10-27	18
2014-11-03	64
2014-11-10	76
2014-11-17	70
2014-11-24	1
2014-12-01	23
2014-12-08	7
2014-12-15	50
2014-12-22	80
2014-12-29	16
2015-01-05	288
2015-01-12	177
2015-01-19	125
2015-01-26	82
2015-02-02	202
2015-02-09	219
2015-02-16	160
2015-02-23	129
2015-03-02	58
2015-03-09	15
2015-03-16	29
2015-03-23	35
2015-03-30	10
2015-04-06	52
2015-04-13	14
2015-04-20	1
2015-05-04	52
2015-05-11	9
2015-05-18	2
2015-05-25	38
2015-06-01	254
2015-06-08	85
2015-06-15	163
2015-06-22	113
2015-06-29	923
2015-07-06	13
2015-07-13	52
2015-07-20	42
2015-07-27	48
2015-08-03	10
2015-08-10	29
2015-08-24	67

2015-08-31	148
2015-09-07	28
2015-09-14	46
2015-09-21	164
2015-09-28	34
2015-10-05	24
2015-10-12	119
2015-10-19	47
2015-10-26	25
2015-11-02	45
2015-11-09	54
2015-11-16	76
2015-11-23	85
2015-11-30	38
2015-12-07	29
2015-12-14	153
2015-12-21	34
2015-12-28	83
2016-01-04	258
2016-01-11	53
2016-01-18	141
2016-01-25	18
2016-02-01	468
2016-02-08	246
2016-02-15	141
2016-02-22	304
2016-02-29	4
2016-03-07	2
2016-03-14	93
2016-03-21	10
2016-03-28	52
2016-04-04	66
2016-04-11	34
2016-04-18	83
2016-04-25	2
2016-05-02	7
2016-05-09	6
2016-05-16	2
2016-05-23	117
2016-05-30	264
2016-06-06	34
2016-06-13	100
2016-06-20	76
2016-06-27	126
2016-07-04	9
2016-07-11	900
2016-07-18	343
2016-07-25	98
2016-08-01	332
2016-08-08	130
2016-08-15	17
2016-08-22	18
2016-08-29	114
2016-09-05	12
2016-09-12	30
2016-09-19	2
2016-09-26	93
2016-10-03	33
2016-10-10	104
2016-10-17	59
2016-10-24	119
2016-10-31	84
2016-11-14	22

```
2016-11-21       9
2016-11-28     118
2016-12-05      26
2016-12-12     189
2016-12-19     221
2016-12-26     120
2017-01-02     263
2017-01-09     124
2017-01-16     421
2017-01-23     297
2017-01-30     141
2017-02-06     250
2017-02-13      69
2017-02-20     292
2017-02-27     111
2017-03-06       1
2017-03-13      33
2017-03-20     121
2017-03-27     106
2017-04-10      15
2017-04-17      38
2017-05-01      28
2017-05-08      70
2017-05-15      11
2017-05-22      28
2017-05-29     361
2017-06-05     739
2017-06-12     112
2017-06-19     394
2017-06-26     791
2017-07-03      36
2017-07-10      66
2017-07-17      17
2017-07-24      17
2017-07-31      35
2017-08-07       6
2017-08-14     120
2017-08-21      39
2017-08-28       6
2017-09-04     100
2017-09-11       5
2017-09-18      81
2017-09-25      20
2017-10-02      61
2017-10-09     141
2017-10-16     109
2017-10-23     160
2017-10-30      20
2017-11-06      23
2017-11-13      45
2017-11-20      39
2017-11-27      16
2017-12-04     101
2017-12-11      22
2017-12-18     386
2017-12-25     450
2018-01-01     657
2018-01-08     293
2018-01-15     209
2018-01-22     262
2018-01-29      26
2018-02-05     172
2018-02-19      83
```

```
2018-02-26     246
2018-03-05       9
2018-03-12      44
2018-03-19     187
2018-03-26      69
2018-04-02     101
2018-04-09      27
2018-04-16      15
2018-04-23      63
2018-04-30      21
2018-05-07      35
2018-05-14      85
2018-05-21     169
2018-05-28     925
2018-06-04     868
2018-06-11     310
2018-06-18     170
2018-06-25     216
2018-07-02     102
2018-07-09     100
2018-07-16      66
2018-07-23       9
2018-07-30      25
2018-08-06     233
2018-08-13      52
2018-08-20      74
2018-08-27      24
2018-09-03      86
2018-09-10      95
2018-09-17      44
2018-09-24     133
2018-10-01      63
2018-10-08     101
2018-10-15     148
2018-10-22     160
2018-10-29     146
2018-11-05      18
2018-11-12       9
2018-11-19      14
2018-11-26      64
2018-12-03     128
2018-12-10     172
2018-12-17    1601
2018-12-24     403
2018-12-31     262
2019-01-07     340
2019-01-14     151
2019-01-21     158
2019-01-28       9
2019-02-04      46
2019-02-11      76
2019-02-18     220
2019-02-25     180
2019-03-04     223
2019-03-11      93
2019-03-18      61
2019-03-25      24
2019-04-01      16
2019-04-08       3
2019-04-15      39
2019-04-22      27
2019-04-29      19
2019-05-13       2
```

2019-05-20	147
2019-05-27	67
2019-06-03	1198
2019-06-10	1294
2019-06-17	272
2019-06-24	607
2019-07-01	90
2019-07-08	10
2019-07-15	11
2019-07-22	52
2019-07-29	31
2019-08-05	59
2019-08-12	44
2019-08-19	66
2019-08-26	8
2019-09-02	26
2019-09-09	417
2019-09-16	109
2019-09-23	167
2019-09-30	153
2019-10-07	133
2019-10-14	108
2019-10-21	3
2019-10-28	38
2019-11-04	1
2019-11-11	34
2019-11-18	26
2019-11-25	68
2019-12-02	77
2019-12-09	905
2019-12-16	1432
2019-12-23	449
2019-12-30	434
2020-01-06	316
2020-01-13	60
2020-01-20	16
2020-01-27	16
2020-02-03	180
2020-02-10	173
2020-02-17	33
2020-02-24	139
2020-03-02	122
2020-03-09	238
2020-03-16	5
2020-03-23	18
2020-03-30	156
2020-04-06	3523
2020-04-27	1
2020-05-04	1
2020-05-11	8
2020-06-15	20
2020-06-22	88
2020-06-29	233
2020-07-06	136
2020-07-13	60
2020-07-20	264
2020-07-27	537
2020-08-03	430
2020-08-10	81
2020-08-17	332
2020-08-24	367
2020-08-31	173
2020-09-14	130

```
2020-09-21       4
2020-09-28       4
2020-10-05     521
2020-10-12    1263
2020-10-19     250
2020-10-26      69
2020-11-02      75
2020-11-09     145
2020-11-16      41
2020-11-23     163
2020-11-30      71
2020-12-07      58
2020-12-14     174
2020-12-21     532
2020-12-28     949
2021-01-04     816
2021-01-11     243
2021-01-18     199
2021-01-25     236
2021-02-01       4
2021-02-08     651
2021-02-15      25
2021-02-22     111
2021-03-01      13
2021-03-08     304
2021-03-15     752
2021-03-22     170
2021-03-29      76
2021-04-05     132
2021-04-12      80
2021-04-19      31
2021-04-26       5
2021-05-03       2
2021-05-10      18
2021-05-17     632
2021-05-24     550
2021-05-31    1168
2021-06-07    1235
2021-06-14     231
2021-06-21     428
2021-06-28     503
2021-07-05      50
2021-07-12      54
2021-07-19      25
2021-07-26      42
2021-08-02      40
2021-08-09      35
2021-08-16       6
2021-08-23      10
2021-08-30      44
2021-09-06      38
2021-09-13      90
2021-09-20      40
2021-09-27      20
2021-10-04     582
2021-10-11     725
2021-10-18      65
2021-10-25      50
2021-11-01     220
2021-11-08      41
2021-11-15       5
2021-11-22     144
2021-12-06     269
```

2021-12-13	559
2021-12-20	1068
2021-12-27	1967
2022-01-03	854
2022-01-10	190
2022-01-17	74
2022-01-31	2
2022-02-07	14
2022-02-14	14
2022-02-21	44
2022-03-07	60
2022-03-14	196
2022-03-21	390
2022-03-28	504
2022-04-04	57
2022-04-11	284
2022-04-18	32
2022-05-02	18
2022-05-09	140
2022-05-16	281
2022-05-23	132
2022-05-30	1686
2022-06-06	1391
2022-06-13	575
2022-06-20	514
2022-06-27	576
2022-07-04	15
2022-07-11	5
2022-07-18	14
2022-07-25	31
2022-08-01	38
2022-08-08	2
2022-08-15	82
2022-08-29	26
2022-09-05	196
2022-09-12	579
2022-09-19	345
2022-09-26	157
2022-10-03	71
2022-10-10	37
2022-10-17	6
2022-10-24	572
2022-10-31	1381
2022-11-07	1816
2022-11-14	166
2022-11-21	312
2022-11-28	114
2022-12-05	451
2022-12-12	279
2022-12-19	471
2022-12-26	266
2023-01-02	45
2023-01-09	15
2023-01-16	1
2023-01-23	6
2023-01-30	11
2023-02-06	2
2023-02-13	7
2023-02-20	9
2023-02-27	182
2023-03-06	1
2023-03-13	552
2023-03-27	361

```
2023-04-03     257
2023-04-10      89
2023-04-17       3
2023-05-08       5
2023-05-15       8
2023-05-22     266
2023-05-29     234
2023-06-05     765
2023-06-12    1815
2023-06-19    1769
2023-06-26    1152
2023-07-03     137
2023-07-10      46
2023-07-17       9
2023-07-24      49
2023-07-31       8
2023-08-07      10
2023-08-14       5
2023-08-21      13
2023-08-28       8
2023-09-04       1
2023-09-11      72
2023-09-18     130
Name: count, dtype: int64
```

In [26]:
```
#market_values_A = player_valuations_df[['player_id', 'market_value_in_eur']]
```

In [27]:
```
print(market_values_A.shape)
print(market_values_A.head())
print(market_values_A.nunique())
```

```
(101275, 9)
     player_id  last_season             datetime        date    dateweek  \
186       3333         2023  2004-10-04 00:00:00  2004-10-04  2004-10-04
343       4391         2023  2004-10-04 00:00:00  2004-10-04  2004-10-04
743       8246         2023  2004-10-04 00:00:00  2004-10-04  2004-10-04
782       9500         2023  2004-10-04 00:00:00  2004-10-04  2004-10-04
884      12029         2023  2004-10-04 00:00:00  2004-10-04  2004-10-04

     market_value_in_eur  n  current_club_id  \
186              7500000  1             1237
343              1300000  1              383
743                50000  1                3
782               750000  1              903
884              1500000  1             1421

    player_club_domestic_competition_id
186                                 GB1
343                                 NL1
743                                  L1
782                                 SC1
884                                 FR1
player_id                          6407
last_season                           1
datetime                           3330
date                               3330
dateweek                            918
market_value_in_eur                 240
n                                     1
current_club_id                     238
player_club_domestic_competition_id  14
dtype: int64
```

In the best case ther are only 6407 values for unique player_id. Less then what the players_df currently contains (19383).

In [28]:
```
market_values_B = player_valuations_df.copy()
```

In [29]:
```
market_values_B = pd.to_datetime(market_values_B['dateweek'])
```

In [30]:
```
market_values_B = player_valuations_df[player_valuations_df['dateweek'] > '2022-12-
```

In [31]:
```
market_values_B.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 15569 entries, 425094 to 440662
Data columns (total 9 columns):
 #   Column                                Non-Null Count  Dtype 
---  ------                                --------------  ----- 
 0   player_id                             15569 non-null  int64 
 1   last_season                           15569 non-null  int64 
 2   datetime                              15569 non-null  object
 3   date                                  15569 non-null  object
 4   dateweek                              15569 non-null  object
 5   market_value_in_eur                   15569 non-null  int64 
 6   n                                     15569 non-null  int64 
 7   current_club_id                       15569 non-null  int64 
 8   player_club_domestic_competition_id   15569 non-null  object
dtypes: int64(5), object(4)
memory usage: 1.2+ MB
```

In [32]:
```
market_values_B.nunique()
```

Out[32]:
```
player_id                             12602
last_season                              12
datetime                                148
date                                    148
dateweek                                 37
market_value_in_eur                     125
n                                         1
current_club_id                         413
player_club_domestic_competition_id      14
dtype: int64
```

In the best case ther are only 12602 values for unique player_id. Less then what the players_df currently contains (19383).

In [33]:
```
market_values_C = players_df[['player_id', 'market_value_in_eur']]
```

In [34]:
```
print(market_values_C.shape)
print(market_values_C.head())
print(market_values_C.nunique())
```

```
(30302, 2)
   player_id  market_value_in_eur
0        598                  NaN
1        670                  NaN
2       1323                  NaN
3       3195                  NaN
4       3259                  NaN
player_id              30302
market_value_in_eur      125
```

We will readdress the missing values in 'market_value_in_eur' and 'highest_market_value_in_eur' below.

Feature Engineering

Feature generation

In [35]:
```
players_df.columns
```

Out[35]:
```
Index(['player_id', 'first_name', 'last_name', 'name', 'last_season',
       'current_club_id', 'player_code', 'country_of_birth', 'city_of_birth',
       'country_of_citizenship', 'date_of_birth', 'sub_position', 'position',
       'foot', 'height_in_cm', 'market_value_in_eur',
       'highest_market_value_in_eur', 'contract_expiration_date', 'image_url',
       'url', 'current_club_domestic_competition_id', 'current_club_name'],
      dtype='object')
```

In [36]:
```
pd.set_option('display.max_columns', None) # to display all columns
```

In [37]:
```
players_df['date_of_birth'].describe()
```

Out[37]:
```
count          30255
unique          8892
top       1996-01-19
freq              20
Name: date_of_birth, dtype: object
```

In [38]:
```
players_df[players_df['date_of_birth'].isnull() == True]
```

Out[38]:

	player_id	first_name	last_name	name	last_season	current_club_id	player_c
2720	335646	Nick	Volk	Nick Volk	2015	105	nick-
2756	345068	Abdullah	Karakoc	Abdullah Karakoc	2014	3216	abdu kar
3029	539248	Kalle	Christensen	Kalle Christensen	2017	2414	k christe
3144	640558	Damian	van der Haar	Damian van der Haar	2021	1269	damian- der-
3387	16565	Daniel	Halfar	Daniel Halfar	2014	3	daniel-h
4328	221883	Abdul	Basit	Abdul Basit	2016	1301	abdul-l
5105	714313	José	Salinas	José Salinas	2020	1531	jose-sa
5493	58379	Maximilian	Beister	Maximilian Beister	2015	39	maximi be
6774	531161	Rasmus	Horup Hansen	Rasmus Horup Hansen	2017	2414	rasr horup-ha
6834	573816	Ibrahim	Suaib	Ibrahim Suaib	2017	6992	ibrahim-s
7427	83188	Marc	Lais	Marc Lais	2012	60	marc
10522	523200	Kristoffer	Larsen	Kristoffer Larsen	2017	2414	kristo la
11068	50765	Mahmut	Temür	Mahmut Temür	2014	3216	mahı te
11879	42463	Timo	Gebhart	Timo Gebhart	2013	4	timo-geb
11889	51190	Marc	Vucinovic	Marc Vucinovic	2014	127	m vucir
12295	457583	Dimitri	Swisch	Dimitri Swisch	2013	9007	dimitri-sw
12447	576729	Sergiy	Rusyan	Sergiy Rusyan	2017	18303	sergiy-ru
12585	701335	Cihat	Topatan	Cihat Topatan	2021	11688	cihat-top
12711	1067951	Hjalte	Toftegaard	Hjalte Toftegaard	2022	2414	hj tofteg
12871	36189	Christian	Dorda	Christian Dorda	2014	968	chris d
13768	837	Thorben	Marx	Thorben Marx	2014	18	thorben-r
13770	2950	Robert	Almer	Robert Almer	2014	42	robert-a
			t	Fabian Ernst	2012	10484	fabian-e

	player_id	first_name	last_name	name	last_season	current_club_id	player_c
15882	94598	Stefan	Wannenwetsch	Stefan Wannenwetsch	2015	4795	ste wannenwe
16200	632716	Kalle	Bak	Kalle Bak	2019	5818	kalle
16419	3942	Martin	Lanig	Martin Lanig	2014	24	martin-l
16432	6322	Patrick	Helmes	Patrick Helmes	2014	3	pat hel
17349	335635	Jan	Finger	Jan Finger	2015	105	jan-fi
17458	381132	Daniel	Thur	Daniel Thur	2016	105	daniel-
18362	20444	Tobias	Weis	Tobias Weis	2013	24	tobias-
19255	328791	Miguel	Blanco-Lopez	Miguel Blanco-Lopez	2016	24	mig blanco-l
19350	381134	Johannes	Wolff	Johannes Wolff	2016	105	johan
21385	546133	Johannes	Frahm	Johannes Frahm	2019	5724	johan fr
22259	49681	Aykut	Öztürk	Aykut Öztürk	2013	2381	aykut-oz
22986	335633	Noel	Wembacher	Noel Wembacher	2015	105	r wemba
23974	879619	Sem	Bonte	Sem Bonte	2022	133	sem-b
24644	246051	Martin	Haesum	Martin Haesum	2012	1053	ma hae
25460	1944	Deniz	Dogan	Deniz Dogan	2013	23	deniz-dc
25874	521323	Filippos	Selkos	Filippos Selkos	2019	169	filippos-se
25926	1587	Mario	Eggimann	Mario Eggimann	2012	42	ma eggim
26032	32611	Sezer	Badur	Sezer Badur	2015	589	sezer-b
26043	58124	Julian	Schieber	Julian Schieber	2020	167	julian-schi
26252	116272	Benedikt	Röcker	Benedikt Röcker	2018	206	bene rc
26458	202753	Oguz	Özcelik	Oguz Özcelik	2012	2414	oguz-oz
27385	1086787	Demba	N'Diaye	Demba N'Diaye	2022	1420	demba-nc
				Djibi Seck	2023	601	djibi-

	player_id	first_name	last_name	name	last_season	current_club_id	player_c
29562	539732	Yuriy	Berdey	Yuriy Berdey	2017	26459	yuriy-be

Identfying the players with missing DOB but whom contain a market value for using in ML modelling

In [39]:
```
players_df[(players_df['date_of_birth'].isnull() == True) & (players_df['market_val
```

Out[39]:

	player_id	first_name	last_name	name	last_season	current_club_id	player_code	coun
5493	58379	Maximilian	Beister	Maximilian Beister	2015	39	maximilian-beister	

In [40]:
```
# Replacing the mssing DOB with correct value from transfermarkt
players_df.loc[players_df['name'] == 'Maximilian Beister', 'date_of_birth'] = '1990
```

In [41]:
```
# Confirming change
players_df.loc[players_df['name'] == 'Maximilian Beister']
```

Out[41]:

	player_id	first_name	last_name	name	last_season	current_club_id	player_code	coun
5493	58379	Maximilian	Beister	Maximilian Beister	2015	39	maximilian-beister	

In [42]:
```
# Converting from object to datetime type
players_df['date_of_birth'] = pd.to_datetime(players_df['date_of_birth'])
```

In [43]:
```
# Replacing the remaing missng DOB values with the mean
players_df['date_of_birth'] = players_df['date_of_birth'].fillna(players_df['date_c
```

In [44]:
```
# Confirming imputation
players_df[players_df['date_of_birth'].isnull() == True]
```

Out[44]:

	player_id	first_name	last_name	name	last_season	current_club_id	player_code	country_of_bir

In [45]:
```
players_df['date_of_birth'].isnull().sum()
```

Out[45]:
```
0
```

Creating a feature for Age

In [46]:
```
# Calculating the age of each player
players_df['date_of_birth'] = pd.to_datetime(players_df['date_of_birth'])
# Instantiating an object now for the current datetime
now = datetime.now()
# Creating a feature for age based on the difference between the current datetime c
players_df['age'] = (now - players_df['date_of_birth']).apply(lambda x: x.days) / 3
# rounding up their age and casting to integer type
players_df['age'] = players_df['age'].round().astype(int)
```

Creating a feature for remaining contract length

In [47]:
```
# Calculating the time remaining on player contracts
players_df['contract_expiration_date'] = pd.to_datetime(players_df['contract_expira

players_df['remaining_contract_days'] = (players_df['contract_expiration_date']- nc
```

In [48]:
```
# Confirming changes
players_df.head()
```

Out[48]:

	player_id	first_name	last_name	name	last_season	current_club_id	player_code	country_
0	598	Timo	Hildebrand	Timo Hildebrand	2014	24	timo-hildebrand	
1	670	Martin	Petrov	Martin Petrov	2012	714	martin-petrov	
2	1323	Martin	Amedick	Martin Amedick	2012	24	martin-amedick	
3	3195	Jermaine	Pennant	Jermaine Pennant	2013	512	jermaine-pennant	
4	3259	Damien	Duff	Damien Duff	2013	931	damien-duff	

In [49]:
```
players_df['age'].value_counts()
```

Out[49]:
```
age
30    1847
27    1793
29    1768
26    1766
28    1731
25    1646
31    1607
24    1589
23    1487
32    1480
22    1391
33    1274
21    1225
34    1165
35    1072
36     976
20     936
37     899
38     763
39     685
40     554
19     539
41     481
42     387
43     351
44     234
18     192
45     166
46     107
47      56
48      43
17      36
49      24
50      16
52       5
16       5
51       2
53       2
54       1
55       1
Name: count, dtype: int64
```

In [50]:
```python
# Simple function to group/categorise ages
def age_groups(age):
    if age <= 21:
        return 'Ages Under 22'
    elif 22 <= age <= 26:
        return 'Ages 22 - 26'
    elif 27 <= age <= 32:
        return 'Ages 27 - 32'
    elif 33 <= age <= 38:
        return 'Ages 33 - 38'
    elif 39 <= age <= 44:
        return 'Ages 39 - 44'
    elif 45 <= age <= 50:
        return 'Ages 45 - 50'
    else:
        return 'Ages Over 50'

# Applying the function to the players df
players_df['age_group'] = players_df['age'].apply(age_groups)
```

In [51]:
```
players_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30302 entries, 0 to 30301
Data columns (total 25 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   player_id                             30302 non-null  int64
 1   first_name                            28337 non-null  object
 2   last_name                             30302 non-null  object
 3   name                                  30302 non-null  object
 4   last_season                           30302 non-null  int64
 5   current_club_id                       30302 non-null  int64
 6   player_code                           30302 non-null  object
 7   country_of_birth                      27613 non-null  object
 8   city_of_birth                         28099 non-null  object
 9   country_of_citizenship                29759 non-null  object
 10  date_of_birth                         30302 non-null  datetime64[ns]
 11  sub_position                          30130 non-null  object
 12  position                              30302 non-null  object
 13  foot                                  27913 non-null  object
 14  height_in_cm                          28204 non-null  float64
 15  market_value_in_eur                   19383 non-null  float64
 16  highest_market_value_in_eur           28981 non-null  float64
 17  contract_expiration_date              18835 non-null  datetime64[ns]
 18  image_url                             30302 non-null  object
 19  url                                   30302 non-null  object
 20  current_club_domestic_competition_id  30302 non-null  object
 21  current_club_name                     30302 non-null  object
 22  age                                   30302 non-null  int32
 23  remaining_contract_days               18835 non-null  float64
 24  age_group                             30302 non-null  object
dtypes: datetime64[ns](2), float64(4), int32(1), int64(3), object(15)
memory usage: 5.7+ MB
```

Processing data for Visualising later

Adding the year and player positions to 'player_valuations_df'

In [52]:
```
# Converting datetime from object to datetime type
player_valuations_df['datetime']=pd.to_datetime(player_valuations_df['datetime'])
#Extracting the year from datetime and storing it to a new fetaure column 'year'
player_valuations_df['year']=player_valuations_df['datetime'].dt.year
```

In [53]:
```
more_player_info = players_df[['player_id', 'foot', 'sub_position', 'position', 'ag
```

In [54]:
```
more_player_info.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30302 entries, 0 to 30301
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   player_id     30302 non-null  int64
 1   foot          27913 non-null  object
 2   sub_position  30130 non-null  object
 3   position      30302 non-null  object
 4   age           30302 non-null  int32
 5   age_group     30302 non-null  object
dtypes: int32(1), int64(1), object(4)
```

In [55]:
```
player_valuations_df = player_valuations_df.merge(more_player_info, left_on='player
```

In [56]:
```
player_valuations_df.head()
```

Out[56]:

	player_id	last_season	datetime	date	dateweek	market_value_in_eur	n	current_club_id	play
0	3132	2013	2003-12-09	2003-12-09	2003-12-08	400000	1	126	
1	3132	2013	2004-10-04	2004-10-04	2004-10-04	2000000	1	126	
2	3132	2013	2007-10-04	2007-10-04	2007-10-01	2200000	1	126	
3	3132	2013	2008-05-04	2008-05-04	2008-04-28	2800000	1	126	
4	3132	2013	2008-10-09	2008-10-09	2008-10-06	1500000	1	126	

In [57]:
```
player_valuations_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 440571 entries, 0 to 440570
Data columns (total 15 columns):
 #   Column                                Non-Null Count   Dtype
---  ------                                --------------   -----
 0   player_id                             440571 non-null  int64
 1   last_season                           440571 non-null  int64
 2   datetime                              440571 non-null  datetime64[ns]
 3   date                                  440571 non-null  object
 4   dateweek                              440571 non-null  object
 5   market_value_in_eur                   440571 non-null  int64
 6   n                                     440571 non-null  int64
 7   current_club_id                       440571 non-null  int64
 8   player_club_domestic_competition_id   440571 non-null  object
 9   year                                  440571 non-null  int32
 10  foot                                  426727 non-null  object
 11  sub_position                          439768 non-null  object
 12  position                              440571 non-null  object
 13  age                                   440571 non-null  int32
 14  age_group                             440571 non-null  object
dtypes: datetime64[ns](1), int32(2), int64(5), object(7)
memory usage: 47.1+ MB
```

In [58]:
```
filtered_position_df = player_valuations_df.dropna(subset=['position'])
filtered_position_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 440571 entries, 0 to 440570
Data columns (total 15 columns):
 #   Column                                Non-Null Count   Dtype
---  ------                                --------------   -----
 0   player_id                             440571 non-null  int64
 1   last_season                           440571 non-null  int64
 2   datetime                              440571 non-null  datetime64[ns]
 3   date                                  440571 non-null  object
 4   dateweek                              440571 non-null  object
 5   market_value_in_eur                   440571 non-null  int64
 6   n                                     440571 non-null  int64
 7   current_club_id                       440571 non-null  int64
 8   player_club_domestic_competition_id   440571 non-null  object
 9   year                                  440571 non-null  int32
 10  foot                                  426727 non-null  object
 11  sub_position                          439768 non-null  object
 12  position                              440571 non-null  object
 13  age                                   440571 non-null  int32
 14  age_group                             440571 non-null  object
dtypes: datetime64[ns](1), int32(2), int64(5), object(7)
memory usage: 47.1+ MB
```

In [59]:
```
filtered_position_df['position'].value_counts()
```

Out[59]:
```
position
Defender      142596
Midfield      128467
Attack        121930
Goalkeeper     46775
Missing          803
Name: count, dtype: int64
```

In [60]:
```
filtered_position_df = player_valuations_df[player_valuations_df['position'] != 'Mi
```

In [61]:
```
filtered_position_df['position'].value_counts()
```

Out[61]:
```
position
Defender      142596
Midfield      128467
Attack        121930
Goalkeeper     46775
Name: count, dtype: int64
```

Attempting to impute missing position based on sub_position

In [62]:
```
players_df['position'].value_counts()
```

Out[62]:
```
position
Defender      9654
Midfield      8767
Attack        8235
Goalkeeper    3474
Missing        172
Name: count, dtype: int64
```

In [63]:
```
na_position_df = players_df[players_df['position'].isnull() == True]
na_position_df.shape
```

Out[63]:
```
(0, 25)
```

In [64]:
```
missing_position_df = players_df[players_df['position'] == 'Missing']
```

Out[64]: (172, 25)

Checking the missing position df for high market value players

In [65]:
```
missing_position_df_sorted_MV = missing_position_df.sort_values("market_value_in_eu
```

In [66]:
```
missing_position_df_sorted_MV.head(173)
```

Out[66]:

	player_id	first_name	last_name	name	last_season	current_club_id	
26294	131480	Il-lok	Yun	Il-lok Yun	2020	969	
7375	66112	Keko	Gontán	Keko Gontán	2018	366	
10119	288005	Joaquín	Arzura	Joaquín Arzura	2020	6418	jo
3154	652672	Kai	Fotheringham	Kai Fotheringham	2022	1519	kai-f
11542	220769	Nikola	Asceric	Nikola Asceric	2017	3999	n
28593	297241	Liam	Burt	Liam Burt	2016	124	
1463	936901	Dominic	Sadi	Dominic Sadi	2022	989	
9765	654947	Dylan	Demuynck	Dylan Demuynck	2022	3508	dyl
25359	733434	Jack	Wadham	Jack Wadham	2022	989	.
26756	340063	Gianluca	Piccoli	Gianluca Piccoli	2018	416	gia
2419	193955	Stefanos	Martsakis	Stefanos Martsakis	2014	653	
29372	1031590	Sotiris	Kontouris	Sotiris Kontouris	2023	6418	sot
16246	671068	Bassim	Boukteb	Bassim Boukteb	2021	2715	bas
18262	1094044	Zacharias	Papadimitriou	Zacharias Papadimitriou	2023	3385	p
27268	746999	Fergus	Dee	Fergus Dee	2020	2553	
86	22395	Adil	Hermach	Adil Hermach	2012	415	
282	64385	José Manuel	Catalá	José Manuel Catalá	2014	2079	-
353	90600	Josselin	Thibaut	Josselin Thibaut	2014	29228	jos
540	168887	Gianni	Van Landschoot	Gianni Van Landschoot	2012	520	
586	189507	Jonathan	Black	Jonathan Black	2014	2760	jo
614	199730	Claudio	Andreoli	Claudio Andreoli	2012	398	cla
718	244258	Dmytro	Lyakh	Dmytro Lyakh	2013	2227	

	player_id	first_name	last_name	name	last_season	current_club_id	
1228	540580	Maksym	Sorochenko	Maksym Sorochenko	2016	4482	
1772	182689	Rafail	Soukias	Rafail Soukias	2014	21957	
2498	225546	Marvin	Novotny	Marvin Novotny	2012	192	ma
2601	271446	NaN	Kiu	Kiu	2013	3302	
2675	298659	Vasyl	Grinka	Vasyl Grinka	2013	2227	
2939	464142	Diego	Snepvangers	Diego Snepvangers	2017	132	
3092	598917	Georgios	Christopoulos	Georgios Christopoulos	2018	441	
3109	612809	Milan	Govaers	Milan Govaers	2022	2861	r
3433	33100	Suleiman	Omo	Suleiman Omo	2018	2672	s
3778	273542	Noureddine	Boutzamar	Noureddine Boutzamar	2015	133	
4297	210202	James	Horsfield	James Horsfield	2018	511	jar
4328	221883	Abdul	Basit	Abdul Basit	2016	1301	
4347	231165	Fotis	Doumanis	Fotis Doumanis	2013	128	fc
4430	266911	Georgios	Peppas	Georgios Peppas	2013	2672	geo
4625	348269	Oleksandr	Dekhtiarenko	Oleksandr Dekhtiarenko	2014	6993	
4975	584434	Adrian	Imeri	Adrian Imeri	2018	3060	
5673	118645	Branislav	Nikic	Branislav Nikic	2016	5219	b
5834	376783	Onur	Bilgin	Onur Bilgin	2014	20100	
5923	1063540	Mamadu	Silla	Mamadu Silla	2022	1147	
6056	168374	Mykhaylo	Replyuk	Mykhaylo Replyuk	2014	6996	
6142	199731	Tommaso	Barluzzi	Tommaso Barluzzi	2012	398	
6186	216945	Sam	Van Den Abeel	Sam Van Den Abeel	2012	566	s
			aylor	Mark Taylor	2012	2553	

	player_id	first_name	last_name	name	last_season	current_club_id	
6368	290083	Souleymane Baba	Diomandé	Souleymane Baba Diomandé	2014	204	ba
6494	347209	Bradley	Vanneste	Bradley Vanneste	2014	601	brad
6650	435666	Stanislav	Kotyo	Stanislav Kotyo	2015	6996	st
6731	501614	Roman	Ilnytskyi	Roman Ilnytskyi	2016	4482	ro
6743	509125	Lucas	Morales	Lucas Morales	2017	29228	l
6778	533147	Ortwin	Helderwert	Ortwin Helderwert	2016	498	
6812	556166	Jacky	Ullrich	Jacky Ullrich	2016	172	
7003	726466	Panagiotis	Anagnostopoulos	Panagiotis Anagnostopoulos	2022	3999	anag
7150	14821	David	van Zanten	David van Zanten	2013	465	
7404	74860	Michael	Tidser	Michael Tidser	2013	2759	r
7867	13723	Emmerik	De Vriese	Emmerik De Vriese	2012	2727	
7897	49474	Shamil	Lakhiyalov	Shamil Lakhiyalov	2012	2696	shar
8137	256560	Gerasimos	Panagakis	Gerasimos Panagakis	2012	6676	
8173	273869	Brahim	Barmaki	Brahim Barmaki	2012	1411	bra
8409	389564	NaN	Cléber	Cléber	2016	1085	
8539	468661	Lucas	Triki	Lucas Triki	2016	595	
8549	477799	Antonis	Gerekos	Antonis Gerekos	2016	5219	ant
8837	830225	Maksym	Dubenkov	Maksym Dubenkov	2019	18303	
8856	889717	Pablo	Montero	Pablo Montero	2021	3709	pa
9080	41080	Rok	Straus	Rok Straus	2014	5220	
9280	90591	Vincent	Martin	Vincent Martin	2014	29228	v
9373	128103	Steven	Ross	Steven Ross	2014	2759	
			olini	Gabriele Angiolini	2013	1210	

	player_id	first_name	last_name	name	last_season	current_club_id	
9745	564561	Mathurin	Sakho	Mathurin Sakho	2018	1416	ma
9997	175664	Gerson	Fidalgo	Gerson Fidalgo	2013	4750	g
10151	297015	Roman	Shpytsya	Roman Shpytsya	2013	2227	ror
10180	319299	Panagiotis	Grosios	Panagiotis Grosios	2017	6676	
10348	398429	Alexandros	Grigorakis	Alexandros Grigorakis	2016	5219	
10425	459322	Callum	Rowe	Callum Rowe	2020	405	
10831	1127980	Jamie	Williamson	Jamie Williamson	2022	2759	jam
11235	90596	Heidi	Oudina	Heidi Oudina	2014	29228	
11478	191179	NaN	Douglas Pará	Douglas Pará	2012	982	
11802	553075	Jari	De Roover	Jari De Roover	2017	3508	j
11803	553380	Anatoliy	Ulyanov	Anatoliy Ulyanov	2017	16247	ana
11889	51190	Marc	Vucinovic	Marc Vucinovic	2014	127	m
12010	296674	Kenneth	Fournier	Kenneth Fournier	2013	601	ken
12046	321086	Emil	Nielsen	Emil Nielsen	2012	2414	
12188	386554	Ghjuvan Andria	Piereschi	Ghjuvan Andria Piereschi	2015	3558	gh
12223	402961	Nikolaos	Lazaridis	Nikolaos Lazaridis	2018	3060	nikc
12295	457583	Dimitri	Swisch	Dimitri Swisch	2013	9007	c
12359	511121	Roman	Babyak	Roman Babyak	2016	4482	rc
12386	533734	Klinton	Noka	Klinton Noka	2017	5219	
12565	673977	Eric	Asomani	Eric Asomani	2019	28643	
12862	34060	Nejmeddin	Daghfous	Nejmeddin Daghfous	2012	39	
13385	231909	Emilio	Kishta	Emilio Kishta	2013	28956	

	player_id	first_name	last_name	name	last_season	current_club_id	
13396	238488	NaN	Leonardo	Leonardo	2012	2424	
13608	342138	Rodion	Budchenko	Rodion Budchenko	2014	6993	
14107	402964	Marios	Pavlis	Marios Pavlis	2018	3060	
14112	405800	Thomas	Biancardini	Thomas Biancardini	2015	3558	
14323	578631	Bas	Breukers	Bas Breukers	2018	385	
14830	54679	Burim	Hashani	Burim Hashani	2018	126	b
15120	158704	Jon Ander	Garrido	Jon Ander Garrido	2022	2687	
15203	193078	Zaine	Francis-Angol	Zaine Francis-Angol	2014	987	
15374	256954	Stratos	Bentas	Stratos Bentas	2013	441	s
15534	332928	Georgios	Litskas	Georgios Litskas	2014	7185	ge
15554	339560	Abdul	Razak	Abdul Razak	2014	28956	
15672	402050	Julian	de Blasis	Julian de Blasis	2015	6676	ju
15685	412951	Lorenzo	van Kleef	Lorenzo van Kleef	2019	1268	
15866	56607	Anco	Jansen	Anco Jansen	2020	1283	
15934	246267	NaN	Felipe Desco	Felipe Desco	2012	1436	
16259	690704	Grégoire	Guiot	Grégoire Guiot	2021	2861	g
16279	718190	Jorge	Cofrades	Jorge Cofrades	2020	5358	jc
16337	815888	Cem-Ali	Dogan	Cem-Ali Dogan	2021	10	c
16590	41261	Habib	Habibou	Habib Habibou	2016	273	h
16766	85670	Lars	Hutten	Lars Hutten	2018	385	
16769	86158	Onur	Cenik	Onur Cenik	2014	1506	
17211	271432	Sergiy	Kurta	Sergiy Kurta	2012	6996	
			mars	Florimond Smars	2016	172	flor

	player_id	first_name	last_name	name	last_season	current_club_id	
17797	57333	Jergé	Hoefdraad	Jergé Hoefdraad	2015	133	jer
18257	1054052	Paul	Nsio	Paul Nsio	2022	124	
18478	45611	Leigh	Griffiths	Leigh Griffiths	2021	511	
18652	90587	Jerome	Leblois	Jerome Leblois	2014	29228	je
18921	189549	Denys	Bilous	Denys Bilous	2014	2783	
19012	227367	Joppe	Janssens	Joppe Janssens	2012	498	jo
19112	269674	Erdem	Kökcam	Erdem Kökcam	2012	6890	er
19135	276507	Stef	Vervoort	Stef Vervoort	2014	968	
19157	286042	Ayron	Verkindere	Ayron Verkindere	2014	520	ayro
19189	297002	Izzet	Bilyalov	Izzet Bilyalov	2013	2227	
19305	349917	Aleksandr	Chibirov	Aleksandr Chibirov	2014	3729	
19312	354464	Nugzar	Muzashvili	Nugzar Muzashvili	2014	7185	
19486	477185	Bogdan	Gladun	Bogdan Gladun	2016	4482	bo
19709	4429	Matthew	Amoah	Matthew Amoah	2013	1304	ma
20500	90593	Cedric	Meurisse	Cedric Meurisse	2014	29228	ce
20768	193585	Nicolas	Sumsky	Nicolas Sumsky	2014	2999	ni
20955	270678	Victor	Rousseau	Victor Rousseau	2012	3911	vi
21051	331550	Lionel Nshole	Makondi	Lionel Nshole Makondi	2014	354	l
21294	462784	Jourdain	Assen	Jourdain Assen	2016	6676	jo
21850	280139	Theodoros	Tsikoudakis	Theodoros Tsikoudakis	2013	21957	
22048	1111275	Maksim	Kardavar	Maksim Kardavar	2013	2227	mak
22049	1127979	Zach	MacPhee	Zach MacPhee	2022	2759	z
			Malki	Sanharib Malki	2015	10484	sa

	player_id	first_name	last_name	name	last_season	current_club_id	
22154	26199	NaN	Verza	Verza	2017	3368	
22610	164600	Morten	Hansen	Morten Hansen	2012	5817	m
22795	244915	Oleksiy	Zinkevych	Oleksiy Zinkevych	2017	18303	
23095	381920	Artem	Zakharov	Artem Zakharov	2015	6994	art
23197	451848	Geoffrey	Acheampong	Geoffrey Acheampong	2016	595	
23229	477800	Dimitrios	Trepeklis	Dimitrios Trepeklis	2017	5219	
23414	644144	Jawad	Dramé	Jawad Dramé	2018	417	
24328	125829	Adem	Candir	Adem Candir	2012	1506	
24505	193803	Ruben	Ketels	Ruben Ketels	2012	498	
24791	318394	Stamatis	Togrou	Stamatis Togrou	2013	6676	sta
25137	509124	Thomas	Demol	Thomas Demol	2017	29228	th
26178	90602	Saverio	Vitelli	Saverio Vitelli	2014	29228	
26484	213306	Tiziano	Cherubini	Tiziano Cherubini	2012	2921	tizia
26648	284486	Grigoris	Chandrinos	Grigoris Chandrinos	2014	5219	
26665	290558	Georgios	Makrostergios	Georgios Makrostergios	2017	6676	r
26682	297017	Kostyantyn	Sumelidi	Kostyantyn Sumelidi	2013	2227	
26705	318354	Christos	Karydopoulos	Christos Karydopoulos	2013	6676	k
27035	534678	Grigoris	Fotopoulos	Grigoris Fotopoulos	2018	6676	
27107	584378	Lucas	Geurde	Lucas Geurde	2017	1245	
27248	727111	Anastasios	Papadopoulos	Anastasios Papadopoulos	2019	128	p
27372	1015619	Ahmed	Abdullahi	Ahmed Abdullahi	2022	157	ahm
27380	1063517	Rayan Edrees	Fallatah	Rayan Edrees Fallatah	2022	653	r

	player_id	first_name	last_name	name	last_season	current_club_id	
27385	1086787	Demba	N'Diaye	Demba N'Diaye	2022	1420	d
27395	1532	Wesley	Sonck	Wesley Sonck	2012	28643	
27634	52045	Athanasios	Panteliadis	Athanasios Panteliadis	2018	2672	
28633	321600	Giannis-Alexandros	Stogios	Giannis-Alexandros Stogios	2014	7185	
28637	324148	Thomas	Michot	Thomas Michot	2014	28643	th
28739	375658	Oleg	Yefimchuk	Oleg Yefimchuk	2014	2783	ol
28846	457129	Ben	Hinchliffe	Ben Hinchliffe	2016	3008	ł
28972	567449	Pascal	Debacker	Pascal Debacker	2016	172	pas
29466	232598	Mulumba	Mukendi	Mulumba Mukendi	2013	12438	
29479	261071	James	Efmorfidis	James Efmorfidis	2020	235	jam
29562	539732	Yuriy	Berdey	Yuriy Berdey	2017	26459	
30007	265396	NaN	Pepe	Pepe	2012	4750	
30119	163058	NaN	Fabinho	Fabinho	2013	410	
30169	256011	Maxime	Medaglia	Maxime	2012	14171	

In [67]:
```
missing_position_df.head(173)
```

Out[67]:

	player_id	first_name	last_name	name	last_season	current_club_id	
86	22395	Adil	Hermach	Adil Hermach	2012	415	
282	64385	José Manuel	Catalá	José Manuel Catalá	2014	2079	
353	90600	Josselin	Thibaut	Josselin Thibaut	2014	29228	jo
540	168887	Gianni	Van Landschoot	Gianni Van Landschoot	2012	520	
586	189507	Jonathan	Black	Jonathan Black	2014	2760	jo
614	199730	Claudio	Andreoli	Claudio Andreoli	2012	398	cla
718	244258	Dmytro	Lyakh	Dmytro Lyakh	2013	2227	
1228	540580	Maksym	Sorochenko	Maksym Sorochenko	2016	4482	
1463	936901	Dominic	Sadi	Dominic Sadi	2022	989	
1772	182689	Rafail	Soukias	Rafail Soukias	2014	21957	
2419	193955	Stefanos	Martsakis	Stefanos Martsakis	2014	653	
2498	225546	Marvin	Novotny	Marvin Novotny	2012	192	ma
2601	271446	NaN	Kiu	Kiu	2013	3302	
2675	298659	Vasyl	Grinka	Vasyl Grinka	2013	2227	
2939	464142	Diego	Snepvangers	Diego Snepvangers	2017	132	
3092	598917	Georgios	Christopoulos	Georgios Christopoulos	2018	441	
3109	612809	Milan	Govaers	Milan Govaers	2022	2861	r
3154	652672	Kai	Fotheringham	Kai Fotheringham	2022	1519	kai-
3433	33100	Suleiman	Omo	Suleiman Omo	2018	2672	s
3778	273542	Noureddine	Boutzamar	Noureddine Boutzamar	2015	133	
4297	210202	James	Horsfield	James Horsfield	2018	511	ja
4328	221883	Abdul	Basit	Abdul Basit	2016	1301	

	player_id	first_name	last_name	name	last_season	current_club_id	
4347	231165	Fotis	Doumanis	Fotis Doumanis	2013	128	fc
4430	266911	Georgios	Peppas	Georgios Peppas	2013	2672	geo
4625	348269	Oleksandr	Dekhtiarenko	Oleksandr Dekhtiarenko	2014	6993	
4975	584434	Adrian	Imeri	Adrian Imeri	2018	3060	
5673	118645	Branislav	Nikic	Branislav Nikic	2016	5219	b
5834	376783	Onur	Bilgin	Onur Bilgin	2014	20100	
5923	1063540	Mamadu	Silla	Mamadu Silla	2022	1147	
6056	168374	Mykhaylo	Replyuk	Mykhaylo Replyuk	2014	6996	
6142	199731	Tommaso	Barluzzi	Tommaso Barluzzi	2012	398	
6186	216945	Sam	Van Den Abeel	Sam Van Den Abeel	2012	566	s
6217	235782	Mark	Taylor	Mark Taylor	2012	2553	
6368	290083	Souleymane Baba	Diomandé	Souleymane Baba Diomandé	2014	204	ba
6494	347209	Bradley	Vanneste	Bradley Vanneste	2014	601	brad
6650	435666	Stanislav	Kotyo	Stanislav Kotyo	2015	6996	st
6731	501614	Roman	Ilnytskyi	Roman Ilnytskyi	2016	4482	ro
6743	509125	Lucas	Morales	Lucas Morales	2017	29228	l
6778	533147	Ortwin	Helderwert	Ortwin Helderwert	2016	498	
6812	556166	Jacky	Ullrich	Jacky Ullrich	2016	172	
7003	726466	Panagiotis	Anagnostopoulos	Panagiotis Anagnostopoulos	2022	3999	anag
7150	14821	David	van Zanten	David van Zanten	2013	465	
7375	66112	Keko	Gontán	Keko Gontán	2018	366	
7404	74860	Michael	Tidser	Michael Tidser	2013	2759	m
			riese	Emmerik De Vriese	2012	2727	

	player_id	first_name	last_name	name	last_season	current_club_id	
7897	49474	Shamil	Lakhiyalov	Shamil Lakhiyalov	2012	2696	shar
8137	256560	Gerasimos	Panagakis	Gerasimos Panagakis	2012	6676	
8173	273869	Brahim	Barmaki	Brahim Barmaki	2012	1411	bra
8409	389564	NaN	Cléber	Cléber	2016	1085	
8539	468661	Lucas	Triki	Lucas Triki	2016	595	
8549	477799	Antonis	Gerekos	Antonis Gerekos	2016	5219	ant
8837	830225	Maksym	Dubenkov	Maksym Dubenkov	2019	18303	
8856	889717	Pablo	Montero	Pablo Montero	2021	3709	pa
9080	41080	Rok	Straus	Rok Straus	2014	5220	
9280	90591	Vincent	Martin	Vincent Martin	2014	29228	v
9373	128103	Steven	Ross	Steven Ross	2014	2759	
9548	199196	Gabriele	Angiolini	Gabriele Angiolini	2013	1210	
9745	564561	Mathurin	Sakho	Mathurin Sakho	2018	1416	ma
9765	654947	Dylan	Demuynck	Dylan Demuynck	2022	3508	dyl
9997	175664	Gerson	Fidalgo	Gerson Fidalgo	2013	4750	g
10119	288005	Joaquín	Arzura	Joaquín Arzura	2020	6418	jo
10151	297015	Roman	Shpytsya	Roman Shpytsya	2013	2227	ror
10180	319299	Panagiotis	Grosios	Panagiotis Grosios	2017	6676	
10348	398429	Alexandros	Grigorakis	Alexandros Grigorakis	2016	5219	
10425	459322	Callum	Rowe	Callum Rowe	2020	405	
10831	1127980	Jamie	Williamson	Jamie Williamson	2022	2759	jam
11235	90596	Heidi	Oudina	Heidi Oudina	2014	29228	

	player_id	first_name	last_name	name	last_season	current_club_id	
11478	191179	NaN	Douglas Pará	Douglas Pará	2012	982	
11542	220769	Nikola	Asceric	Nikola Asceric	2017	3999	ı
11802	553075	Jari	De Roover	Jari De Roover	2017	3508	j
11803	553380	Anatoliy	Ulyanov	Anatoliy Ulyanov	2017	16247	ana
11889	51190	Marc	Vucinovic	Marc Vucinovic	2014	127	m
12010	296674	Kenneth	Fournier	Kenneth Fournier	2013	601	ken
12046	321086	Emil	Nielsen	Emil Nielsen	2012	2414	
12188	386554	Ghjuvan Andria	Piereschi	Ghjuvan Andria Piereschi	2015	3558	gh
12223	402961	Nikolaos	Lazaridis	Nikolaos Lazaridis	2018	3060	nikc
12295	457583	Dimitri	Swisch	Dimitri Swisch	2013	9007	c
12359	511121	Roman	Babyak	Roman Babyak	2016	4482	rc
12386	533734	Klinton	Noka	Klinton Noka	2017	5219	
12565	673977	Eric	Asomani	Eric Asomani	2019	28643	
12862	34060	Nejmeddin	Daghfous	Nejmeddin Daghfous	2012	39	
13385	231909	Emilio	Kishta	Emilio Kishta	2013	28956	
13396	238488	NaN	Leonardo	Leonardo	2012	2424	
13608	342138	Rodion	Budchenko	Rodion Budchenko	2014	6993	
14107	402964	Marios	Pavlis	Marios Pavlis	2018	3060	
14112	405800	Thomas	Biancardini	Thomas Biancardini	2015	3558	
14323	578631	Bas	Breukers	Bas Breukers	2018	385	
14830	54679	Burim	Hashani	Burim Hashani	2018	126	b
15120	158704	Jon Ander	Garrido	Jon Ander Garrido	2022	2687	
			ngol	Zaine Francis-Angol	2014	987	

	player_id	first_name	last_name	name	last_season	current_club_id	
15374	256954	Stratos	Bentas	Stratos Bentas	2013	441	s
15534	332928	Georgios	Litskas	Georgios Litskas	2014	7185	ge
15554	339560	Abdul	Razak	Abdul Razak	2014	28956	
15672	402050	Julian	de Blasis	Julian de Blasis	2015	6676	ju
15685	412951	Lorenzo	van Kleef	Lorenzo van Kleef	2019	1268	
15866	56607	Anco	Jansen	Anco Jansen	2020	1283	
15934	246267	NaN	Felipe Desco	Felipe Desco	2012	1436	
16246	671068	Bassim	Boukteb	Bassim Boukteb	2021	2715	bas
16259	690704	Grégoire	Guiot	Grégoire Guiot	2021	2861	g
16279	718190	Jorge	Cofrades	Jorge Cofrades	2020	5358	jc
16337	815888	Cem-Ali	Dogan	Cem-Ali Dogan	2021	10	c
16590	41261	Habib	Habibou	Habib Habibou	2016	273	h
16766	85670	Lars	Hutten	Lars Hutten	2018	385	
16769	86158	Onur	Cenik	Onur Cenik	2014	1506	
17211	271432	Sergiy	Kurta	Sergiy Kurta	2012	6996	
17564	447143	Florimond	Smars	Florimond Smars	2016	172	flor
17797	57333	Jergé	Hoefdraad	Jergé Hoefdraad	2015	133	jer
18257	1054052	Paul	Nsio	Paul Nsio	2022	124	
18262	1094044	Zacharias	Papadimitriou	Zacharias Papadimitriou	2023	3385	p
18478	45611	Leigh	Griffiths	Leigh Griffiths	2021	511	
18652	90587	Jerome	Leblois	Jerome Leblois	2014	29228	je
18921	189549	Denys	Bilous	Denys Bilous	2014	2783	
			sens	Joppe Janssens	2012	498	jo

	player_id	first_name	last_name	name	last_season	current_club_id	
19112	269674	Erdem	Kökcam	Erdem Kökcam	2012	6890	er
19135	276507	Stef	Vervoort	Stef Vervoort	2014	968	
19157	286042	Ayron	Verkindere	Ayron Verkindere	2014	520	ayro
19189	297002	Izzet	Bilyalov	Izzet Bilyalov	2013	2227	
19305	349917	Aleksandr	Chibirov	Aleksandr Chibirov	2014	3729	
19312	354464	Nugzar	Muzashvili	Nugzar Muzashvili	2014	7185	
19486	477185	Bogdan	Gladun	Bogdan Gladun	2016	4482	bo
19709	4429	Matthew	Amoah	Matthew Amoah	2013	1304	ma
20500	90593	Cedric	Meurisse	Cedric Meurisse	2014	29228	ce
20768	193585	Nicolas	Sumsky	Nicolas Sumsky	2014	2999	ni
20955	270678	Victor	Rousseau	Victor Rousseau	2012	3911	vi
21051	331550	Lionel Nshole	Makondi	Lionel Nshole Makondi	2014	354	l
21294	462784	Jourdain	Assen	Jourdain Assen	2016	6676	jo
21850	280139	Theodoros	Tsikoudakis	Theodoros Tsikoudakis	2013	21957	
22048	1111275	Maksim	Kardavar	Maksim Kardavar	2013	2227	mak
22049	1127979	Zach	MacPhee	Zach MacPhee	2022	2759	z
22131	20351	Sanharib	Malki	Sanharib Malki	2015	10484	sa
22154	26199	NaN	Verza	Verza	2017	3368	
22610	164600	Morten	Hansen	Morten Hansen	2012	5817	m
22795	244915	Oleksiy	Zinkevych	Oleksiy Zinkevych	2017	18303	
23095	381920	Artem	Zakharov	Artem Zakharov	2015	6994	art
23197	451848	Geoffrey	Acheampong	Geoffrey Acheampong	2016	595	

	player_id	first_name	last_name	name	last_season	current_club_id	
23229	477800	Dimitrios	Trepeklis	Dimitrios Trepeklis	2017	5219	
23414	644144	Jawad	Dramé	Jawad Dramé	2018	417	
24328	125829	Adem	Candir	Adem Candir	2012	1506	
24505	193803	Ruben	Ketels	Ruben Ketels	2012	498	
24791	318394	Stamatis	Togrou	Stamatis Togrou	2013	6676	sta
25137	509124	Thomas	Demol	Thomas Demol	2017	29228	th
25359	733434	Jack	Wadham	Jack Wadham	2022	989	
26178	90602	Saverio	Vitelli	Saverio Vitelli	2014	29228	
26294	131480	Il-Iok	Yun	Il-Iok Yun	2020	969	
26484	213306	Tiziano	Cherubini	Tiziano Cherubini	2012	2921	tizia
26648	284486	Grigoris	Chandrinos	Grigoris Chandrinos	2014	5219	
26665	290558	Georgios	Makrostergios	Georgios Makrostergios	2017	6676	r
26682	297017	Kostyantyn	Sumelidi	Kostyantyn Sumelidi	2013	2227	
26705	318354	Christos	Karydopoulos	Christos Karydopoulos	2013	6676	
26756	340063	Gianluca	Piccoli	Gianluca Piccoli	2018	416	gia
27035	534678	Grigoris	Fotopoulos	Grigoris Fotopoulos	2018	6676	
27107	584378	Lucas	Geurde	Lucas Geurde	2017	1245	
27248	727111	Anastasios	Papadopoulos	Anastasios Papadopoulos	2019	128	p
27268	746999	Fergus	Dee	Fergus Dee	2020	2553	
27372	1015619	Ahmed	Abdullahi	Ahmed Abdullahi	2022	157	ahm
27380	1063517	Rayan Edrees	Fallatah	Rayan Edrees Fallatah	2022	653	r
27385	1086787	Demba	N'Diaye	Demba N'Diaye	2022	1420	d
			onck	Wesley Sonck	2012	28643	

	player_id	first_name	last_name	name	last_season	current_club_id	
27634	52045	Athanasios	Panteliadis	Athanasios Panteliadis	2018	2672	
28593	297241	Liam	Burt	Liam Burt	2016	124	
28633	321600	Giannis-Alexandros	Stogios	Giannis-Alexandros Stogios	2014	7185	
28637	324148	Thomas	Michot	Thomas Michot	2014	28643	th
28739	375658	Oleg	Yefimchuk	Oleg Yefimchuk	2014	2783	ol
28846	457129	Ben	Hinchliffe	Ben Hinchliffe	2016	3008	ł
28972	567449	Pascal	Debacker	Pascal Debacker	2016	172	pas
29372	1031590	Sotiris	Kontouris	Sotiris Kontouris	2023	6418	sot
29466	232598	Mulumba	Mukendi	Mulumba Mukendi	2013	12438	
29479	261071	James	Efmorfidis	James Efmorfidis	2020	235	jam
29562	539732	Yuriy	Berdey	Yuriy Berdey	2017	26459	
30007	265396	NaN	Pepe	Pepe	2012	4750	
30119	163058	NaN	Fabinho	Fabinho	2013	410	
30169	256011	Maxime	Medaglia	Maxime	2012	14171	

In [68]:
```
players_df[(players_df['position'] == 'Missing') & (players_df['sub_position'].isnu
```

Out[68]:

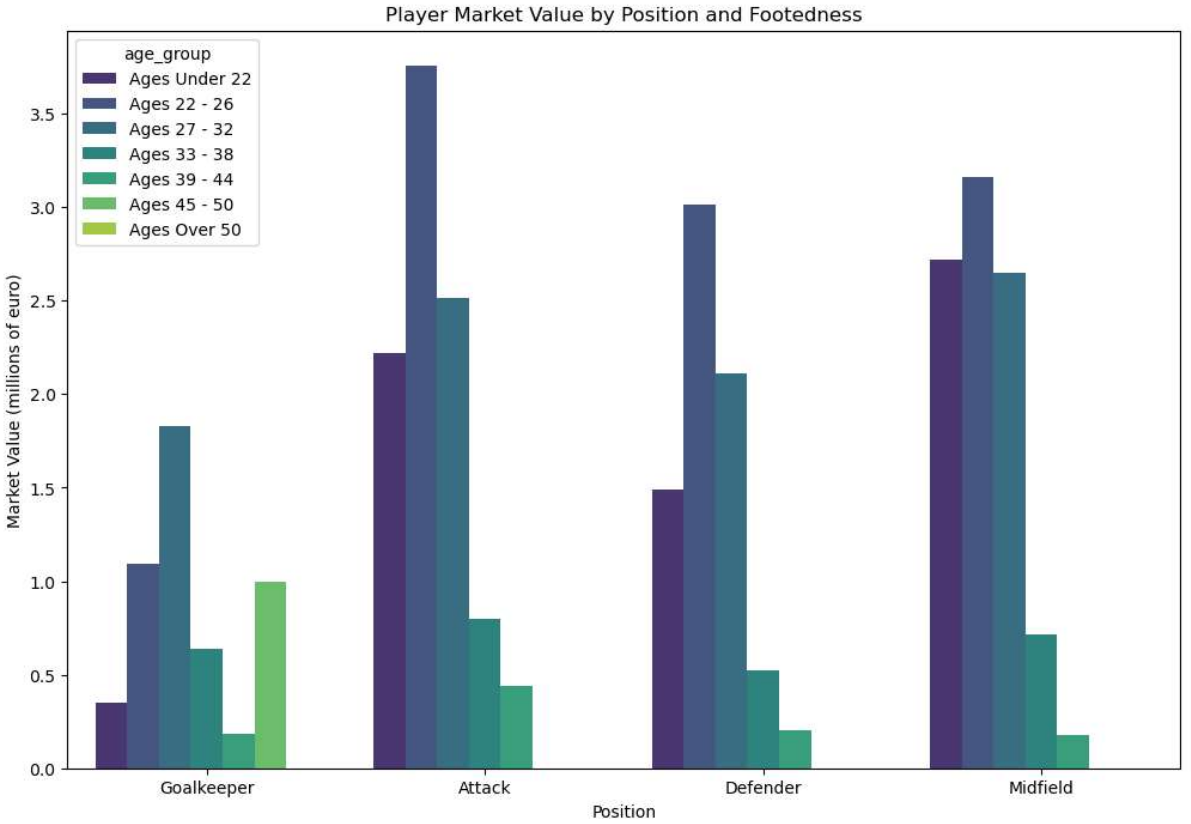
player_id	first_name	last_name	name	last_season	current_club_id	player_code	country_of_bir

Unfortunately there are no sub_positions to use for imputing missing positions

In [69]:
```
# Removing entries with missing position
players_df = players_df[players_df['position'] != 'Missing']
```

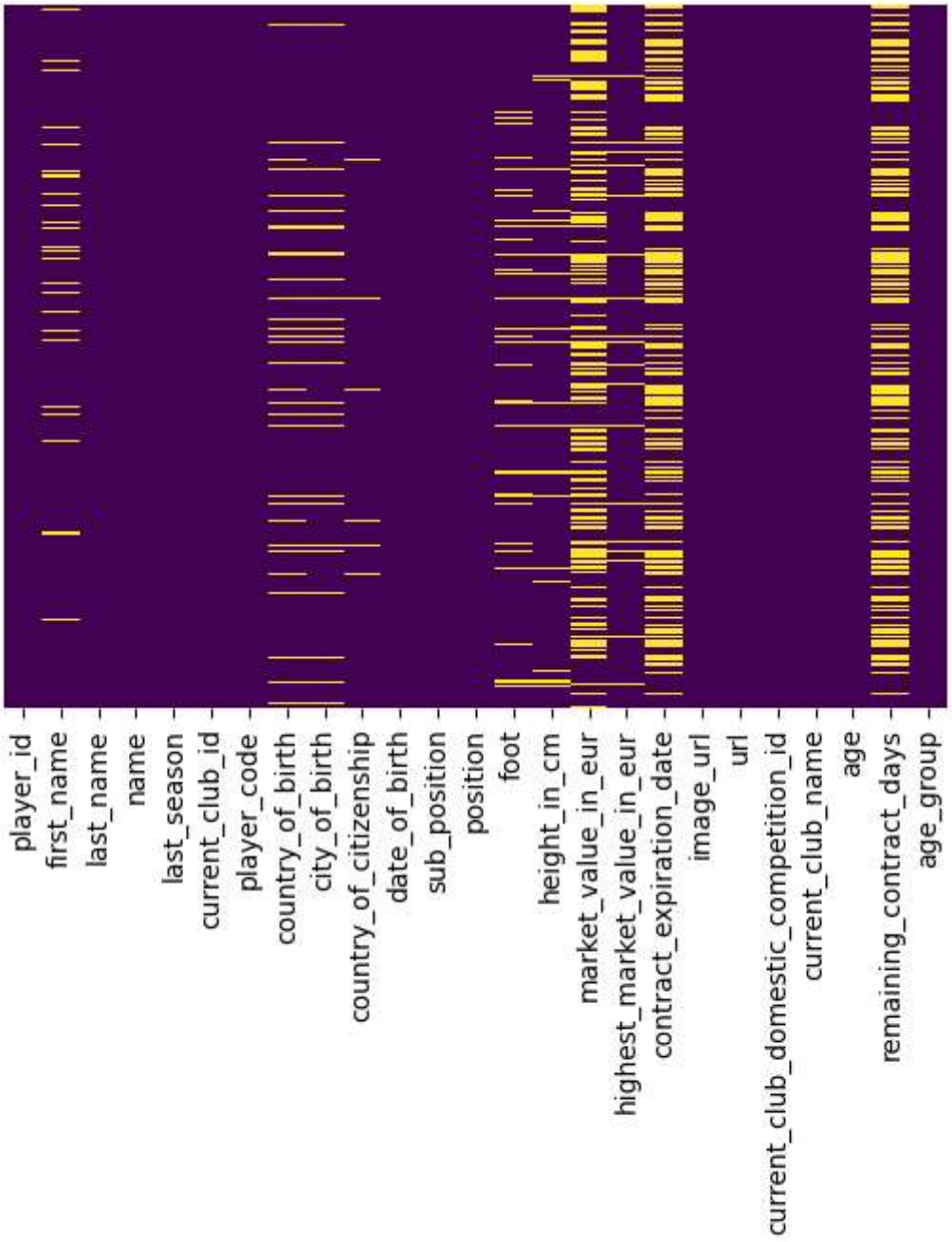
Addressing the Missing values for 'market_value_in_eur'

In [70]:
```
plt.figure(figsize=(12, 8))
sns.barplot(x = 'position' , y=players_df['market_value_in_eur']/1000000, hue= 'age
plt.title("Player Market Value by Position and Footedness")
plt.ylabel('Market Value (millions of euro)')
plt.xlabel('Position')
plt.show()
```



In [71]:
```
sns.heatmap(players_df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[71]:
```
<Axes: >
```



In [72]:
```
players_df['market_value_in_eur'].isnull().sum()
```

Out[72]:
```
10762
```

In [73]:
```
players_df['market_value_in_eur'].describe()
```

Out[73]:
```
count    1.936800e+04
mean     2.236328e+06
std      7.343296e+06
min      1.000000e+04
25%      1.750000e+05
50%      3.500000e+05
75%      1.000000e+06
max      1.800000e+08
Name: market_value_in_eur, dtype: float64
```

In [74]:
```
# Calculating the mean market values based on position and age category
mean_values = players_df.groupby(['position', 'age_group'])['market_value_in_eur'].

# Displaying the mean values
```

	position	age_group	market_value_in_eur
0	Attack	Ages 22 - 26	3.756665e+06
1	Attack	Ages 27 - 32	2.517620e+06
2	Attack	Ages 33 - 38	7.983708e+05
3	Attack	Ages 39 - 44	4.390385e+05
4	Attack	Ages 45 - 50	NaN
5	Attack	Ages Over 50	NaN
6	Attack	Ages Under 22	2.217676e+06
7	Defender	Ages 22 - 26	3.017293e+06
8	Defender	Ages 27 - 32	2.113122e+06
9	Defender	Ages 33 - 38	5.213000e+05
10	Defender	Ages 39 - 44	2.053261e+05
11	Defender	Ages 45 - 50	NaN
12	Defender	Ages Under 22	1.489432e+06
13	Goalkeeper	Ages 22 - 26	1.093938e+06
14	Goalkeeper	Ages 27 - 32	1.829864e+06
15	Goalkeeper	Ages 33 - 38	6.415633e+05
16	Goalkeeper	Ages 39 - 44	1.860000e+05
17	Goalkeeper	Ages 45 - 50	1.000000e+06
18	Goalkeeper	Ages Over 50	NaN
19	Goalkeeper	Ages Under 22	3.484932e+05
20	Midfield	Ages 22 - 26	3.162304e+06
21	Midfield	Ages 27 - 32	2.646813e+06
22	Midfield	Ages 33 - 38	7.153819e+05
23	Midfield	Ages 39 - 44	1.768868e+05
24	Midfield	Ages 45 - 50	NaN
25	Midfield	Ages Under 22	2.719000e+06

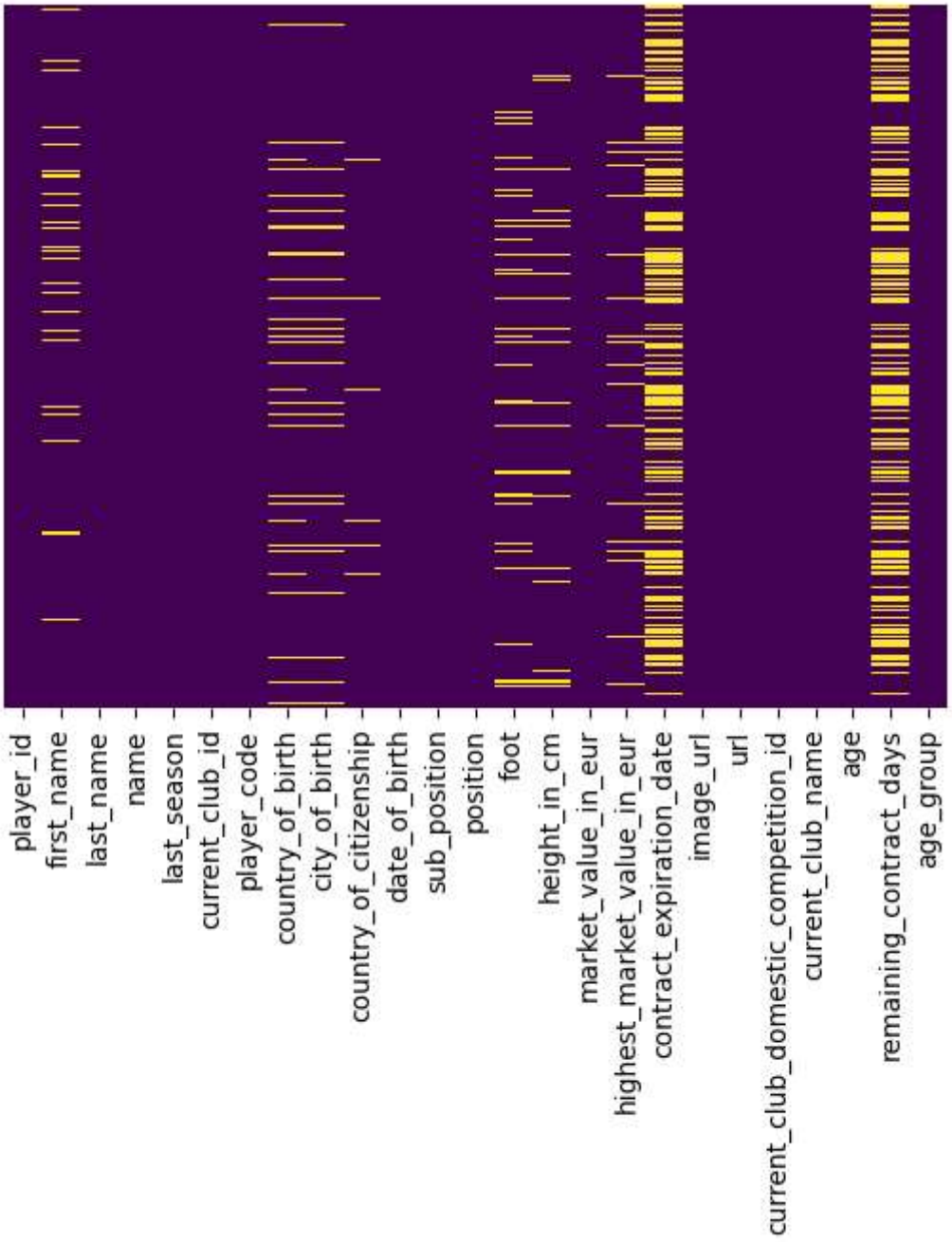
In [75]:
```
# Writing a function of conditional statements for imputing the mean market values
def impute_market_value(cols):
    market_value_in_eur = cols[0]
    position = cols[1]
    age_group = cols[2]

    if pd.isnull(market_value_in_eur):

        if position == 'Attack':
            if age_group == 'Ages 22 - 26':
                return 3.761589e+06
            elif age_group == 'Ages 27 - 32':
                return 2.519384e+06
            elif age_group == 'Ages 33 - 38':
                return 7.900107e+05
            elif age_group == 'Ages 39 - 44':
                return 4.437255e+05
            elif age_group == 'Ages Under 22':
                return 2.209577e+06
            else:
                return 2.236328e+06

        elif position == 'Defender':
            if age_group == 'Ages 22 - 26':
                return 2.989714e+06
            elif age_group == 'Ages 27 - 32':
                return 2.109582e+06
            elif age_group == 'Ages 33 - 38':
                return 5.224117e+05
            elif age_group == 'Ages 39 - 44':
                return 2.053261e+05
            elif age_group == 'Ages Under 22':
                return 1.579725e+06
            else:
```

```
        elif position == 'Goalkeeper':
            if age_group == 'Ages 22 - 26':
                return 1.093724e+06
            elif age_group == 'Ages 27 - 32':
                return 1.836525e+06
            elif age_group == 'Ages 33 - 38':
                return 6.415633e+05
            elif age_group == 'Ages 39 - 44':
                return 1.860000e+05
            elif age_group == 'Ages 45 - 50':
                return 1.000000e+06
            elif age_group == 'Ages Under 22':
                return 3.466327e+05
            else:
                return 2.236328e+06

        elif position == 'Midfield':
            if age_group == 'Ages 22 - 26':
                return 3.168564e+06
            elif age_group == 'Ages 27 - 32':
                return 2.641277e+06
            elif age_group == 'Ages 33 - 38':
                return 7.158922e+05
            elif age_group == 'Ages 39 - 44':
                return 1.768868e+05
            elif age_group == 'Ages Under 22':
                return 2.715097e+06
            else:
                return 2.236328e+06

        else:
            return 2.236328e+06

    else:
        return market_value_in_eur
```

In [76]:
```
# Applying the function to the players df
players_df['market_value_in_eur'] = players_df[['market_value_in_eur','position','a
```

In [77]:
```
sns.heatmap(players_df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[77]:
```
<Axes: >
```



In [78]:
```
players_df['market_value_in_eur'].isnull().sum()
```

Out[78]:
```
0
```

Addressing the Missing values for 'highest_market_value_in_eur'

In [79]:
```
players_df['highest_market_value_in_eur'].value_counts()
```

Out[79]:

highest_market_value_in_eur	
100000.0	1325
300000.0	1319
50000.0	1302
1000000.0	1257
200000.0	1242
500000.0	1238
400000.0	1195
150000.0	1059
1500000.0	1056
2000000.0	1005
250000.0	918
600000.0	824
800000.0	809
3000000.0	776
2500000.0	721
700000.0	638
350000.0	602
4000000.0	592
5000000.0	579
75000.0	496
3500000.0	461
900000.0	442
1200000.0	414
750000.0	373
450000.0	364
6000000.0	360
10000000.0	357
7000000.0	289
8000000.0	284
25000.0	282
125000.0	269
4500000.0	259
1800000.0	228
15000000.0	226
1250000.0	225
175000.0	210
12000000.0	207
650000.0	195
20000000.0	184
9000000.0	183
7500000.0	155
25000000.0	147
18000000.0	139
1300000.0	136
5500000.0	134
850000.0	134
225000.0	129
30000000.0	123
550000.0	121
6500000.0	120
1750000.0	116
275000.0	108
1400000.0	107
2200000.0	101
14000000.0	89
40000000.0	86
35000000.0	83
22000000.0	79
13000000.0	78
1600000.0	78
2800000.0	76
1100000.0	74

11000000.0	72
325000.0	70
1700000.0	69
50000000.0	66
8500000.0	61
2300000.0	51
17000000.0	49
375000.0	48
45000000.0	44
2250000.0	41
28000000.0	35
60000000.0	34
2750000.0	34
70000000.0	33
3200000.0	30
425000.0	29
3800000.0	28
2600000.0	26
2400000.0	26
32000000.0	26
475000.0	24
10000.0	23
90000000.0	22
55000000.0	22
4200000.0	22
2700000.0	21
19000000.0	21
65000000.0	19
80000000.0	18
1900000.0	18
24000000.0	18
100000000.0	18
3250000.0	17
3300000.0	17
3750000.0	17
23000000.0	17
9500000.0	17
17500000.0	16
12500000.0	16
75000000.0	15
21000000.0	13
3400000.0	13
38000000.0	12
950000.0	12
42000000.0	11
27000000.0	11
525000.0	10
10500000.0	10
2100000.0	9
11500000.0	9
150000000.0	9
26000000.0	9
3600000.0	9
4800000.0	9
3700000.0	8
4250000.0	8
575000.0	8
675000.0	8
5750000.0	8
4300000.0	6
18500000.0	6
4600000.0	6
85000000.0	6

```
5200000.0         6
5250000.0         6
4750000.0         5
13500000.0        5
1350000.0         5
625000.0          4
48000000.0        4
110000000.0       4
825000.0          4
28500000.0        4
1650000.0         4
5800000.0         4
120000000.0       4
6250000.0         3
7250000.0         3
37000000.0        3
3100000.0         3
5300000.0         3
36500000.0        3
1150000.0         3
4400000.0         3
14500000.0        3
180000000.0       3
16500000.0        3
1850000.0         2
5600000.0         2
21500000.0        2
24500000.0        2
31000000.0        2
6700000.0         2
6750000.0         2
6200000.0         2
4100000.0         2
36000000.0        2
8250000.0         2
31500000.0        2
8800000.0         2
775000.0          2
925000.0          2
12250000.0        1
725000.0          1
27500000.0        1
47500000.0        1
3880000.0         1
3850000.0         1
8900000.0         1
26500000.0        1
11800000.0        1
1450000.0         1
4700000.0         1
875000.0          1
2650000.0         1
9700000.0         1
17250000.0        1
1040000.0         1
8200000.0         1
11250000.0        1
780000.0          1
160000000.0       1
2130000.0         1
29000000.0        1
200000000.0       1
7800000.0         1
```

```
46000000.0        1
130000000.0       1
6800000.0         1
7600000.0         1
1050000.0         1
2150000.0         1
33000000.0        1
820000.0          1
6600000.0         1
5700000.0         1
8600000.0         1
8750000.0         1
555000.0          1
Name: count, dtype: int64
```

In [80]:
```
players_df['highest_market_value_in_eur'].isnull().sum()
```

Out[80]:
```
1251
```

In [81]:
```
players_df['highest_market_value_in_eur'].describe()
```

Out[81]:
```
count    2.887900e+04
mean     3.582088e+06
std      9.366787e+06
min      1.000000e+04
25%      2.500000e+05
50%      8.000000e+05
75%      2.750000e+06
max      2.000000e+08
Name: highest_market_value_in_eur, dtype: float64
```

In [82]:
```
# Calculating the mean market values based on position and age category
mean_highest_values = players_df.groupby(['position', 'age_group'])['highest_market

# Displaying the mean values
print(mean_highest_values)
```

```
      position      age_group  highest_market_value_in_eur
0       Attack   Ages 22 - 26                 4.216009e+06
1       Attack   Ages 27 - 32                 4.289119e+06
2       Attack   Ages 33 - 38                 4.728712e+06
3       Attack   Ages 39 - 44                 5.751636e+06
4       Attack   Ages 45 - 50                 8.016406e+06
5       Attack   Ages Over 50                          NaN
6       Attack  Ages Under 22                 2.400944e+06
7     Defender   Ages 22 - 26                 3.206923e+06
8     Defender   Ages 27 - 32                 3.318345e+06
9     Defender   Ages 33 - 38                 3.271292e+06
10    Defender   Ages 39 - 44                 3.810223e+06
11    Defender   Ages 45 - 50                 4.756200e+06
12    Defender  Ages Under 22                 1.502660e+06
13  Goalkeeper   Ages 22 - 26                 1.129499e+06
14  Goalkeeper   Ages 27 - 32                 2.282406e+06
15  Goalkeeper   Ages 33 - 38                 2.481094e+06
16  Goalkeeper   Ages 39 - 44                 2.836383e+06
17  Goalkeeper   Ages 45 - 50                 2.665569e+06
18  Goalkeeper   Ages Over 50                 2.066667e+06
19  Goalkeeper  Ages Under 22                 3.485526e+05
20    Midfield   Ages 22 - 26                 3.331310e+06
21    Midfield   Ages 27 - 32                 4.032131e+06
22    Midfield   Ages 33 - 38                 4.301075e+06
23    Midfield   Ages 39 - 44                 4.884429e+06
                                              6.341667e+06
                                              2.719915e+06
```

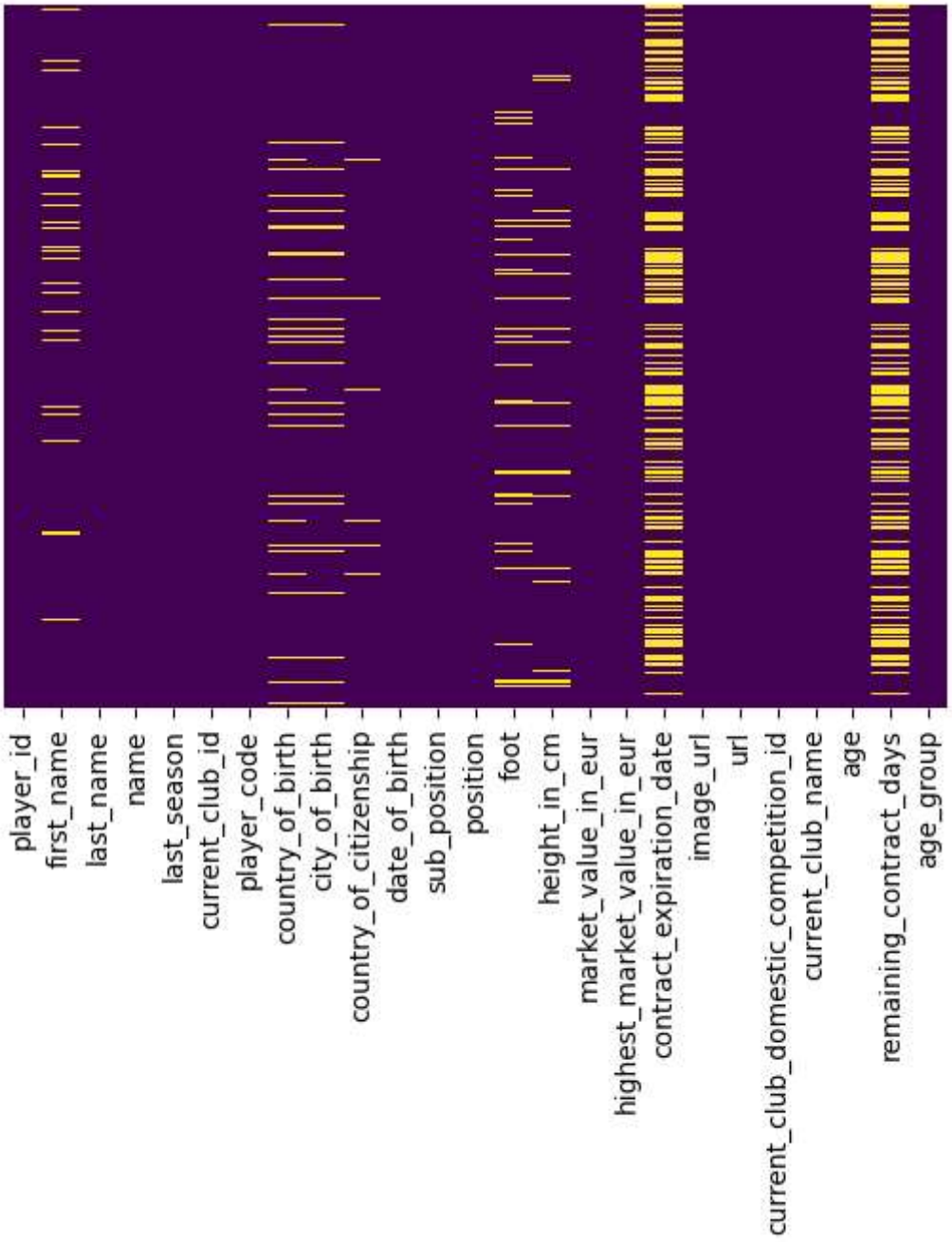
In [83]:
```
# Writing a function of conditional statements for imputing the mean market values
def impute_highest_market_value(cols):
    highest_market_value_in_eur = cols[0]
    position = cols[1]
    age_group = cols[2]

    if pd.isnull(highest_market_value_in_eur):

        if position == 'Attack':
            if age_group == 'Ages 22 - 26':
                return 4.223364e+06
            elif age_group == 'Ages 27 - 32':
                return 4.297336e+06
            elif age_group == 'Ages 33 - 38':
                return 4.713043e+06
            elif age_group == 'Ages 39 - 44':
                return 5.758007e+06
            elif age_group == 'Ages 45 - 50':
                return 8.016406e+06
            elif age_group == 'Ages Under 22':
                return 2.392311e+06
            else:
                return 3.582088e+06

        elif position == 'Defender':
            if age_group == 'Ages 22 - 26':
                return 3.183876e+06
            elif age_group == 'Ages 27 - 32':
                return 3.314468e+06
            elif age_group == 'Ages 33 - 38':
                return 3.277274e+06
            elif age_group == 'Ages 39 - 44':
                return 3.810223e+06
            elif age_group == 'Ages 45 - 50':
                return 4.756200e+06
            elif age_group == 'Ages Under 22':
                return 1.590434e+06
            else:
                return 3.582088e+06

        elif position == 'Goalkeeper':
            if age_group == 'Ages 22 - 26':
                return 1.129222e+06
            elif age_group == 'Ages 27 - 32':
                return 2.288247e+06
            elif age_group == 'Ages 33 - 38':
                return 2.477691e+06
            elif age_group == 'Ages 39 - 44':
                return 2.843714e+06
            elif age_group == 'Ages 45 - 50':
                return 2.665569e+06
            elif age_group == 'Ages Over 50':
                return 2.066667e+06
            elif age_group == 'Ages Under 22':
                return 3.469281e+05
            else:
                return 3.582088e+06

        elif position == 'Midfield':
            if age_group == 'Ages 22 - 26':
                return 3.344822e+06
            elif age_group == 'Ages 27 - 32':
```

```
            elif age_group == 'Ages 33 - 38':
                return 4.301016e+06
            elif age_group == 'Ages 39 - 44':
                return 4.890127e+06
            elif age_group == 'Ages 45 - 50':
                return 6.341667e+06
            elif age_group == 'Ages Under 22':
                return 2.716215e+06
            else:
                return 3.582088e+06

        else:
            return 3.582088e+06

    else:
        return highest_market_value_in_eur
```

In [84]:
```
# Applying the impute_highest_value_function to the players df
players_df['highest_market_value_in_eur'] = players_df[['highest_market_value_in_eu
```

In [85]:
```
sns.heatmap(players_df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[85]:
```
<Axes: >
```



In [86]:
```
players_df['highest_market_value_in_eur'].isnull().sum()
```

Out[86]:
```
0
```

In [87]:
```
#Extracting the year from datetime and storing it to a new fetaure column 'year'
#players_df['year']=players_df['datetime'].dt.year
```

Exploring the Older players within the data

In [88]:
```
older_players_df = players_df[(players_df['age_group'] == 'Ages 45 - 50') | (player
```

In [89]:
```
older_players_df.head(50)
```

Out[89]:

	player_id	first_name	last_name	name	last_season	current_club_id	player_code
0	598	Timo	Hildebrand	Timo Hildebrand	2014	24	timo-hildebrand
1	670	Martin	Petrov	Martin Petrov	2012	714	martin-petrov
4	3259	Damien	Duff	Damien Duff	2013	931	damien-duff
6	3804	Carlo	Nash	Carlo Nash	2013	1123	carlo-nash
8	4112	Rory	Delap	Rory Delap	2012	512	rory-delap
12	4487	Khalid	Sinouh	Khalid Sinouh	2012	467	khalid-sinouh
13	4740	Daniel	Van Buyten	Daniel Van Buyten	2013	27	daniel-van-buyten
16	4873	Armand	Deumi	Armand Deumi	2012	1506	armand-deumi
22	5925	Nicola	Legrottaglie	Nicola Legrottaglie	2013	1627	nicola-legrottaglie
25	6020	Francesco	Tavano	Francesco Tavano	2014	749	francesco-tavano
26	6065	Martin	Jiranek	Martin Jiranek	2013	3691	martin-jiranek
30	7072	Mustafa	Keceli	Mustafa Keceli	2012	3216	mustafa-keceli
31	7501	Juan Carlos	Valerón	Juan Carlos Valerón	2015	472	juan-carlos-valeron
32	7739	Patxi	Puñal	Patxi Puñal	2013	331	patxi-punal
47	13952	NaN	Liedson	Liedson	2012	720	liedson
49	14137	NaN	Briguel	Briguel	2015	1301	briguel
62	15817	Patrick	Deman	Patrick Deman	2014	601	patrick-deman
71	18324	Zoltán	Gera	Zoltán Gera	2013	984	zoltan-gera
75	19342	Ronald	Graafland	Ronald Graafland	2014	234	ronald-graafland
78	20913	Giuseppe	Biava	Giuseppe Biava	2014	800	giuseppe-biava
82	21495	Erik	Heijblok	Erik Heijblok	2013	1090	erik-heijblok
87	22489	Frank	Kristensen	Frank Kristensen	2013	865	frank-kristensen
				Chris Sörensen	2013	5724	chris-sorensen

	player_id	first_name	last_name	name	last_season	current_club_id	player_code
97	24975	Athanasios	Prittas	Athanasios Prittas	2012	7185	athanasios-prittas
102	26229	NaN	Diego	Diego	2016	1085	diego
143	36758	Paul	Mathers	Paul Mathers	2013	465	paul-mathers
166	42631	Oleksandr	Goryainov	Oleksandr Goryainov	2015	6414	oleksandr-goryainov
1492	2386	Marjan	Petkovic	Marjan Petkovic	2013	23	marjan-petkovic
1497	3395	Gábor	Babos	Gábor Babos	2014	132	gabor-babos
1500	4015	Louis	Saha	Louis Saha	2012	398	louis-saha
1506	4605	Vladan	Kujovic	Vladan Kujovic	2014	2282	vladan-kujovic
1509	5469	Sébastien	Chabbert	Sébastien Chabbert	2013	162	sebastien-chabbert
1511	5664	Arnaud	Le Lan	Arnaud Le Lan	2012	1158	arnaud-le-lan
1512	5881	Olivier	Renard	Olivier Renard	2013	172	olivier-renard
1513	5980	Luca	Toni	Luca Toni	2015	276	luca-toni
1515	6076	David	Di Michele	David Di Michele	2012	862	david-di-michele
1516	6106	Georgios	Karagounis	Georgios Karagounis	2013	931	georgios-karagounis
1522	7110	Ilker	Avcibay	Ilker Avcibay	2013	10484	ilker-avcibay
1523	7194	Murat	Erdogan	Murat Erdogan	2012	3216	murat-erdogan
1530	9696	Athanasios	Kostoulas	Athanasios Kostoulas	2012	6676	athanasios-kostoulas
1534	12638	Barry	Nicholson	Barry Nicholson	2013	2553	barry-nicholson
1540	13249	Andy	Marshall	Andy Marshall	2012	405	andy-marshall
1545	13560	Alan	Combe	Alan Combe	2012	43	alan-combe
1552	14716	Ross	Tokely	Ross Tokely	2012	2759	ross-tokely
1553	14937	Stipe	Pletikosa	Stipe Pletikosa	2015	897	stipe-pletikosa
				Maksym Kalynychenko	2013	2227	maksym-kalynychenko

	player_id	first_name	last_name	name	last_season	current_club_id	player_code
1561	16417	Ilias	Kotsios	Ilias Kotsios	2014	2672	ilias-kotsios
1575	19319	Danny	Koevermans	Danny Koevermans	2013	200	danny-koevermans
1596	24006	Volodymyr	Yezerskyi	Volodymyr Yezerskyi	2013	6996	volodymyr-yezerskyi
1622	29193	Rubén	Suárez	Rubén Suárez	2013	128	ruben-suarez

In [90]:
```
older_players_df_sorted_lastyear = older_players_df.sort_values("last_season", asce
```

In [91]:
```
older_players_df_sorted_lastyear.head(50)
```

Out[91]:

	player_id	first_name	last_name	name	last_season	current_club_id	pl
14598	5023	Gianluigi	Buffon	Gianluigi Buffon	2020	506	gianlu
18324	12682	Neil	Alexander	Neil Alexander	2020	1519	neil
18306	7668	Alberto	Cifuentes	Alberto Cifuentes	2020	2687	
23722	22149	Jean-François	Gillet	Jean-François Gillet	2020	3057	jean
27411	4811	NaN	Hilton	Hilton	2020	969	
21700	532	Claudio	Pizarro	Claudio Pizarro	2019	86	claud
14667	23149	Sören	Jochumsen	Sören Jochumsen	2019	2414	j
22105	12814	Julián	Speroni	Julián Speroni	2018	873	julia
1697	19299	Peter	van der Vlag	Peter van der Vlag	2018	1283	pete
18294	5928	Sergio	Pellissier	Sergio Pellissier	2018	862	sergio
23720	21891	Stefano	Sorrentino	Stefano Sorrentino	2018	862	
22151	24951	Hélder	Sousa	Hélder Sousa	2018	2425	he
23685	13957	NaN	Quim	Quim	2017	3336	
30215	7797	Albano	Bizzarri	Albano Bizzarri	2017	410	alban
24024	33927	Benjamin	Nivet	Benjamin Nivet	2017	1095	benja
18276	3441	Timmy	Simons	Timmy Simons	2017	2282	timm
18300	7215	Ahmet	Sahin	Ahmet Sahin	2017	1506	ah
12824	25236	Paulo	Lopes	Paulo Lopes	2017	294	pa
5257	5970	Dario	Dainelli	Dario Dainelli	2017	862	dar
7982	193257	Vladyslav	Gelzin	Vladyslav Gelzin	2017	23611	vlady
5941	19136	Marco	Storari	Marco Storari	2017	5	ma
27420	5986	Ferdinando	Coppola	Ferdinando Coppola	2017	276	fe
			ont	Bogdan Lobont	2017	12	bogd

	player_id	first_name	last_name	name	last_season	current_club_id	pl
25516	16137	Franco	Brienza	Franco Brienza	2016	1025	fran
5255	5958	Francesco	Totti	Francesco Totti	2016	12	fran
20220	24175	Pedro	Taborda	Pedro Taborda	2016	979	pedr
27441	9698	Stefanos	Kotsolis	Stefanos Kotsolis	2016	265	
22057	3146	Shay	Given	Shay Given	2016	512	s
16435	7028	Filip	Daems	Filip Daems	2016	968	f
3880	9646	Dmitriy	Loskov	Dmitriy Loskov	2016	932	dmi
27417	5880	Morgan	De Sanctis	Morgan De Sanctis	2016	162	m
5931	5278	Alexander	Manninger	Alexander Manninger	2016	31	
14658	21770	Gennaro	Sardo	Gennaro Sardo	2016	862	genr
14660	22285	Maurizio	Pugliesi	Maurizio Pugliesi	2016	749	
27392	488	Gerhard	Tremmel	Gerhard Tremmel	2016	2288	
12925	50552	Dimitrios	Konstantopoulos	Dimitrios Konstantopoulos	2016	641	konsta
24074	43411	Roberto	Colombo	Roberto Colombo	2016	1390	
25566	28828	Gabriele	Aldegani	Gabriele Aldegani	2016	2921	
102	26229	NaN	Diego	Diego	2016	1085	
25929	9775	Oleksandr	Shovkovskyi	Oleksandr Shovkovskyi	2016	338	
2163	89399	Ivan	Ershov	Ivan Ershov	2016	3729	iv
23700	16733	Lee	Baxter	Lee Baxter	2016	678	
25940	20865	Lars	Winde	Lars Winde	2016	3426	
10896	13583	Clint	Hill	Clint Hill	2016	124	
3882	12804	Barry	Robson	Barry Robson	2015	370	ba

	player_id	first_name	last_name	name	last_season	current_club_id	pl
18350	18542	François	Modesto	François Modesto	2015	595	
12720	10	Miroslav	Klose	Miroslav Klose	2015	398	mirc
8931	7492	NaN	Manuel Pablo	Manuel Pablo	2015	897	mar
14593	4267	Tim	Howard	Tim Howard	2015	29	ti
7100	3357	Mark	Schwarzer	Mark Schwarzer	2015	1003	mark-

In [92]:
```
older_players_df_sorted_lastyear.shape
```

Out[92]:
```
(419, 25)
```

In [93]:
```
last_season_2021_players_df = players_df[players_df['last_season'] <= 2021]
```

In [94]:
```
last_season_2021_players_df.shape
```

Out[94]:
```
(19961, 25)
```

In [95]:
```
last_season_2021_players_df.head()
```

Out[95]:

	player_id	first_name	last_name	name	last_season	current_club_id	player_code	country
0	598	Timo	Hildebrand	Timo Hildebrand	2014	24	timo-hildebrand	
1	670	Martin	Petrov	Martin Petrov	2012	714	martin-petrov	
2	1323	Martin	Amedick	Martin Amedick	2012	24	martin-amedick	
3	3195	Jermaine	Pennant	Jermaine Pennant	2013	512	jermaine-pennant	
4	3259	Damien	Duff	Damien Duff	2013	931	damien-duff	

In [96]:
```
players_df.shape
```

Out[96]:
```
(30130, 25)
```

After viewing the data above it is apparent that the datasets still contain data of retired players whos market values and ages are innacurate based on their current age, status and the current market. Retaining these entries will introduce bias and innaccurate model player valuations results. A new dataframe of current players, i.e excluding those who retired in or before 2022 is created.

In [97]:
```
current_players_df = players_df[players_df['last_season'] >= 2022]
```

In [98]:
```
current_players_df.shape
```

Out[98]:
```
(10169, 25)
```

In [99]:
```
appearances_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1485697 entries, 0 to 1485696
Data columns (total 13 columns):
 #   Column                 Non-Null Count    Dtype 
---  ------                 --------------    ----- 
 0   appearance_id          1485697 non-null  object
 1   game_id                1485697 non-null  int64 
 2   player_id              1485697 non-null  int64 
 3   player_club_id         1485697 non-null  int64 
 4   player_current_club_id 1485697 non-null  int64 
 5   date                   1485697 non-null  object
 6   player_name            1485373 non-null  object
 7   competition_id         1485697 non-null  object
 8   yellow_cards           1485697 non-null  int64 
 9   red_cards              1485697 non-null  int64 
 10  goals                  1485697 non-null  int64 
 11  assists                1485697 non-null  int64 
 12  minutes_played         1485697 non-null  int64 
dtypes: int64(9), object(4)
memory usage: 147.4+ MB
```

In [100…
```
# Creating a Feature for 'year' from the 'date' column
appearances_df['datetime']=pd.to_datetime(appearances_df['date'], format="%Y-%m-%d"
appearances_df['year']=appearances_df['datetime'].dt.year
```

In [101…
```
appearances_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1485697 entries, 0 to 1485696
Data columns (total 15 columns):
 #   Column                 Non-Null Count    Dtype         
---  ------                 --------------    -----         
 0   appearance_id          1485697 non-null  object        
 1   game_id                1485697 non-null  int64         
 2   player_id              1485697 non-null  int64         
 3   player_club_id         1485697 non-null  int64         
 4   player_current_club_id 1485697 non-null  int64         
 5   date                   1485697 non-null  object        
 6   player_name            1485373 non-null  object        
 7   competition_id         1485697 non-null  object        
 8   yellow_cards           1485697 non-null  int64         
 9   red_cards              1485697 non-null  int64         
 10  goals                  1485697 non-null  int64         
 11  assists                1485697 non-null  int64         
 12  minutes_played         1485697 non-null  int64         
 13  datetime               1485697 non-null  datetime64[ns]
 14  year                   1485697 non-null  int32         
dtypes: datetime64[ns](1), int32(1), int64(9), object(4)
memory usage: 164.4+ MB
```

In [102…
```
accumulated_stats = appearances_df.groupby(['player_id', 'year']).agg({
    'yellow_cards': 'sum',
    'red_cards': 'sum',
    'goals': 'sum',
    'assists': 'sum',
    'minutes_played': 'sum'
}).reset_index()
```

In [103…
```
print(accumulated_stats)
```

```
IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.

Current values:
NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
NotebookApp.rate_limit_window=3.0 (secs)
```

In [104…
```
accumulated_stats.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 91532 entries, 0 to 91531
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   player_id       91532 non-null  int64
 1   year            91532 non-null  int32
 2   yellow_cards    91532 non-null  int64
 3   red_cards       91532 non-null  int64
 4   goals           91532 non-null  int64
 5   assists         91532 non-null  int64
 6   minutes_played  91532 non-null  int64
dtypes: int32(1), int64(6)
memory usage: 4.5 MB
```

In [105…
```
accumulated_stats.nunique()
```

Out[105]:
```
player_id         23740
year                 12
yellow_cards         24
red_cards             4
goals                53
assists              32
minutes_played     4175
dtype: int64
```

In [106…
```
current_players_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 10169 entries, 72 to 30301
Data columns (total 25 columns):
 #   Column                                 Non-Null Count  Dtype
---  ------                                 --------------  -----
 0   player_id                              10169 non-null  int64
 1   first_name                             9568 non-null   object
 2   last_name                              10169 non-null  object
 3   name                                   10169 non-null  object
 4   last_season                            10169 non-null  int64
 5   current_club_id                        10169 non-null  int64
 6   player_code                            10169 non-null  object
 7   country_of_birth                       8843 non-null   object
 8   city_of_birth                          9319 non-null   object
 9   country_of_citizenship                 9627 non-null   object
 10  date_of_birth                          10169 non-null  datetime64[ns]
 11  sub_position                           10169 non-null  object
 12  position                               10169 non-null  object
 13  foot                                   9574 non-null   object
 14  height_in_cm                           9518 non-null   float64
 15  market_value_in_eur                    10169 non-null  float64
 16  highest_market_value_in_eur            10169 non-null  float64
 17  contract_expiration_date               8945 non-null   datetime64[ns]
 18  image_url                              10169 non-null  object
 19  url                                    10169 non-null  object
 20  current_club_domestic_competition_id   10169 non-null  object
 21  current_club_name                      10169 non-null  object
 22  age                                    10169 non-null  int32
 23  remaining_contract_days                8945 non-null   float64
 24  age_group                              10169 non-null  object
dtypes: datetime64[ns](2), float64(4), int32(1), int64(3), object(15)
memory usage: 2.0+ MB
```

In [107…

```
current_players_df.nunique()
```

Out[107]:

```
player_id                              10169
first_name                              3421
last_name                               8445
name                                   10072
last_season                                2
current_club_id                          274
player_code                            10067
country_of_birth                         152
city_of_birth                           4000
country_of_citizenship                   144
date_of_birth                           5035
sub_position                              13
position                                   4
foot                                       3
height_in_cm                              46
market_value_in_eur                      134
highest_market_value_in_eur              159
contract_expiration_date                  48
image_url                               9066
url                                    10169
current_club_domestic_competition_id      14
current_club_name                        274
age                                       28
remaining_contract_days                   48
age_group                                  5
dtype: int64
```

In [108…

```
current_players_df_with_stats = current_players_df.merge(accumulated_stats, left_or
```

In [109…
```
#current_players_df_with_stats = current_players_df.merge(accumulated_stats, on='pl
```

In [110…
```
current_players_df_with_stats.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42736 entries, 0 to 42735
Data columns (total 31 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   player_id                             42736 non-null  int64
 1   first_name                            40076 non-null  object
 2   last_name                             42736 non-null  object
 3   name                                  42736 non-null  object
 4   last_season                           42736 non-null  int64
 5   current_club_id                       42736 non-null  int64
 6   player_code                           42736 non-null  object
 7   country_of_birth                      39927 non-null  object
 8   city_of_birth                         41770 non-null  object
 9   country_of_citizenship                40847 non-null  object
 10  date_of_birth                         42736 non-null  datetime64[ns]
 11  sub_position                          42736 non-null  object
 12  position                              42736 non-null  object
 13  foot                                  42196 non-null  object
 14  height_in_cm                          42260 non-null  float64
 15  market_value_in_eur                   42736 non-null  float64
 16  highest_market_value_in_eur           42736 non-null  float64
 17  contract_expiration_date              38142 non-null  datetime64[ns]
 18  image_url                             42736 non-null  object
 19  url                                   42736 non-null  object
 20  current_club_domestic_competition_id  42736 non-null  object
 21  current_club_name                     42736 non-null  object
 22  age                                   42736 non-null  int32
 23  remaining_contract_days               38142 non-null  float64
 24  age_group                             42736 non-null  object
 25  year                                  42736 non-null  int32
 26  yellow_cards                          42736 non-null  int64
 27  red_cards                             42736 non-null  int64
 28  goals                                 42736 non-null  int64
 29  assists                               42736 non-null  int64
 30  minutes_played                        42736 non-null  int64
dtypes: datetime64[ns](2), float64(4), int32(2), int64(8), object(15)
memory usage: 9.8+ MB
```

In [111…
```
current_players_df_with_stats.nunique()
```

Out[111]:

```
player_id                          8879
first_name                         3119
last_name                          7477
name                               8802
last_season                           2
current_club_id                     274
player_code                        8797
country_of_birth                    147
city_of_birth                      3759
country_of_citizenship              140
date_of_birth                      4711
sub_position                         13
position                              4
foot                                  3
height_in_cm                         45
market_value_in_eur                 132
highest_market_value_in_eur         153
contract_expiration_date             47
image_url                          8249
url                                8879
current_club_domestic_competition_id   14
current_club_name                   274
age                                  28
remaining_contract_days              47
age_group                             5
year                                 12
yellow_cards                         23
red_cards                             4
goals                                53
assists                              32
minutes_played                     4071
dtype: int64
```

Merging Required Datasets

In [112…

```
# Creating a copy of the players_df
merged_players_df = players_df.copy()

# Merging the games and appearances df's
games_and_appearances_df = appearances_df.merge(games_df, on=['game_id'], how='left

# Selecting the last five seasons for gathering stats
seasons = [2019, 2020, 2021, 2022, 2023]

# Creating and initialising a total column for each statistics
for stat in ['games', 'goals', 'assists', 'minutes_played', 'goals_for', 'goals_aga
    merged_players_df[stat + '_total'] = 0

# Creating a function to collate player stats
def player_stats(player_id, season, df):
    df = df[df['player_id'] == player_id]
    df = df[df['season'] == season]
    if (df.shape[0] == 0):
        Out = [(np.nan, season,0,0,0,0,0,0,0,0,0)]
        out_df = pd.DataFrame(data = Out, columns = ['player_id','season','goals','
                                                     'assists','minutes_played','go
        return out_df
    else:
        df["goals_for"] = df.apply(lambda row: row['home_club_goals'] if row['home_
            else row['away_club_goals'] if row['away_club_id'] == row['player_club_
            else np.nan, axis=1)
        ly(lambda row: row['away_club_goals'] if row['h
```

```
            else row['home_club_goals'] if row['away_club_id'] == row['player_club_
            else np.nan, axis=1)
        df['clean_sheet'] = df.apply(lambda row: 1 if row['goals_against'] == 0
            else 0 if row['goals_against'] > 0
            else np.nan, axis=1)
        df = df.groupby(['player_id',"season"],as_index=False).agg({'goals': 'sum',
                                                                    'assists': 's
                                                                    'goals_agains
        out_df = df.rename(columns={'game_id': 'games'})
        return out_df

# Iterating through players
for index in merged_players_df.index:
    id = merged_players_df.loc[index][0]
    name = merged_players_df.loc[index][1]

    # Iterating through seasons
    for season in seasons:
        stats = player_stats(id, season, games_and_appearances_df)
        try:
            for stat in ['games', 'goals', 'assists', 'minutes_played', 'goals_for'
                merged_players_df.at[index, stat + '_{}'.format(season)] = stats[st
                merged_players_df.at[index, stat + '_total'] += stats[stat][0]
        except:
            pass

print('Player and statistics data merged.')
print(merged_players_df.info())
```

```
Player and statistics data merged.
<class 'pandas.core.frame.DataFrame'>
Index: 30130 entries, 0 to 30301
Data columns (total 79 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   player_id                             30130 non-null  int64
 1   first_name                            28173 non-null  object
 2   last_name                             30130 non-null  object
 3   name                                  30130 non-null  object
 4   last_season                           30130 non-null  int64
 5   current_club_id                       30130 non-null  int64
 6   player_code                           30130 non-null  object
 7   country_of_birth                      27528 non-null  object
 8   city_of_birth                         28013 non-null  object
 9   country_of_citizenship                29587 non-null  object
 10  date_of_birth                         30130 non-null  datetime64[ns]
 11  sub_position                          30130 non-null  object
 12  position                              30130 non-null  object
 13  foot                                  27844 non-null  object
 14  height_in_cm                          28099 non-null  float64
 15  market_value_in_eur                   30130 non-null  float64
 16  highest_market_value_in_eur           30130 non-null  float64
 17  contract_expiration_date              18799 non-null  datetime64[ns]
 18  image_url                             30130 non-null  object
 19  url                                   30130 non-null  object
 20  current_club_domestic_competition_id  30130 non-null  object
 21  current_club_name                     30130 non-null  object
 22  age                                   30130 non-null  int32
 23  remaining_contract_days               18799 non-null  float64
 24  age_group                             30130 non-null  object
 25  games_total                           30130 non-null  int64
 26  goals_total                           30130 non-null  int64
 27  assists_total                         30130 non-null  int64
 28  minutes_played_total                  30130 non-null  int64
 29  goals_for_total                       30130 non-null  int64
 30  goals_against_total                   30130 non-null  int64
 31  clean_sheet_total                     30130 non-null  int64
 32  yellow_cards_total                    30130 non-null  int64
 33  red_cards_total                       30130 non-null  int64
 34  games_2019                            30130 non-null  float64
 35  goals_2019                            30130 non-null  float64
 36  assists_2019                          30130 non-null  float64
 37  minutes_played_2019                   30130 non-null  float64
 38  goals_for_2019                        30130 non-null  float64
 39  goals_against_2019                    30130 non-null  float64
 40  clean_sheet_2019                      30130 non-null  float64
 41  yellow_cards_2019                     30130 non-null  float64
 42  red_cards_2019                        30130 non-null  float64
 43  games_2020                            30130 non-null  float64
 44  goals_2020                            30130 non-null  float64
 45  assists_2020                          30130 non-null  float64
 46  minutes_played_2020                   30130 non-null  float64
 47  goals_for_2020                        30130 non-null  float64
 48  goals_against_2020                    30130 non-null  float64
 49  clean_sheet_2020                      30130 non-null  float64
 50  yellow_cards_2020                     30130 non-null  float64
 51  red_cards_2020                        30130 non-null  float64
 52  games_2021                            30130 non-null  float64
 53  goals_2021                            30130 non-null  float64
 54  assists_2021                          30130 non-null  float64
 55  minutes_played_2021                   30130 non-null  float64
 56  [illegible]                           30130 non-null  float64
 57  goals_against_2021                    30130 non-null  float64
```

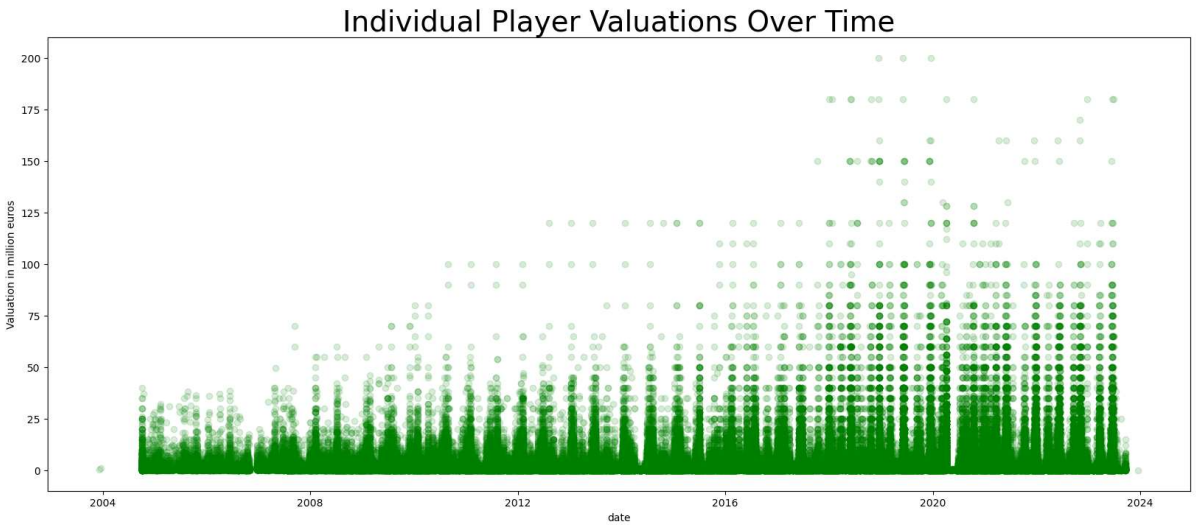
```
 58  clean_sheet_2021                30130 non-null  float64
 59  yellow_cards_2021               30130 non-null  float64
 60  red_cards_2021                  30130 non-null  float64
 61  games_2022                      30130 non-null  float64
 62  goals_2022                      30130 non-null  float64
 63  assists_2022                    30130 non-null  float64
 64  minutes_played_2022             30130 non-null  float64
 65  goals_for_2022                  30130 non-null  float64
 66  goals_against_2022              30130 non-null  float64
 67  clean_sheet_2022                30130 non-null  float64
 68  yellow_cards_2022               30130 non-null  float64
 69  red_cards_2022                  30130 non-null  float64
 70  games_2023                      30130 non-null  float64
 71  goals_2023                      30130 non-null  float64
 72  assists_2023                    30130 non-null  float64
 73  minutes_played_2023             30130 non-null  float64
 74  goals_for_2023                  30130 non-null  float64
 75  goals_against_2023              30130 non-null  float64
 76  clean_sheet_2023                30130 non-null  float64
 77  yellow_cards_2023               30130 non-null  float64
 78  red_cards_2023                  30130 non-null  float64
dtypes: datetime64[ns](2), float64(49), int32(1), int64(12), object(15)
memory usage: 19.3+ MB
None
```

EDA with Visualisations

Evaluating changes in Market Value over time

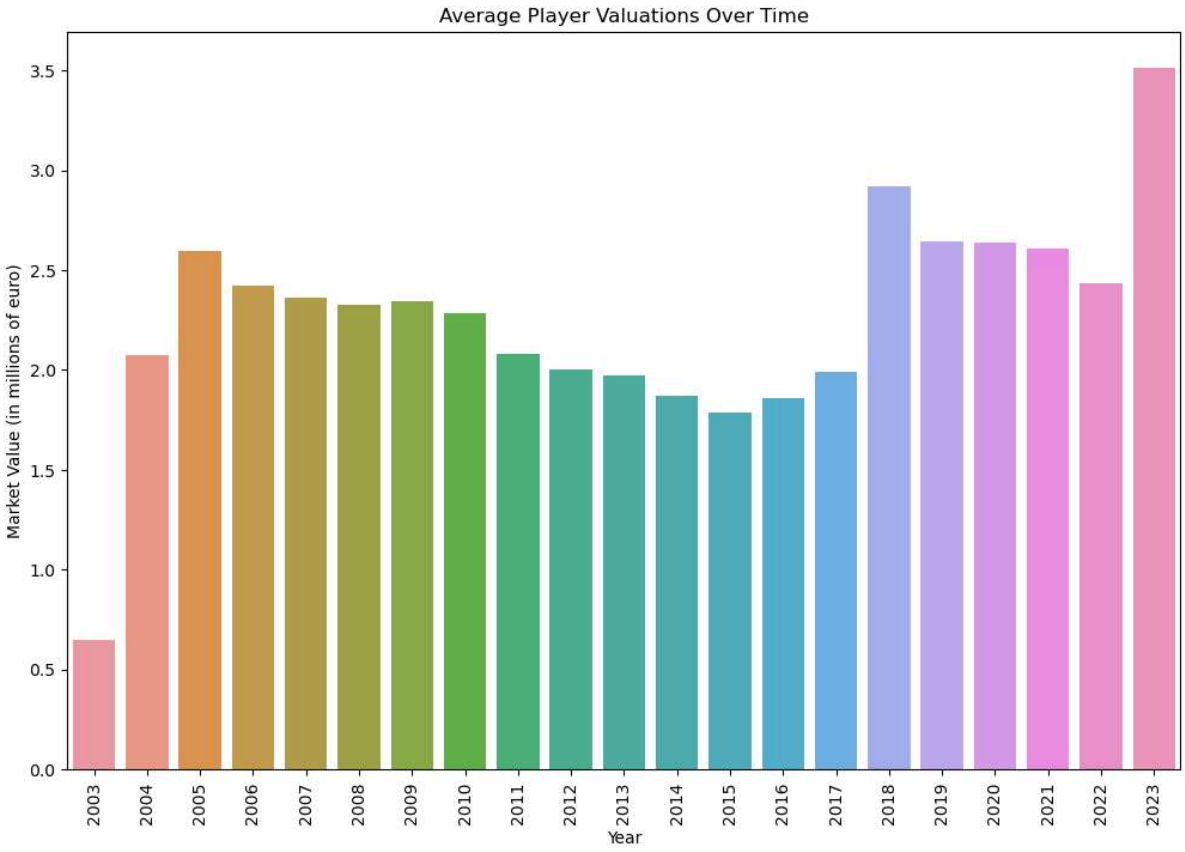
In [113…

```
# Visualising  player valuation data over time
plt.figure(figsize=(20,8))
plt.scatter(player_valuations_df['datetime'],y=player_valuations_df['market_value_i
plt.xlabel('date');plt.ylabel('Valuation in million euros')
plt.title('Individual Player Valuations Over Time',fontsize=28)
plt.show()
```



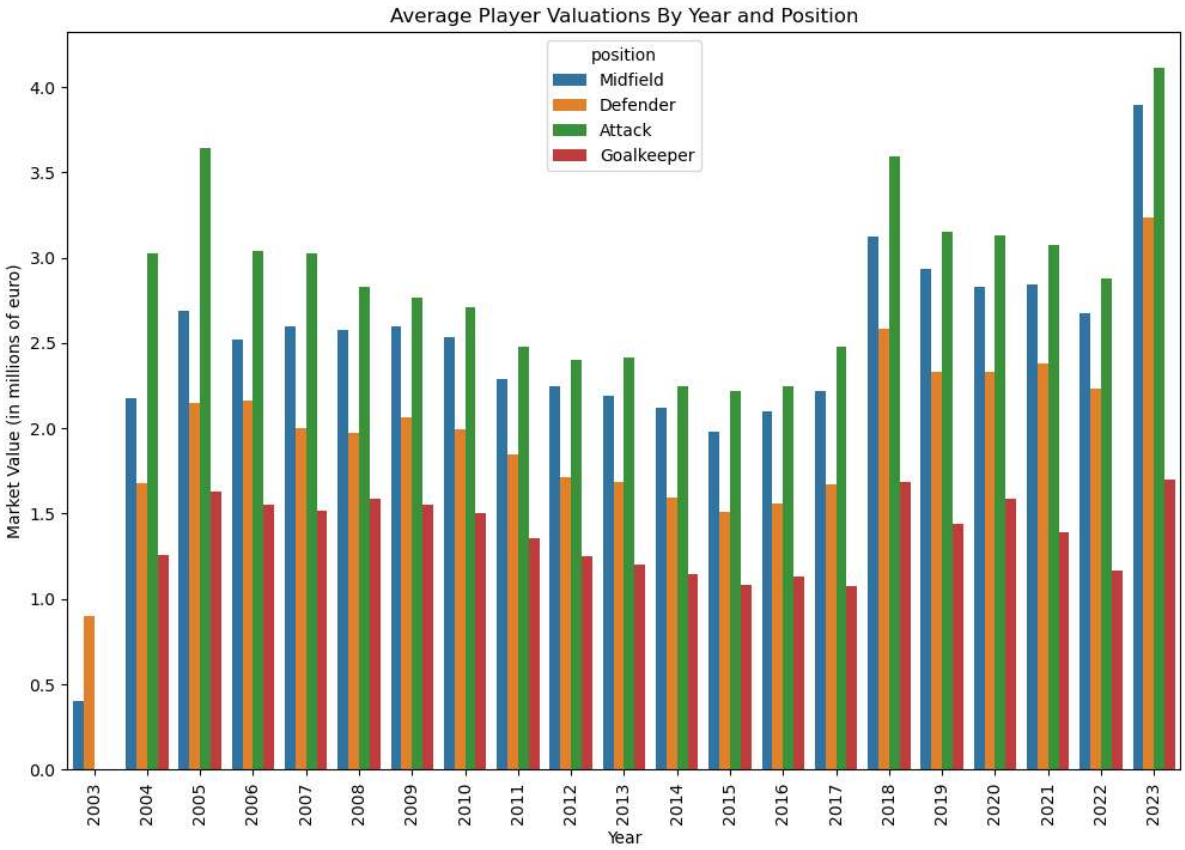
In [114…

```
plt.figure(figsize=(12, 8))
sns.barplot(x = 'year', y=player_valuations_df['market_value_in_eur']/1000000, data
plt.title("Average Player Valuations Over Time")
plt.ylabel('Market Value (in millions of euro)')
plt.xlabel('Year')
plt.xticks(rotation = 'vertical')
```

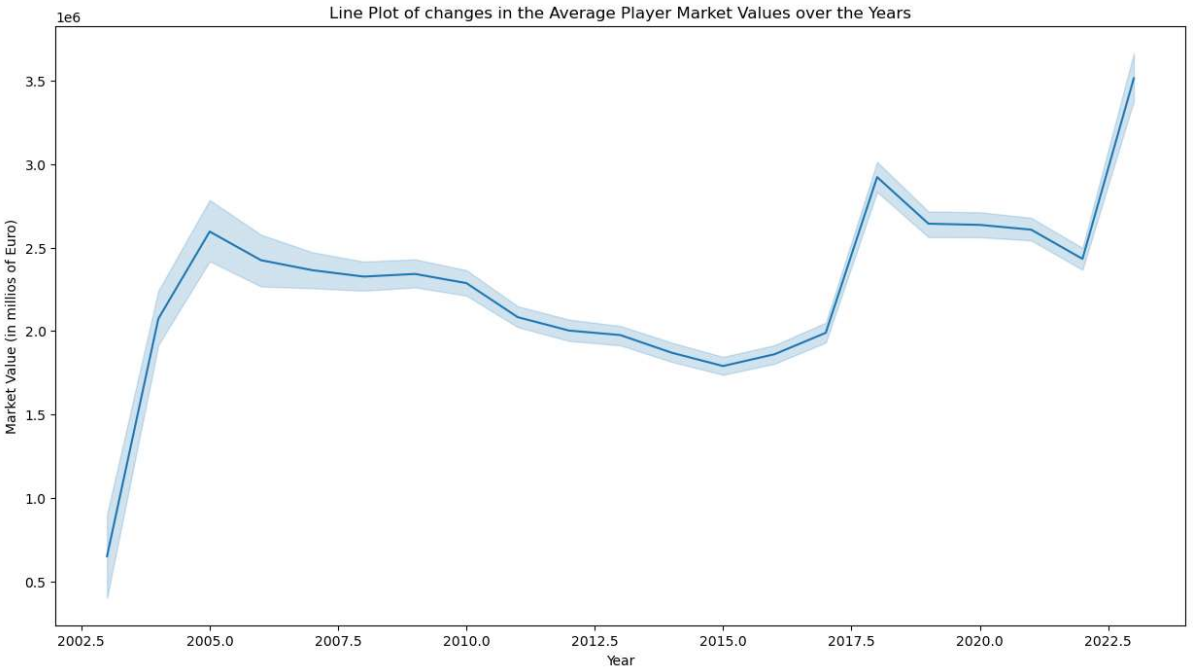


In [115…

```
plt.figure(figsize=(12, 8))
sns.barplot(x = 'year', y=player_valuations_df['market_value_in_eur']/1000000, hue=
plt.title("Average Player Valuations By Year and Position")
plt.ylabel('Market Value (in millions of euro)')
plt.xlabel('Year')
plt.xticks(rotation = 'vertical')
plt.show()
```



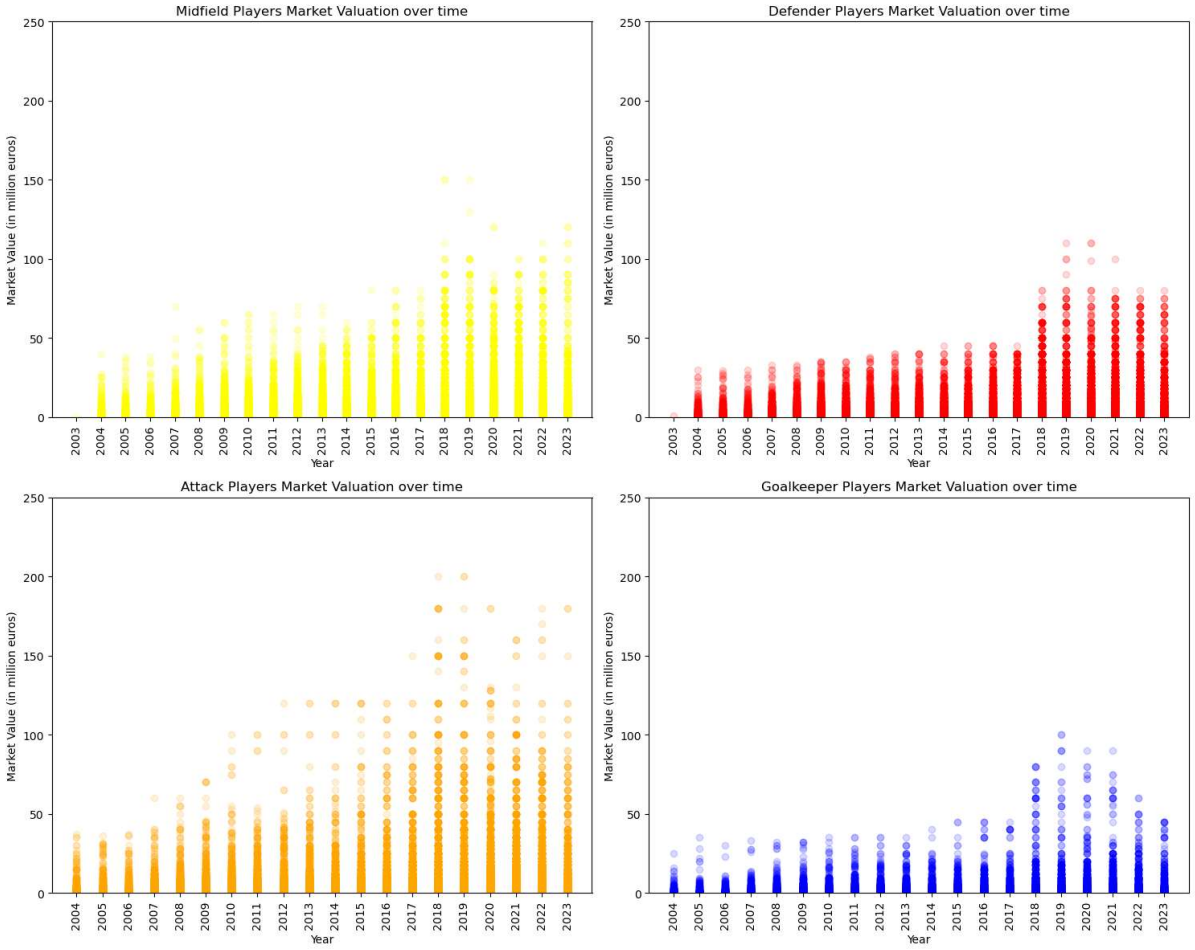
In [116…
```
# Lineplot of market value by year
plt.figure(figsize=(15, 8))
sns.lineplot(x='year', y='market_value_in_eur', data=player_valuations_df)
plt.title('Line Plot of changes in the Average Player Market Values over the Years'
plt.xlabel('Year')
plt.ylabel('Market Value (in millios of Euro)')
plt.show()
```



In [120…
```
filtered_position_df['position'].value_counts()
```

Out[120]:
```
position
Defender      142596
Midfield      128467
Attack        121930
Goalkeeper     46775
Name: count, dtype: int64
```

In [123…
```
#Storing the unique postions
positions = filtered_position_df.position.unique()[:4]
num_positions = len(positions)
# Storing colours for graphs
colour = ['yellow', 'r', 'orange', 'b', 'g']
# Calculating the number of rows and columns for subplots
num_rows = (num_positions // 2) + (num_positions % 2)
num_cols = 2

plt.figure(figsize=(15, 12))

for n, position in enumerate(positions, start=1):
    plt.subplot(num_rows, num_cols, n)
    filtered_position_df1 = filtered_position_df[(filtered_position_df.position ==
    plt.scatter(filtered_position_df1['year'], y=filtered_position_df1['market_valu
    plt.xlabel('Year')
    plt.ylabel('Market Value (in million euros)')
    plt.title(f'{position} Players Market Valuation over time')
    plt.ylim(0, 250)
    plt.xticks(rotation=90, ticks=filtered_position_df1['year'].unique())

plt.tight_layout()
plt.show()
```

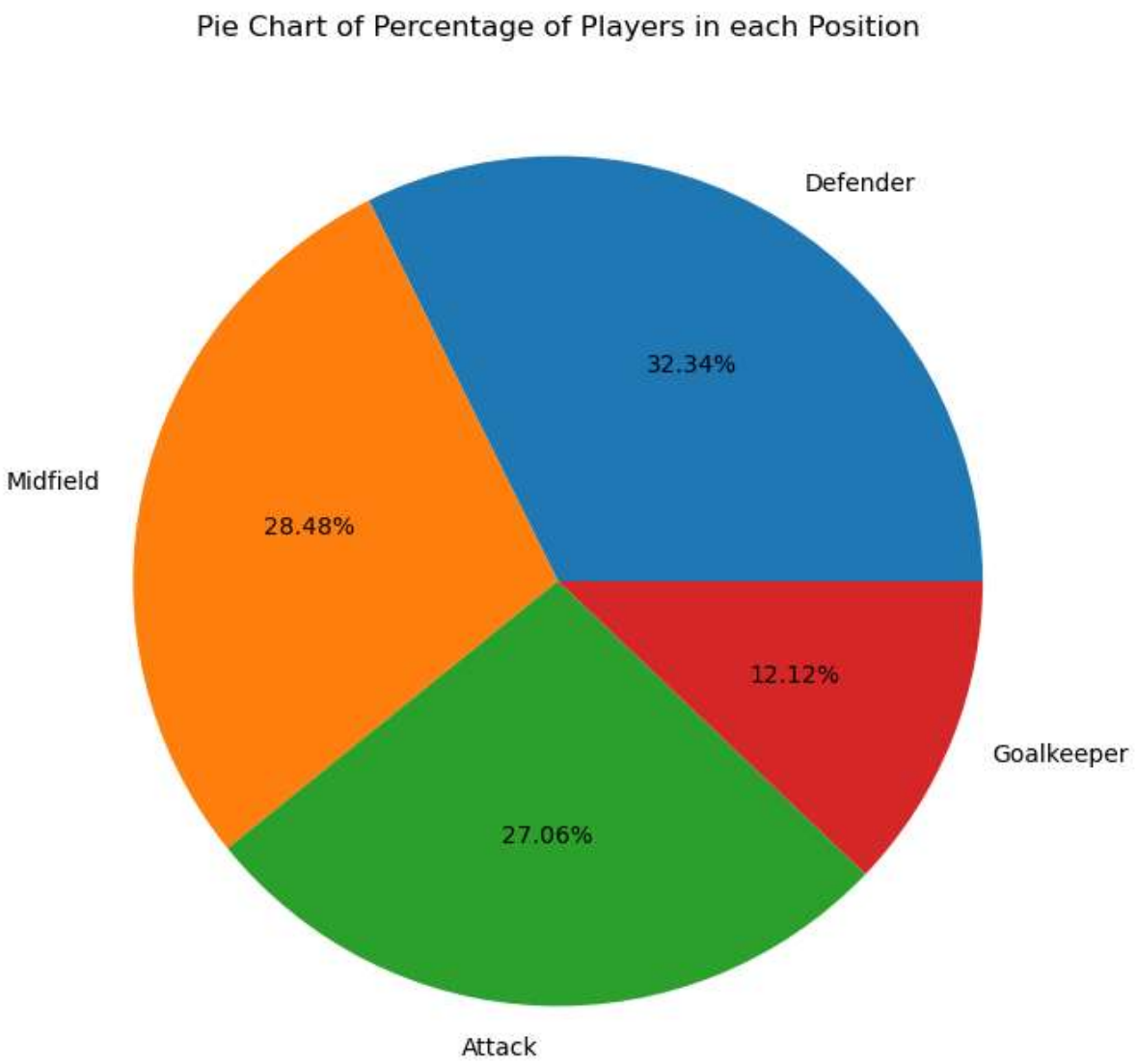


Analysis on the Current Player Market

Market Density

In [130…

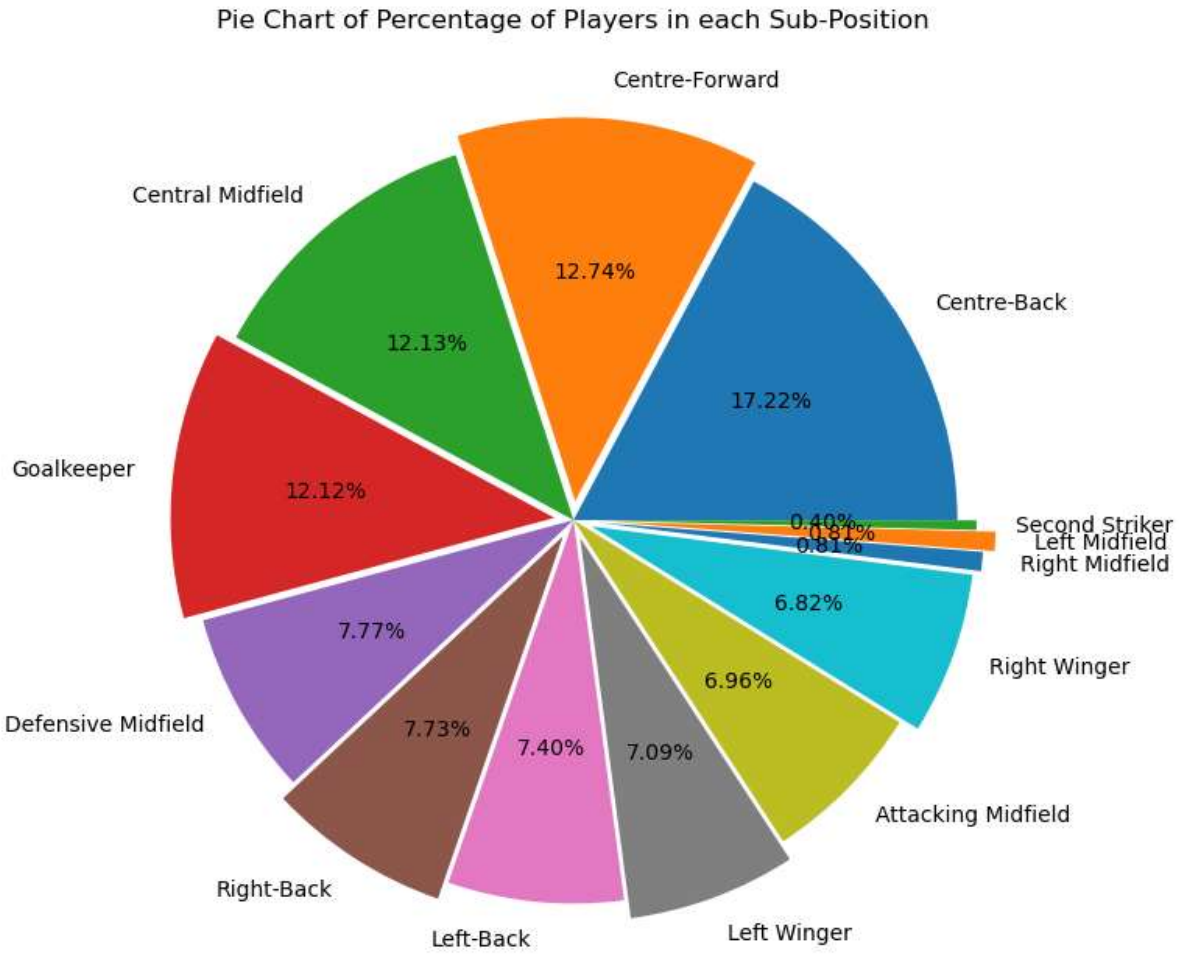
```
current_players_df['position'].value_counts().plot(kind = 'pie', autopct = '%.2f%%'
plt.ylabel('')
plt.title('Pie Chart of Percentage of Players in each Position')
plt.show()
```



We can see that the market of currenlty playing players is evenly distributed between Attack, Midfield and Defence. There are fewer players in the goalkeeper position which is to be expected as only one goalkeeper is ever selected in a starting eleven.

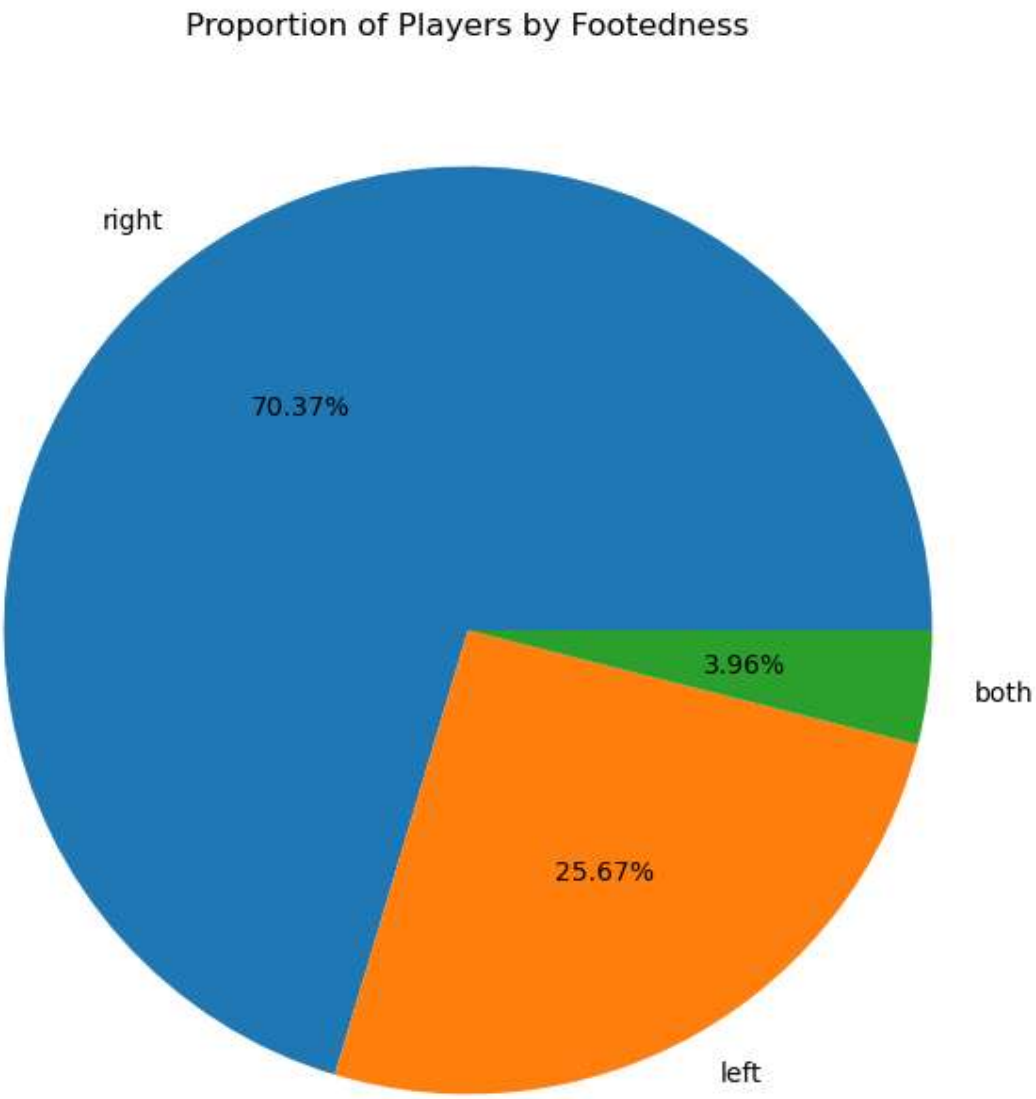
In [135…

```
explode = [0, 0.05, 0, 0.05, 0, 0.05, 0, 0.05, 0, 0.05, 0.07, 0.10, 0.05]
current_players_df['sub_position'].value_counts().plot(kind = 'pie', autopct = '%.2
plt.ylabel('')
plt.title('Pie Chart of Percentage of Players in each Sub-Position')
plt.show()
```



In [136…

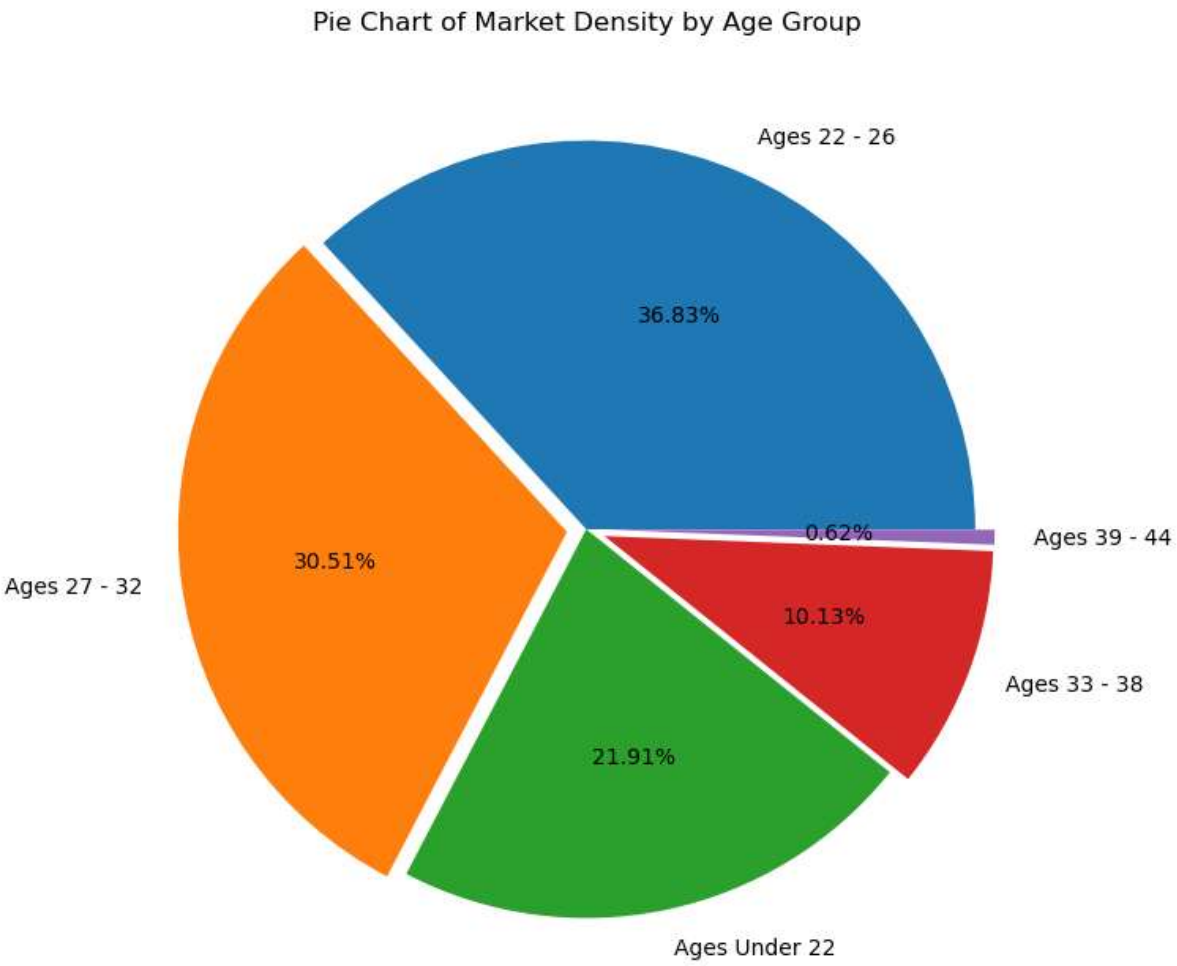
```
current_players_df['foot'].value_counts().plot(kind = 'pie', autopct = '%.2f%%', fi
plt.ylabel('')
plt.title('Proportion of Players by Footedness')
plt.show()
```



We can see from the pie chart above that the large majority of players are right footed. Only 25.67% are left footed which along with the both footed players may make them higher valued in the market due to scarcity.

In [137…

```
explode = [0, 0.05, 0, 0.05, 0.05]
current_players_df['age_group'].value_counts().plot(kind = 'pie', autopct = '%.2f%%
plt.ylabel('')
plt.title('Pie Chart of Market Density by Age Group')
plt.show()
```



As we would expect, approximately 67% of the player market is of players aged between 22 and 32 which is generally considered a players prime years. 21.91% consists of younger up and coming players with roughly 10% in the twilight of their professional careers.

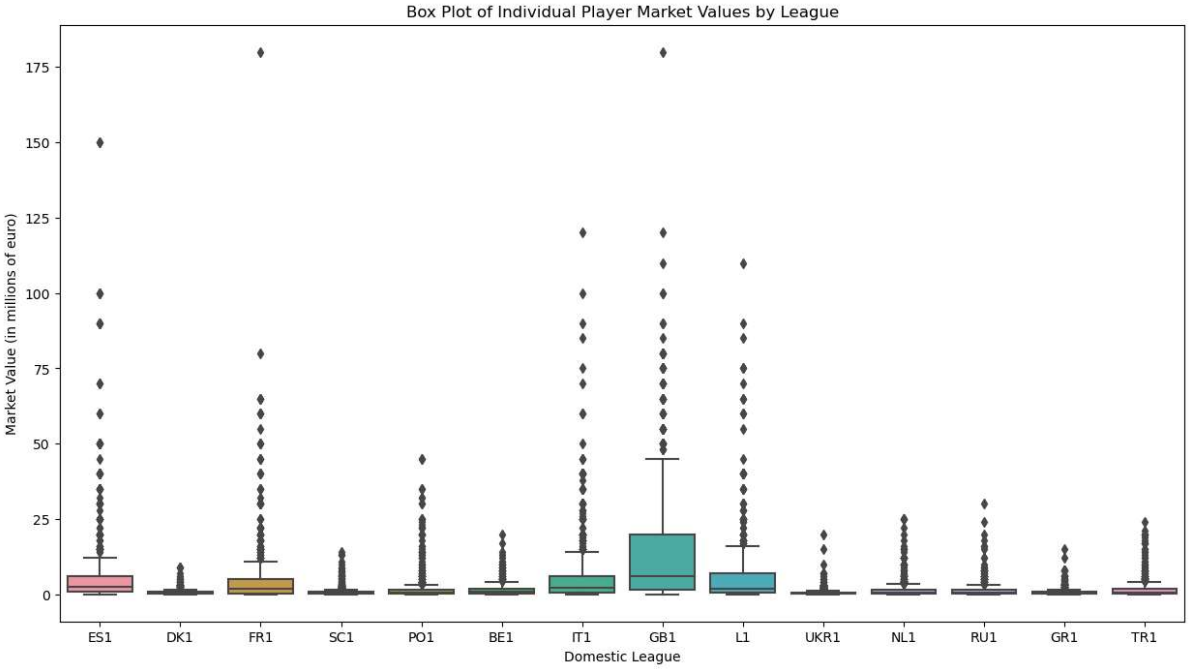
Analysis on the most recent Market Valuations

Player Market Values by Domestic Competition

Individual Player Values

In [138…

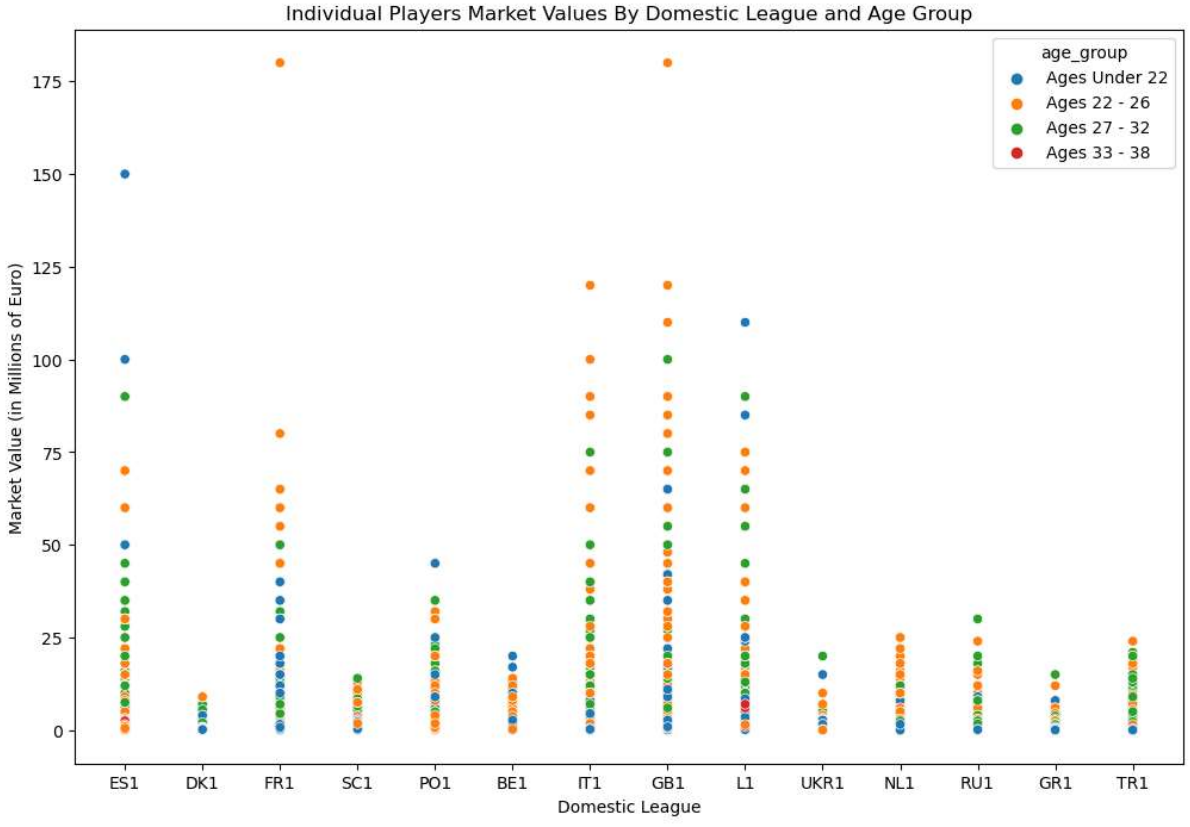
```
# Boxplot of market value by league
plt.figure(figsize=(15, 8))
sns.boxplot(x='current_club_domestic_competition_id', y=current_players_df['market_
plt.xlabel('Domestic League')
plt.ylabel('Market Value (in millions of euro)')
plt.title('Box Plot of Individual Player Market Values by League')
plt.show()
```



The highest valued players curently play in LaLiga (Spain - ES1), Ligue 1 (France - FR1), Serie A (Italy - IT1), Premier League (England - GB1) and Bundesliga (Germany - L1). All five leagues also contain players whose market values far exceed the average within those leagues.

In [141…

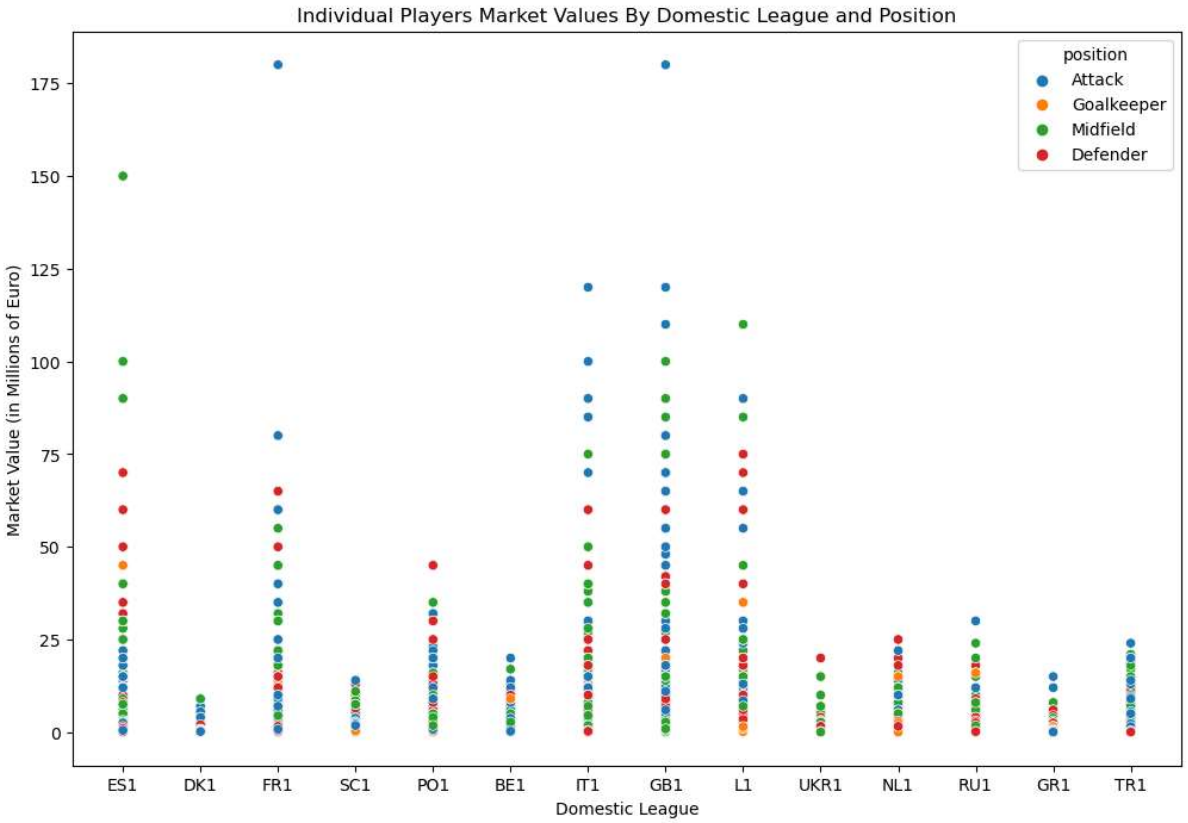
```
# Seaborn Scatterplot of Market Value by League and Age
plt.figure(figsize=(12, 8))
sns.scatterplot(x='current_club_domestic_competition_id', y=current_players_df['mar
plt.title('Individual Players Market Values By Domestic League and Age Group')
plt.xlabel('Domestic League')
plt.ylabel('Market Value (in Millions of Euro)')
plt.show()
```



We see that the highest valued players are generally under 27 years of age.

In [144…

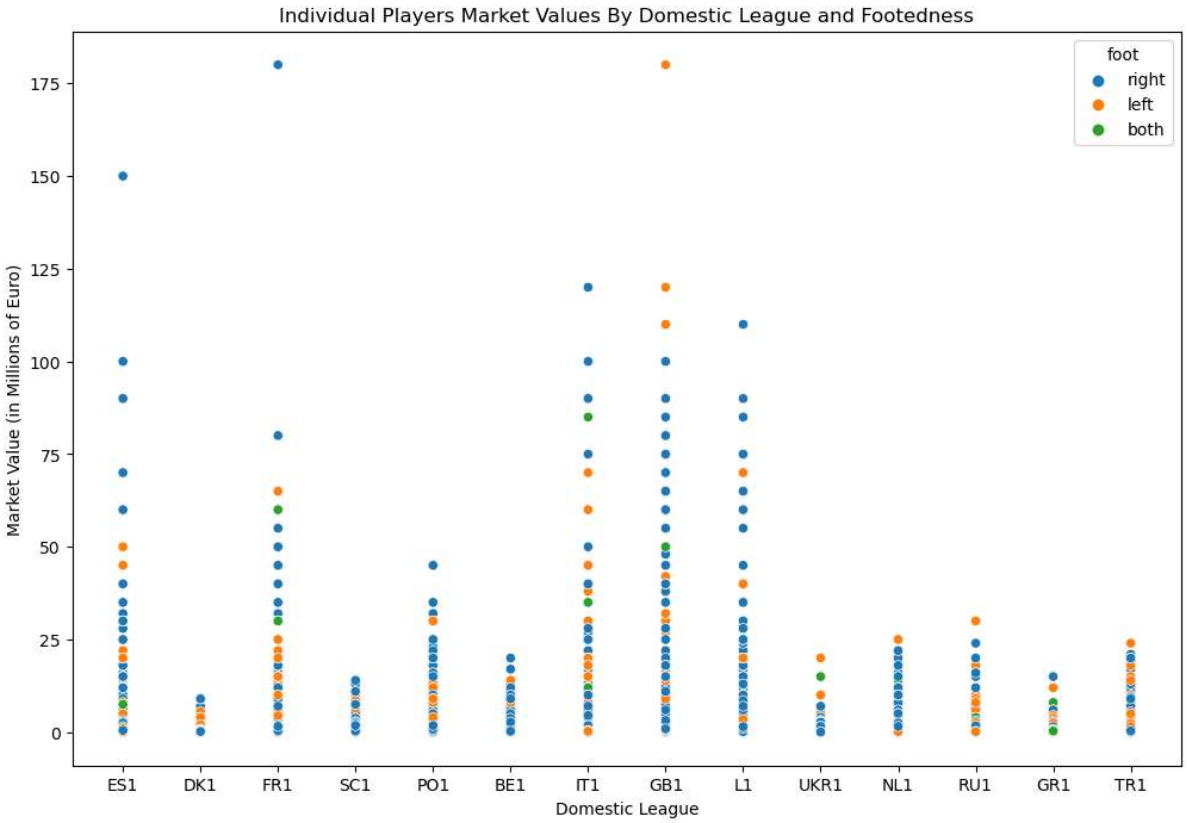
```
# Seaborn Scatterplot of Market Value by League and Position
plt.figure(figsize=(12, 8))
sns.scatterplot(x='current_club_domestic_competition_id', y=current_players_df['mar
plt.title('Individual Players Market Values By Domestic League and Position')
plt.xlabel('Domestic League')
plt.ylabel('Market Value (in Millions of Euro)')
plt.show()
```



Attackers and Midfielders are valued the highest with Goalkeepers generally valued the lowest.

In [145…

```
# Seaborn Scatterplot of Market Value by League and Foot
plt.figure(figsize=(12, 8))
sns.scatterplot(x='current_club_domestic_competition_id', y=current_players_df['mar
plt.title('Individual Players Market Values By Domestic League and Footedness')
plt.xlabel('Domestic League')
plt.ylabel('Market Value (in Millions of Euro)')
plt.show()
```

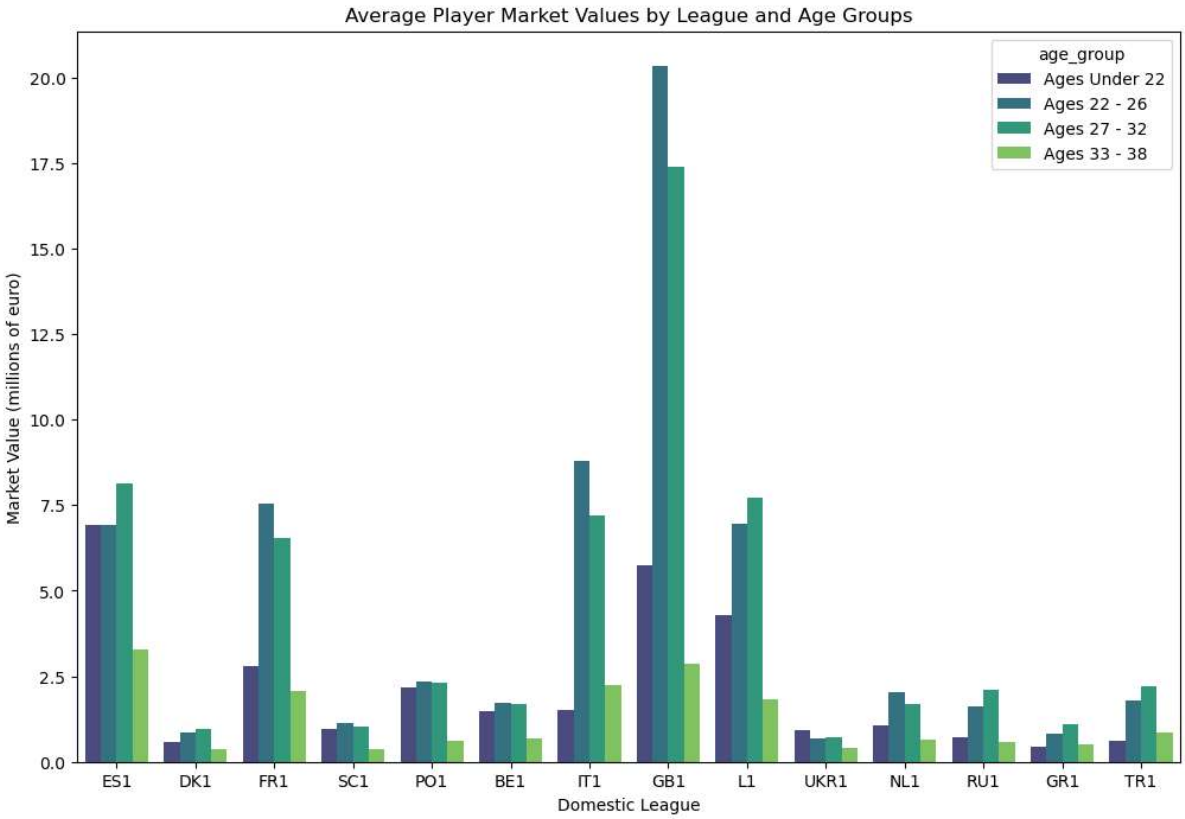


Left footed players are highly valued relative to their market density. Three of the top seven highest valued players are left footed. Additionaly they all apply their trade in the English Premier League.

Average Player Values

In [147…

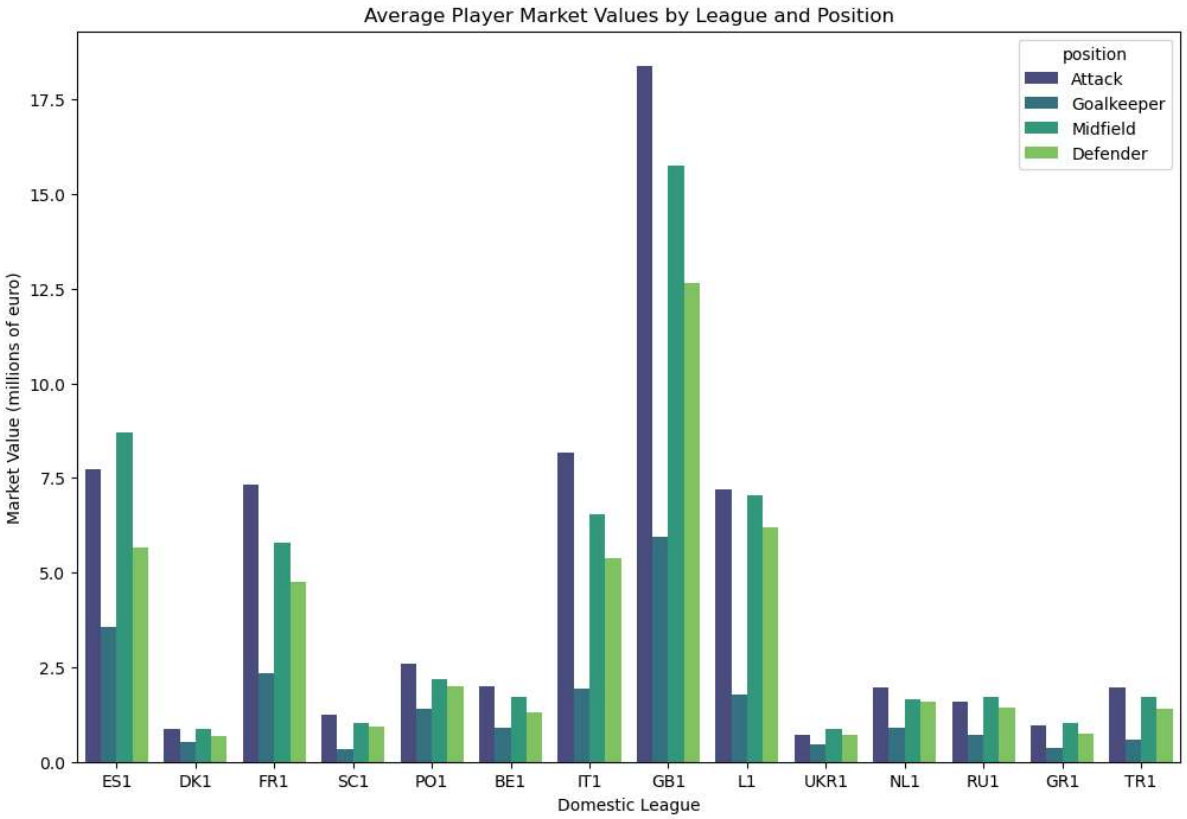
```
plt.figure(figsize=(12, 8))
sns.barplot(x = 'current_club_domestic_competition_id' , y=current_players_df['mark
plt.title("Average Player Market Values by League and Age Groups")
plt.ylabel('Market Value (millions of euro)')
plt.xlabel('Domestic League')
#plt.xticks(rotation = 'vertical')
plt.show()
```



Players aged between 22 and 26 are the highest valued on average. The English Premier League contains the highest average player values for players aged 22 to 26 by more than double that of the next closest league. Spain, England and Germany have the highest average market values for young players under 22. The average market value of player aged between 22 and 26 in The Premier League is around 20 million euro. LaLiga, Ligue 1 and Serie A are th next highest with an average of around 8 million euro.

In [148…

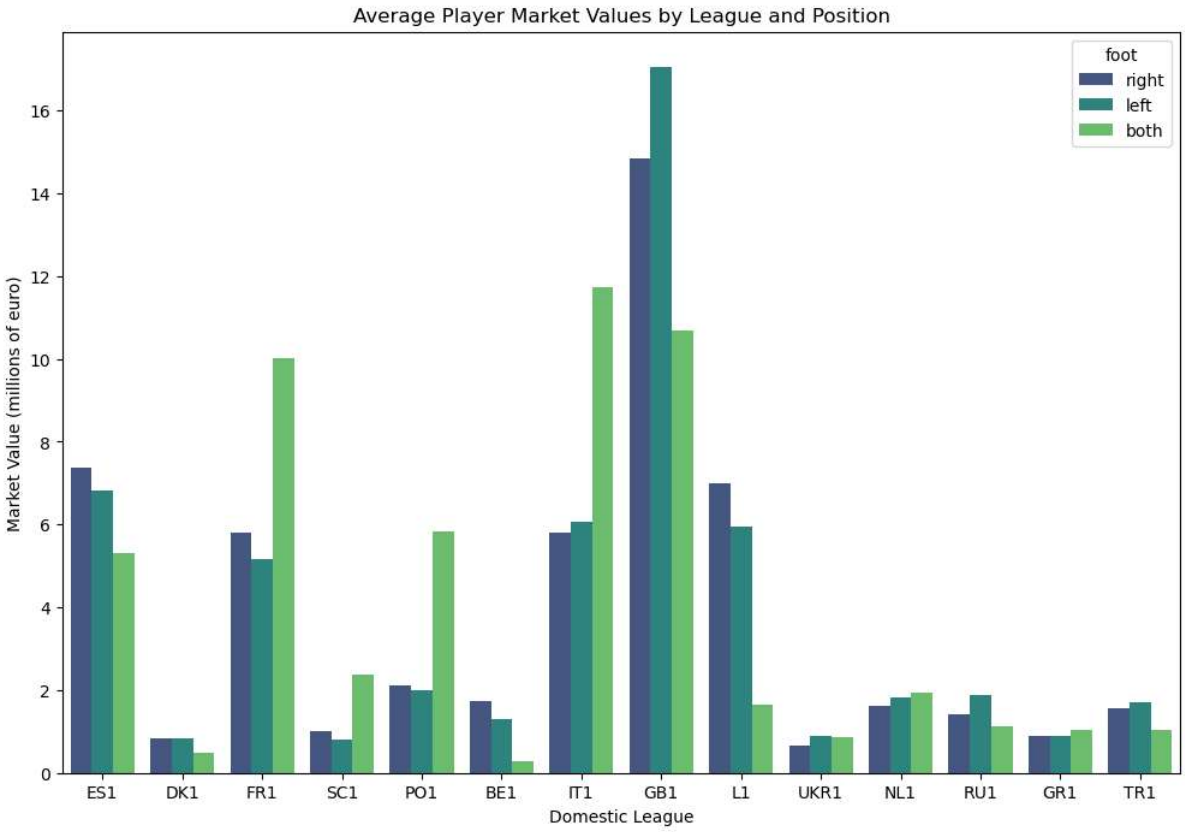
```
plt.figure(figsize=(12, 8))
sns.barplot(x = 'current_club_domestic_competition_id' , y=current_players_df['mark
plt.title("Average Player Market Values by League and Position")
plt.ylabel('Market Value (millions of euro)')
plt.xlabel('Domestic League')
#plt.xticks(rotation = 'vertical')
plt.show()
```



Unsurprisingly, attackers average the highest market value across the majority of domestic leagues. Spain contains more midfielders with high value than attackers, defenders and goalkeepers.

In [149…

```
plt.figure(figsize=(12, 8))
sns.barplot(x = 'current_club_domestic_competition_id' , y=current_players_df['mark
plt.title("Average Player Market Values by League and Position")
plt.ylabel('Market Value (millions of euro)')
plt.xlabel('Domestic League')
#plt.xticks(rotation = 'vertical')
plt.show()
```

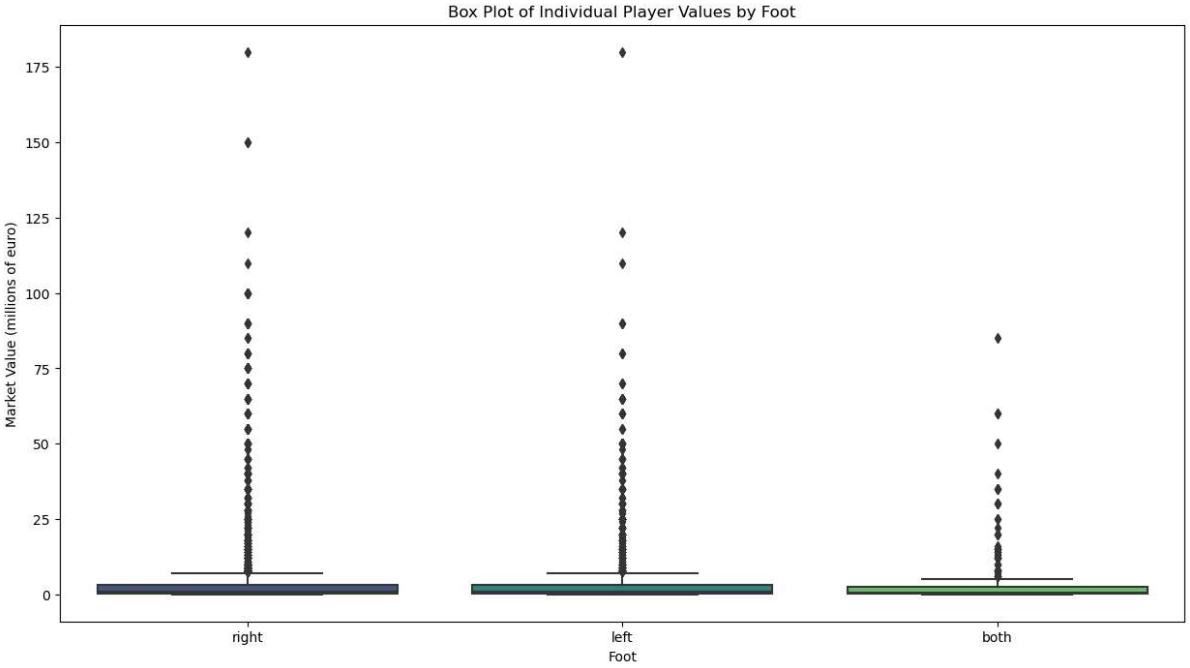


As expected following analysis of the make up of the football player market by footedness, we see in the chart above that left and both footed players combined demand a higher average market value than right footed players. Left footed players in England are very highly valued on average.

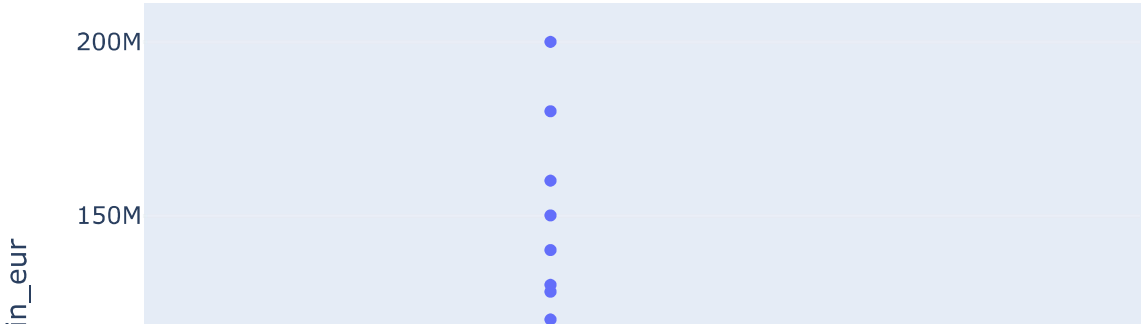
Market Value by player footedness

In [152…

```
# Boxplot of market value by position
plt.figure(figsize=(15, 8))
sns.boxplot(x='foot', y=current_players_df['market_value_in_eur']/1000000, data=cur
plt.xlabel('Foot')
plt.ylabel('Market Value (millions of euro)')
plt.title('Box Plot of Individual Player Values by Foot')
plt.show();
```

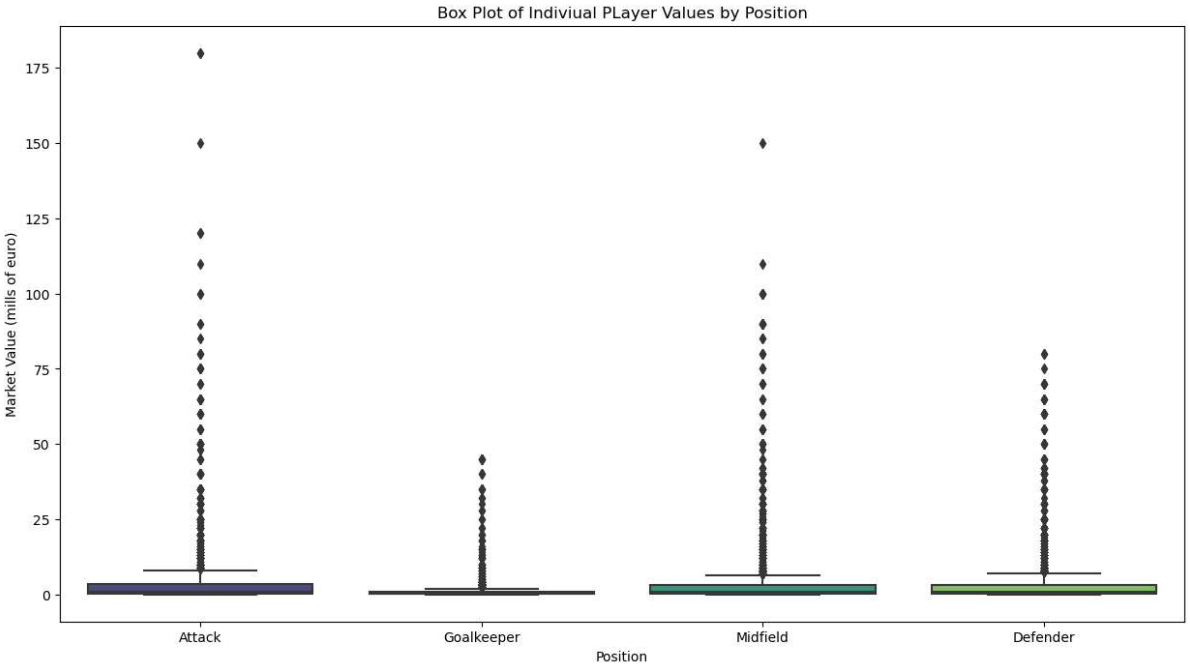


In [74]:
```
# Boxplot of market value by foot
px.box(players_df, x='foot', y='market_value_in_eur')
```



Market value by player position

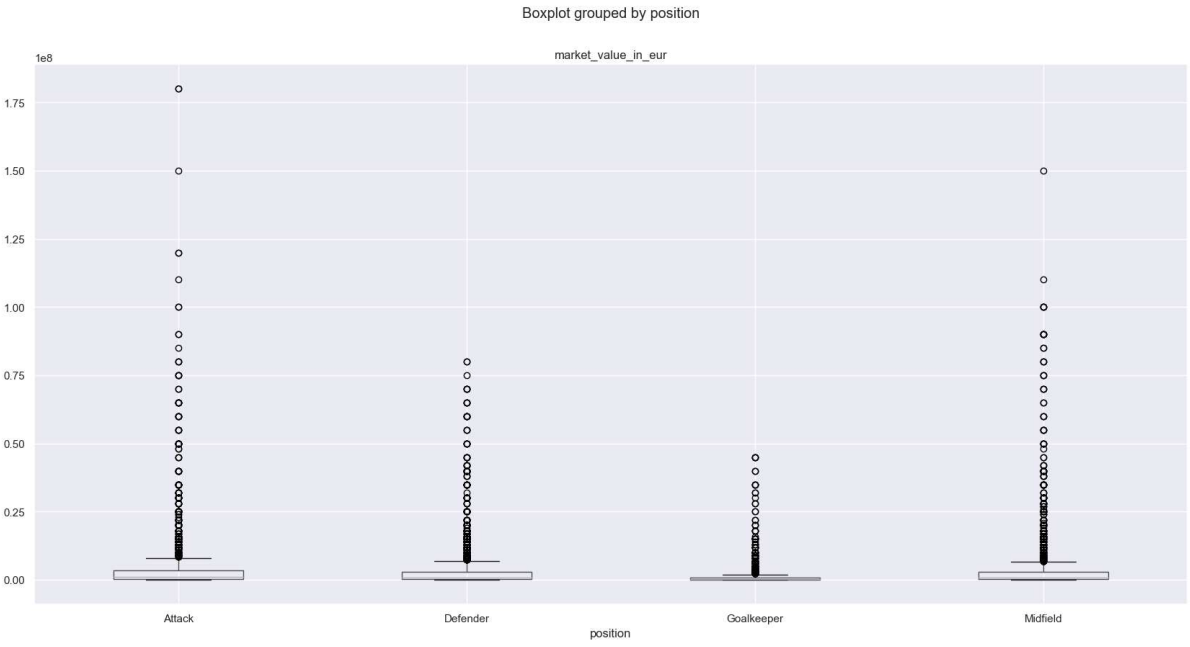
In [153…
```
# Boxplot of market value by position
plt.figure(figsize=(15, 8))
sns.boxplot(x='position', y=current_players_df['market_value_in_eur']/1000000, data
plt.xlabel('Position')
plt.ylabel('Market Value (mills of euro)')
plt.title('Box Plot of Indiviual PLayer Values by Position')
plt.show()
```



In [91]:
```
# Boxplot of market value by position
#px.box(player_valuations_df, x='position', y='market_value_in_eur')
```

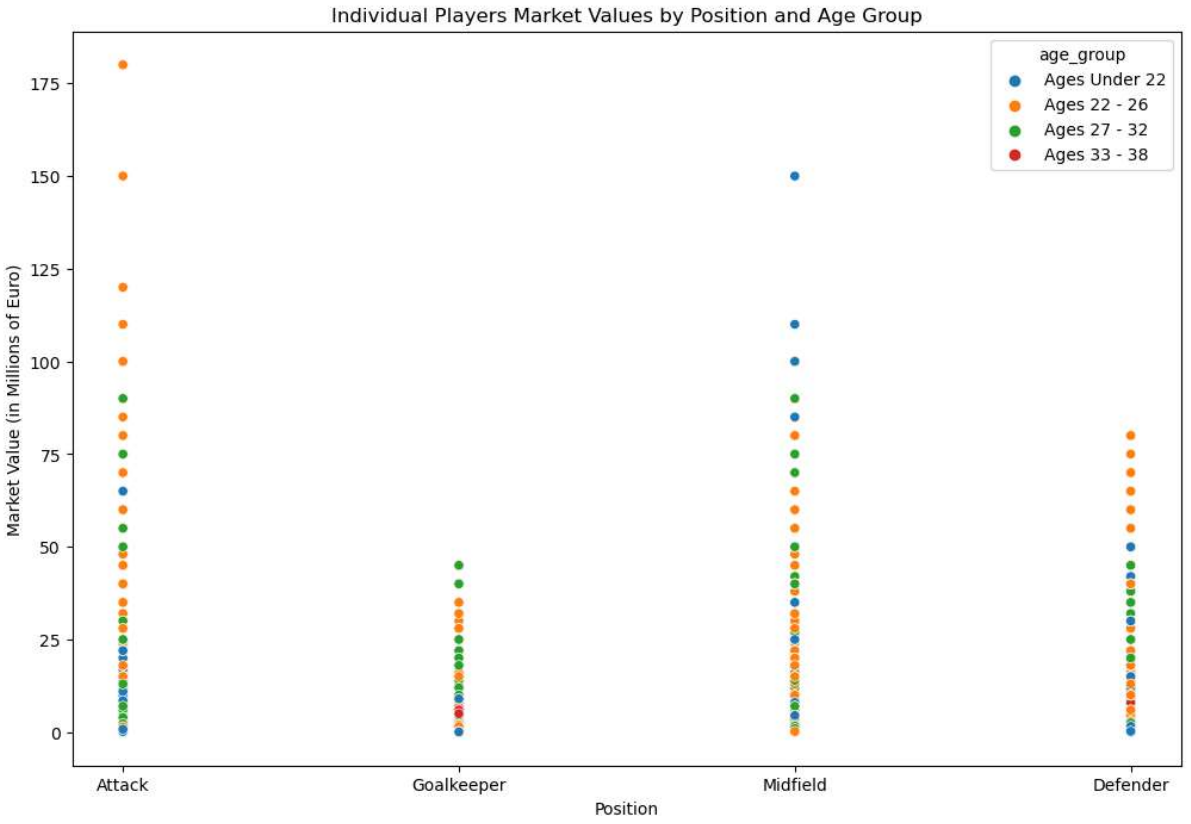
In [392…
```
# Boxplot of market value by position
current_players_df.boxplot('market_value_in_eur', by='position')
```

Out[392]:
```
<Axes: title={'center': 'market_value_in_eur'}, xlabel='position'>
```



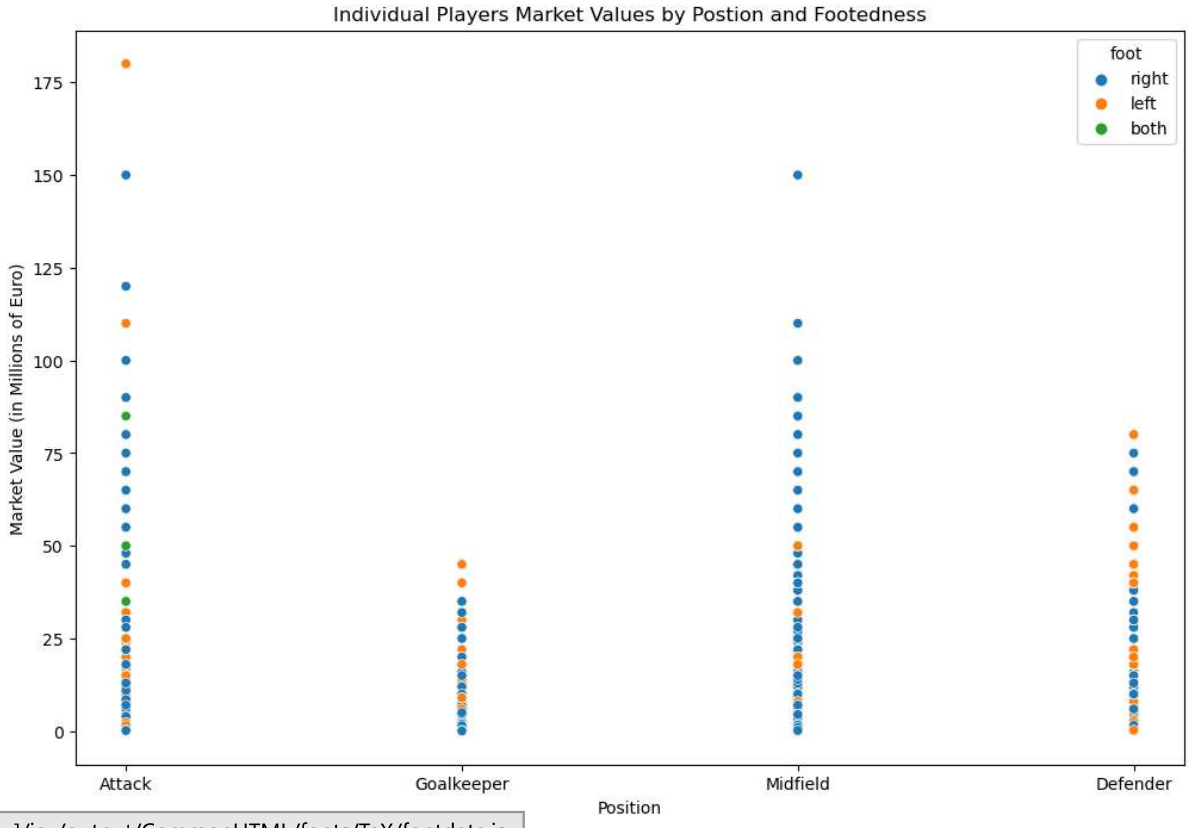
In [158…
```
# Seaborn Scatterplot of Market Value by Position and Age
plt.figure(figsize=(12, 8))
sns.scatterplot(x='position', y=current_players_df['market_value_in_eur']/1000000,
                Values by Position and Age Group')
plt.xlabel('Position')
```

```
plt.ylabel('Market Value (in Millions of Euro)')
plt.show()
```



In [156…

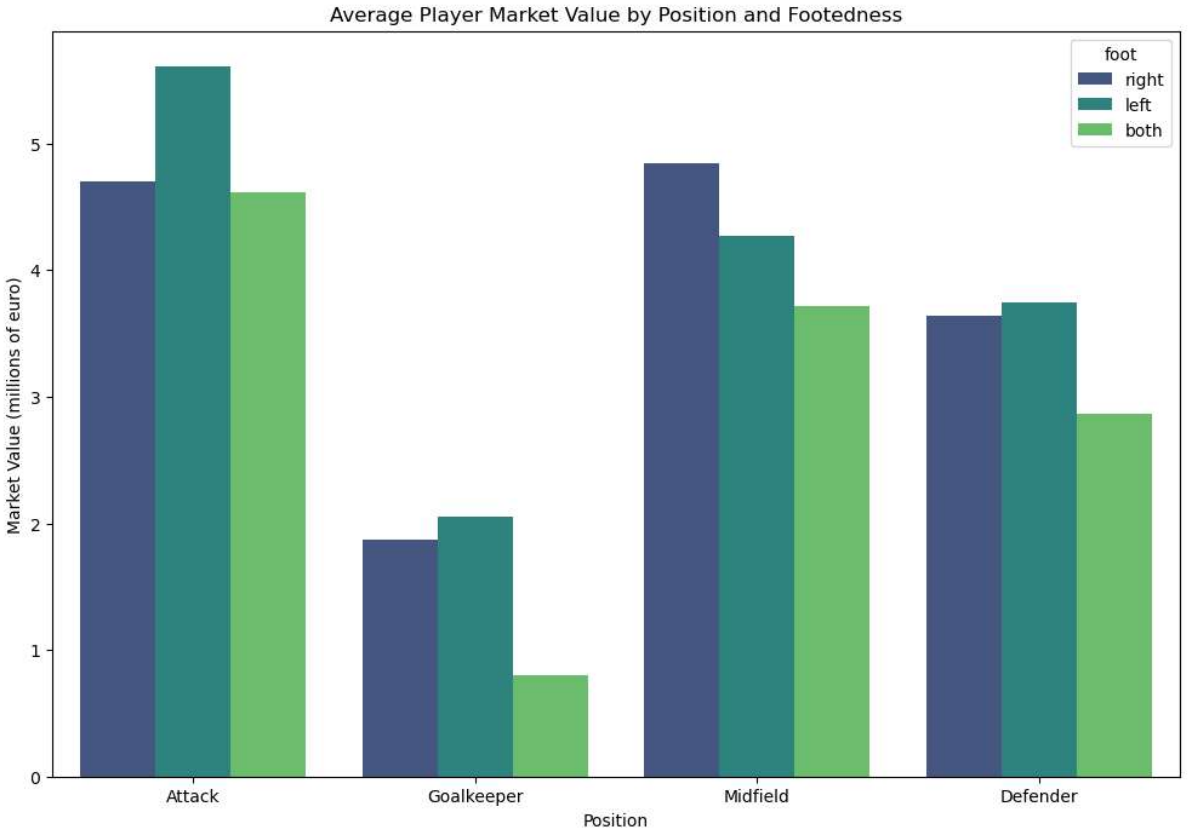
```
# Seaborn Scatterplot of Market Value by Position and Foot
plt.figure(figsize=(12, 8))
sns.scatterplot(x='position', y=current_players_df['market_value_in_eur']/1000000,
plt.title('Individual Players Market Values by Postion and Footedness')
plt.xlabel('Position')
plt.ylabel('Market Value (in Millions of Euro)')
plt.show()
```



Averaged Player Values

In [159…

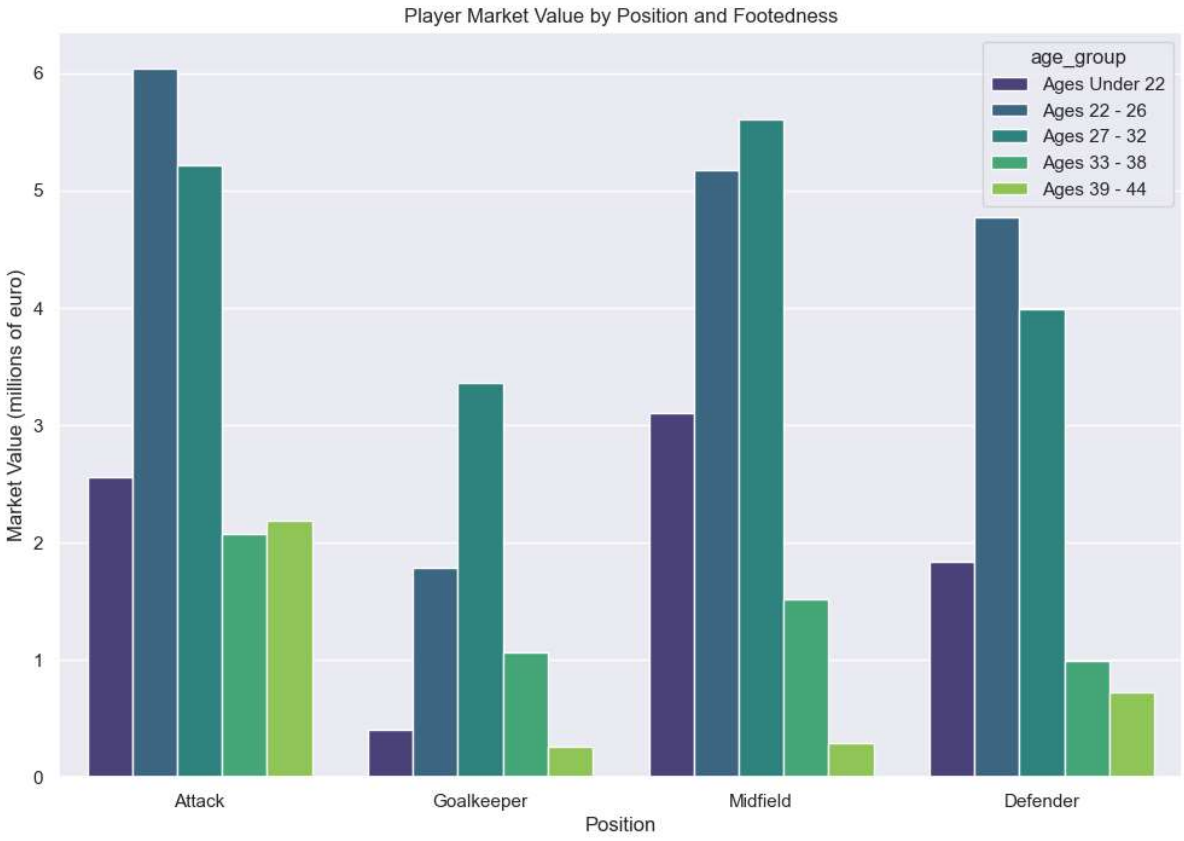
```
plt.figure(figsize=(12, 8))
sns.barplot(x = 'position' , y=current_players_df['market_value_in_eur']/1000000, h
plt.title("Average Player Market Value by Position and Footedness")
plt.ylabel('Market Value (millions of euro)')
plt.xlabel('Position')
#plt.xticks(rotation = 'vertical')
plt.show()
```



Left footed attackers seem to be valued the highest. This is little surprise following earlier analysis of current players and their footedness which showed that only 25.67% of current players are left footed. There is clearly a shortage and demand for such players.

In [425…

```
plt.figure(figsize=(12, 8))
sns.barplot(x = 'position' , y=current_players_df['market_value_in_eur']/1000000, h
plt.title("Player Market Value by Position and Footedness")
plt.ylabel('Market Value (millions of euro)')
plt.xlabel('Position')
#plt.xticks(rotation = 'vertical')
plt.show()
```



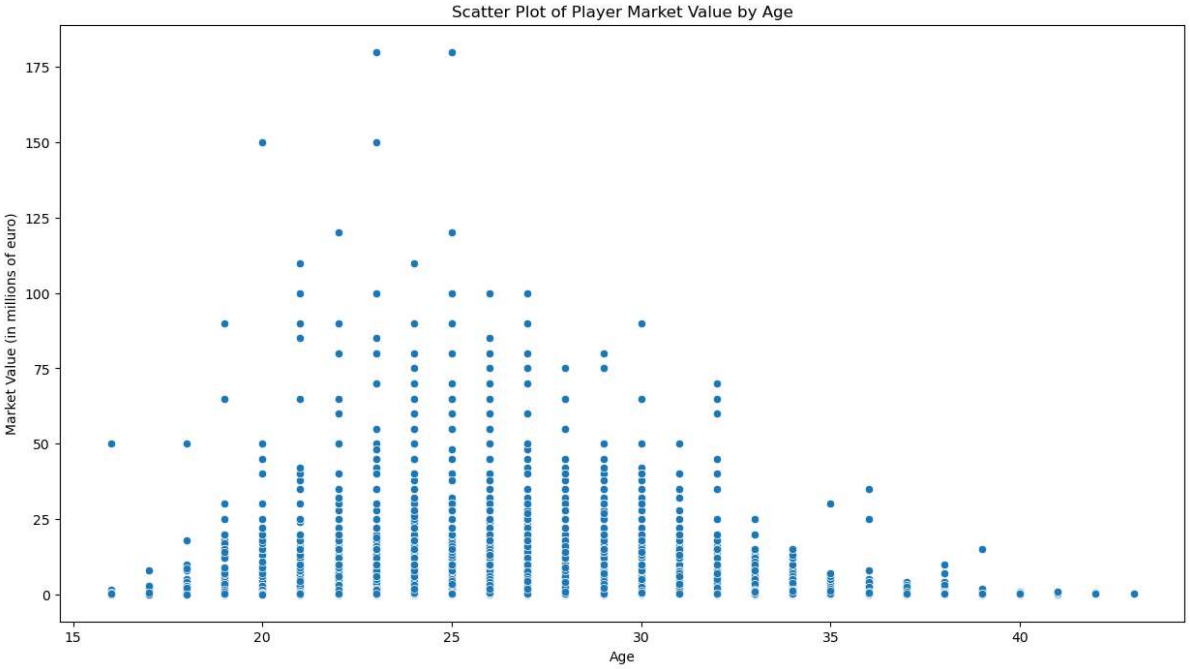
The highest valued age category appears to be between the ages of 22 and 26. However in the goalkeeping department, age and experience is slighty more valued.

Market Value by player Age

Individual Player Values

In [167…

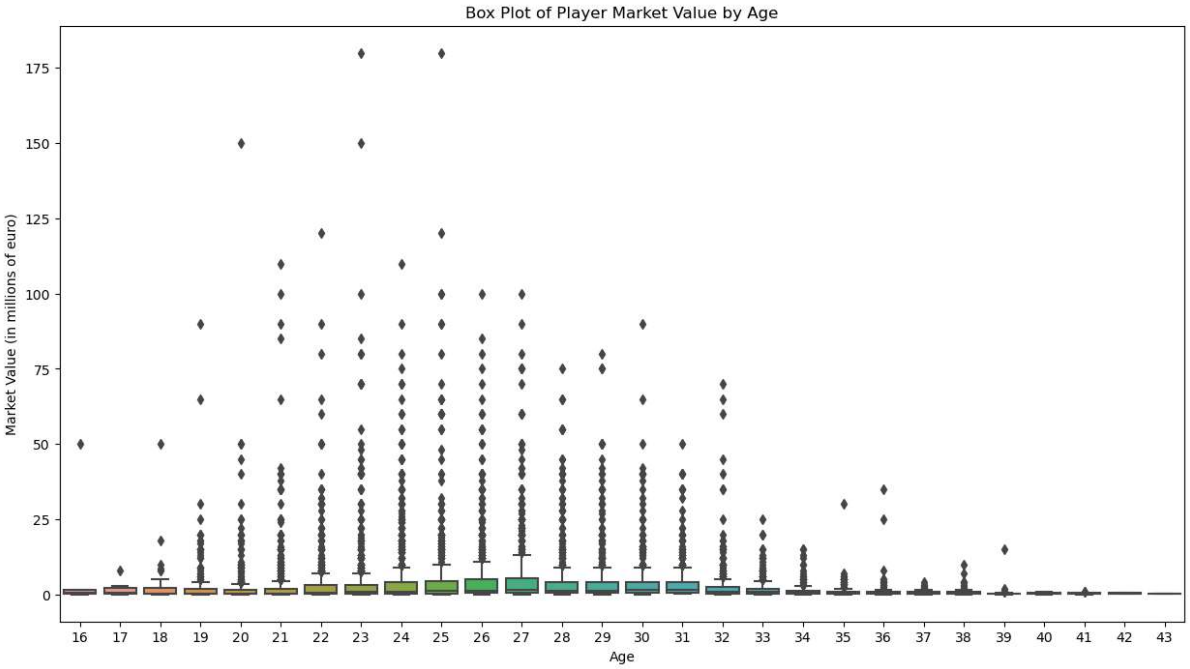
```
# Scatterplot of market value by age
plt.figure(figsize=(15, 8))
sns.scatterplot(x='age', y=current_players_df['market_value_in_eur']/1000000, data=
plt.xlabel('Age')
plt.ylabel('Market Value (in millions of euro)')
plt.title('Scatter Plot of Player Market Value by Age')
plt.show()
```



We see that the highest valued players at around 100 million euro or more are aged between 20 and 27 years.

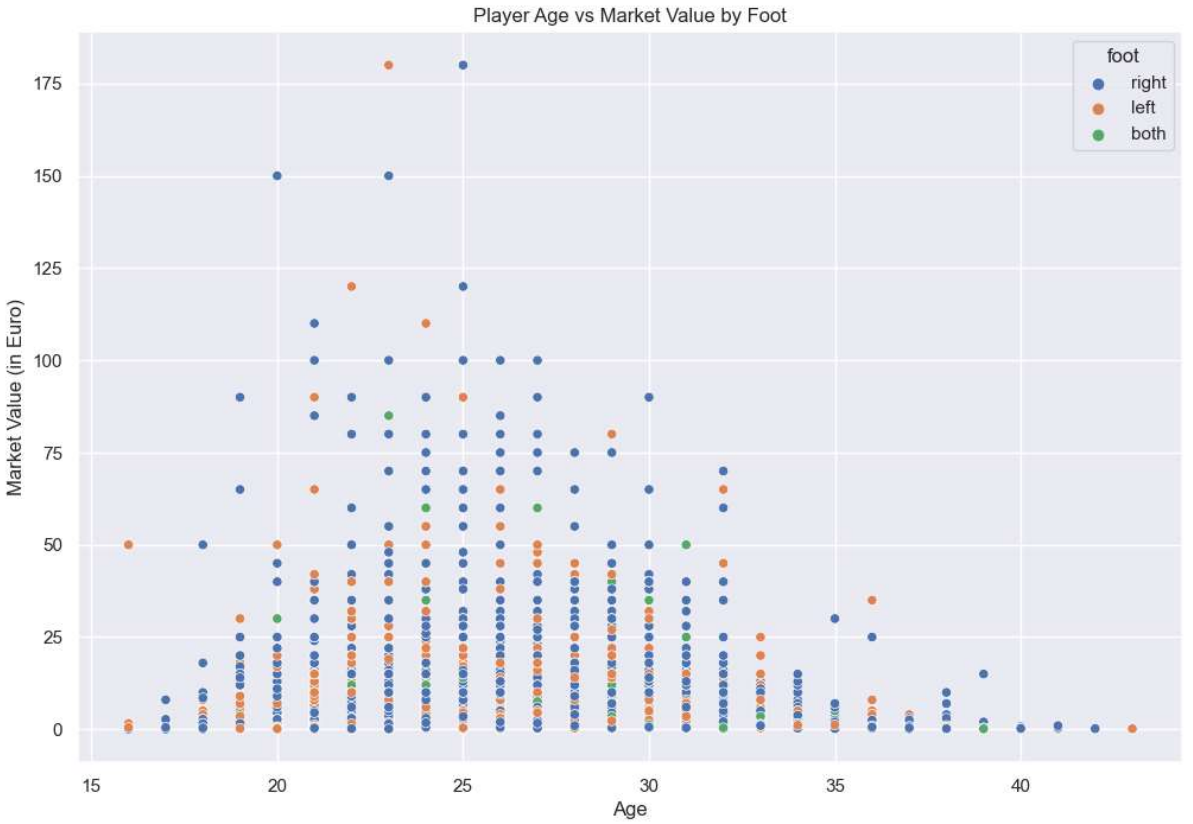
In [88]:
```
# Scatterplot of market value by age
#px.scatter(player_valuations_df, x='age', y='market_value_in_eur')
```

In [161…
```
# Boxplot of market value by position
plt.figure(figsize=(15, 8))
sns.boxplot(x='age', y=current_players_df['market_value_in_eur']/1000000, data= cur
plt.xlabel('Age')
plt.ylabel('Market Value (in millions of euro)')
plt.title('Box Plot of Player Market Value by Age')
plt.show()
```



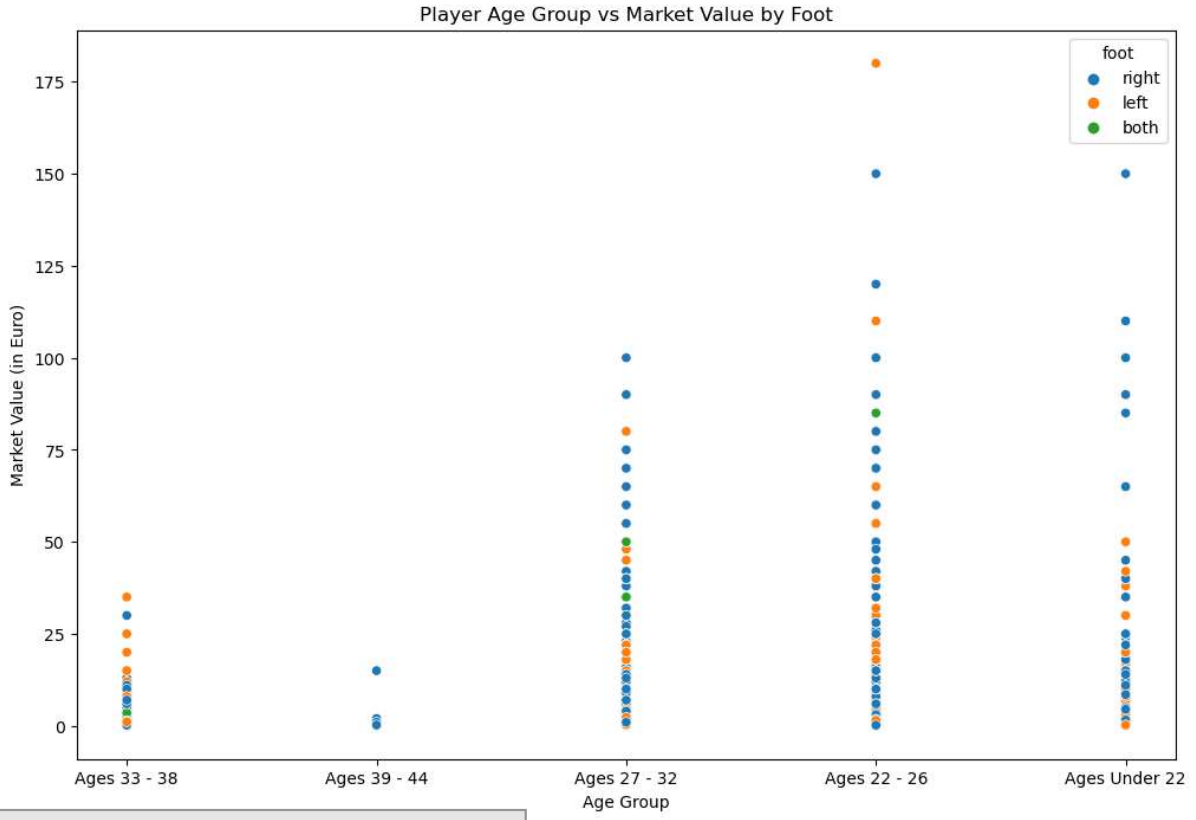
In [430…
```
# Seaborn Scatterplot for Market Value by Age and Foot
plt.figure(figsize=(12, 8))
sns.scatterplot(x='age', y=current_players_df['market_value_in_eur']/1000000, hue='
e by Foot')
```

```
plt.ylabel('Market Value (in Euro)')
plt.show()
```



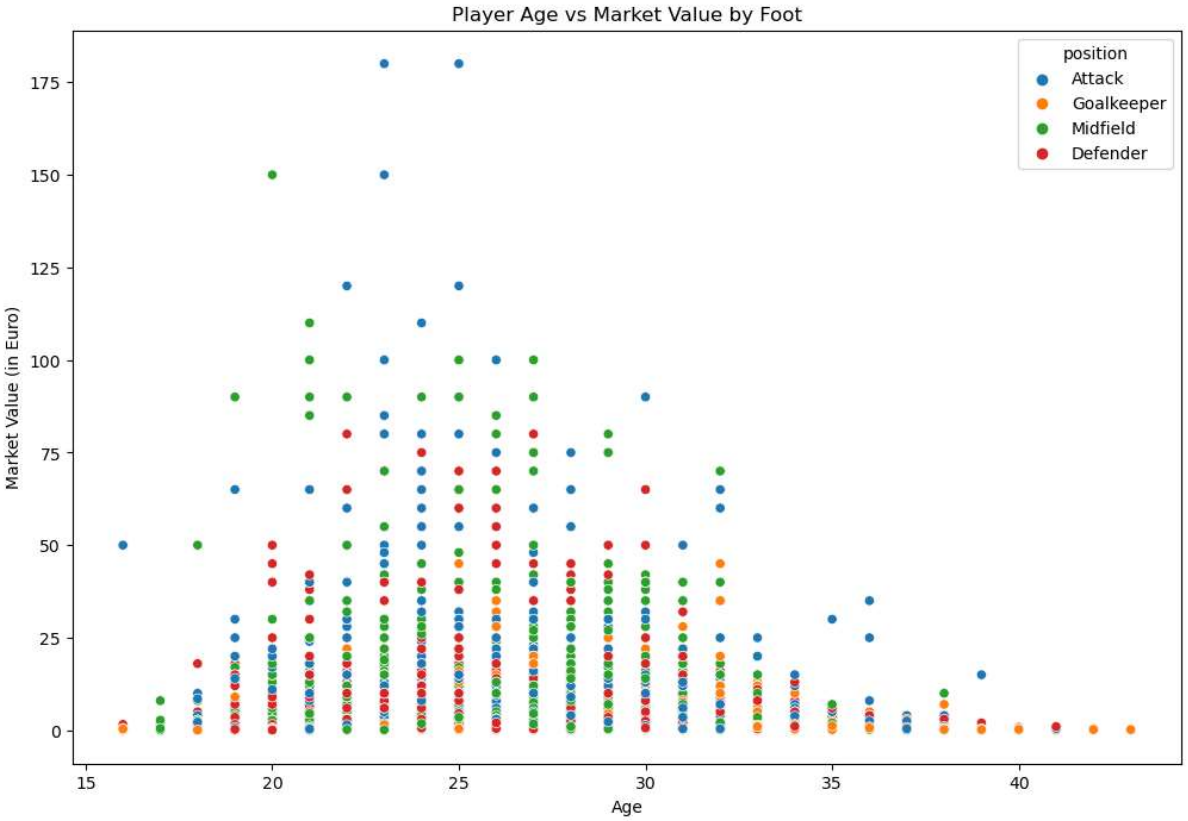
In [168…

```
# Seaborn Scatterplot for Market Value bu age groups and foot
plt.figure(figsize=(12, 8))
sns.scatterplot(x='age_group', y=current_players_df['market_value_in_eur']/1000000,
plt.title('Player Age Group vs Market Value by Foot')
plt.xlabel('Age Group')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



In [162…

```
# Seaborn Scatterplot for Market Value by Age and Position
plt.figure(figsize=(12, 8))
sns.scatterplot(x='age', y=current_players_df['market_value_in_eur']/1000000, hue='
plt.title('Player Age vs Market Value by Position')
plt.xlabel('Age')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



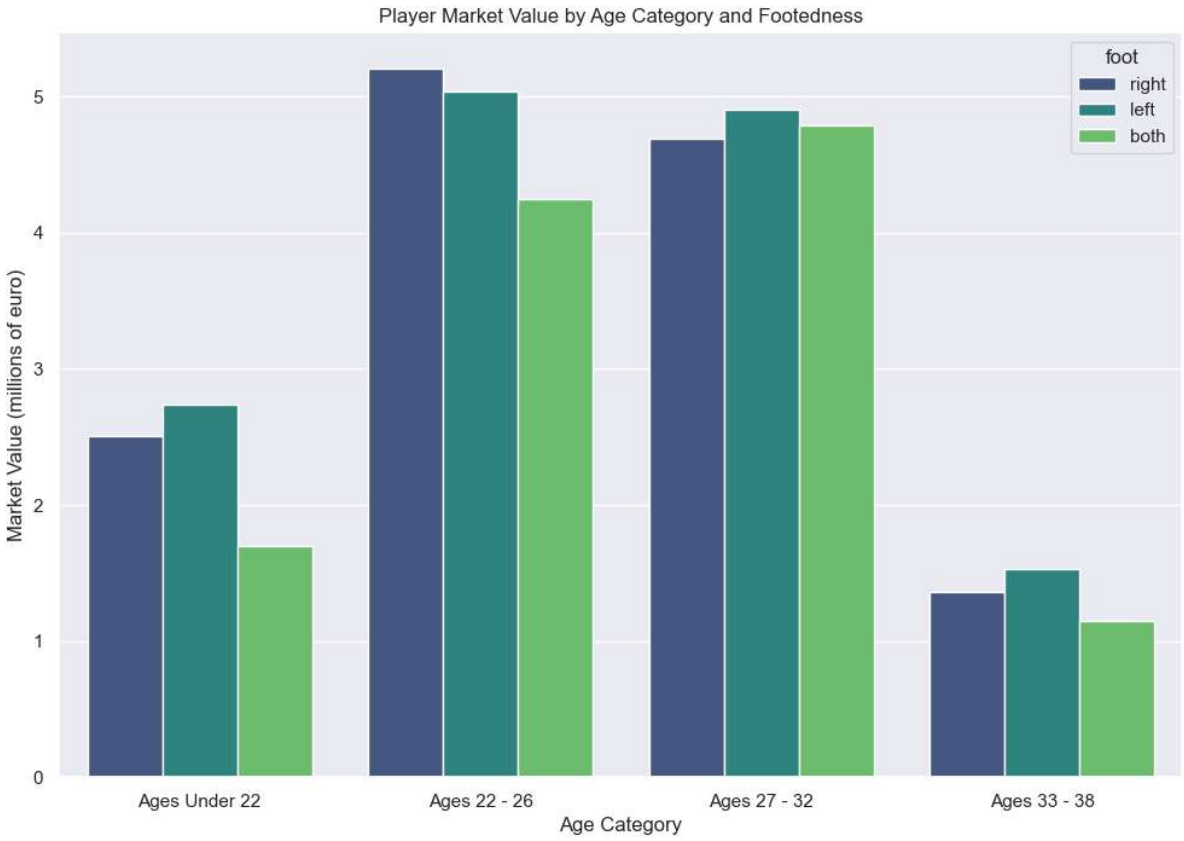
In [ ]:

In [ ]:

Averaged Player Values

In [624…

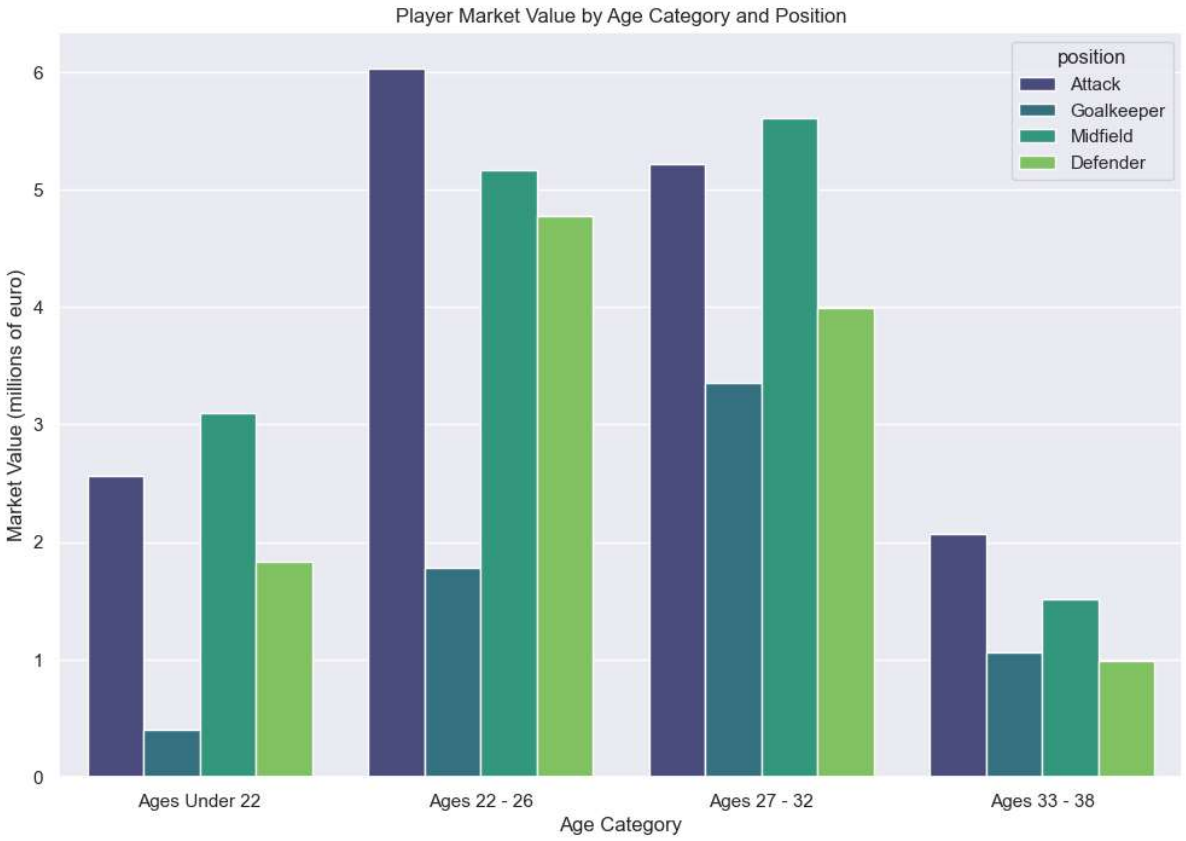
```
plt.figure(figsize=(12, 8))
sns.barplot(x = 'age_group' , y=current_players_df['market_value_in_eur']/1000000,
plt.title("Player Market Value by Age Category and Footedness")
plt.ylabel('Market Value (millions of euro)')
plt.xlabel('Age Category')
#plt.xticks(rotation = 'vertical')
plt.show()
```



Across all domestic leagues, players of all footedness average a higher price between the agges of 22 and 32. The average market value of players is around 5 million or under across all leagues.

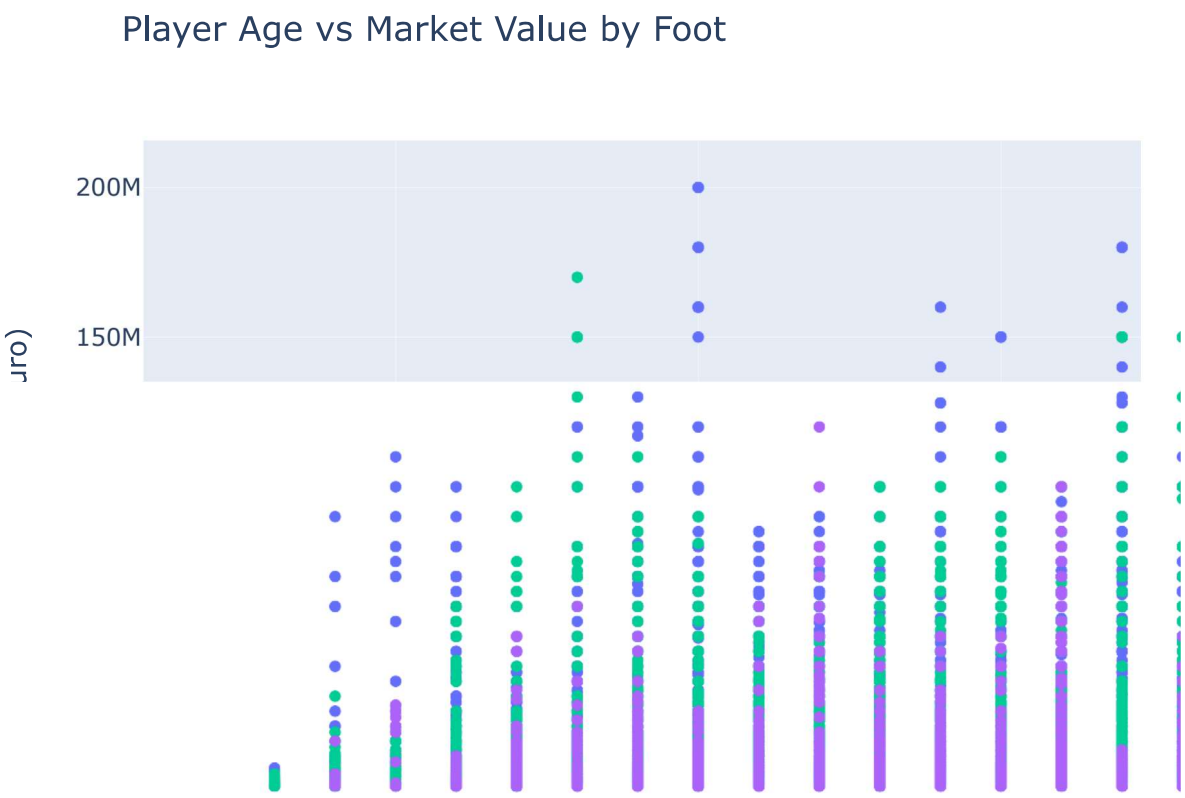
In [629…

```
plt.figure(figsize=(12, 8))
sns.barplot(x = 'age_group' , y=current_players_df['market_value_in_eur']/1000000,
plt.title("Player Market Value by Age Category and Position")
plt.ylabel('Market Value (millions of euro)')
plt.xlabel('Age Category')
#plt.xticks(rotation = 'vertical')
plt.show()
```



The average market value across all leagues is for attackers aged between 22 and 26 with an average value of around 6 million euro. Midfielders are valued higher on avergae at between the ages of 27 and 32 years old.

In [85]:
```
# Plotly Scatter
fig = px.scatter(player_valuations_df, x='age', y='market_value_in_eur', color='foo
                 title='Player Age vs Market Value by Foot')
fig.update_layout(xaxis_title='Age', yaxis_title='Market Value (in Euro)')
fig.show()
```



Analysis of Market Value by Goals scored in a Year

In [170…

```
current_players_df_with_stats.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42736 entries, 0 to 42735
Data columns (total 31 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   player_id                             42736 non-null  int64
 1   first_name                            40076 non-null  object
 2   last_name                             42736 non-null  object
 3   name                                  42736 non-null  object
 4   last_season                           42736 non-null  int64
 5   current_club_id                       42736 non-null  int64
 6   player_code                           42736 non-null  object
 7   country_of_birth                      39927 non-null  object
 8   city_of_birth                         41770 non-null  object
 9   country_of_citizenship                40847 non-null  object
 10  date_of_birth                         42736 non-null  datetime64[ns]
 11  sub_position                          42736 non-null  object
 12  position                              42736 non-null  object
 13  foot                                  42196 non-null  object
 14  height_in_cm                          42260 non-null  float64
 15  market_value_in_eur                   42736 non-null  float64
 16  highest_market_value_in_eur           42736 non-null  float64
 17  contract_expiration_date              38142 non-null  datetime64[ns]
 18  image_url                             42736 non-null  object
 19  url                                   42736 non-null  object
 20  current_club_domestic_competition_id  42736 non-null  object
 21  current_club_name                     42736 non-null  object
 22  age                                   42736 non-null  int32
 23  remaining_contract_days               38142 non-null  float64
 24  age_group                             42736 non-null  object
 25  year                                  42736 non-null  int32
 26  yellow_cards                          42736 non-null  int64
 27  red_cards                             42736 non-null  int64
 28  goals                                 42736 non-null  int64
 29  assists                               42736 non-null  int64
 30  minutes_played                        42736 non-null  int64
dtypes: datetime64[ns](2), float64(4), int32(2), int64(8), object(15)
memory usage: 9.8+ MB
```

In [171…
```
current_players_df_with_stats['goals'] = current_players_df_with_stats['goals'].fil
```

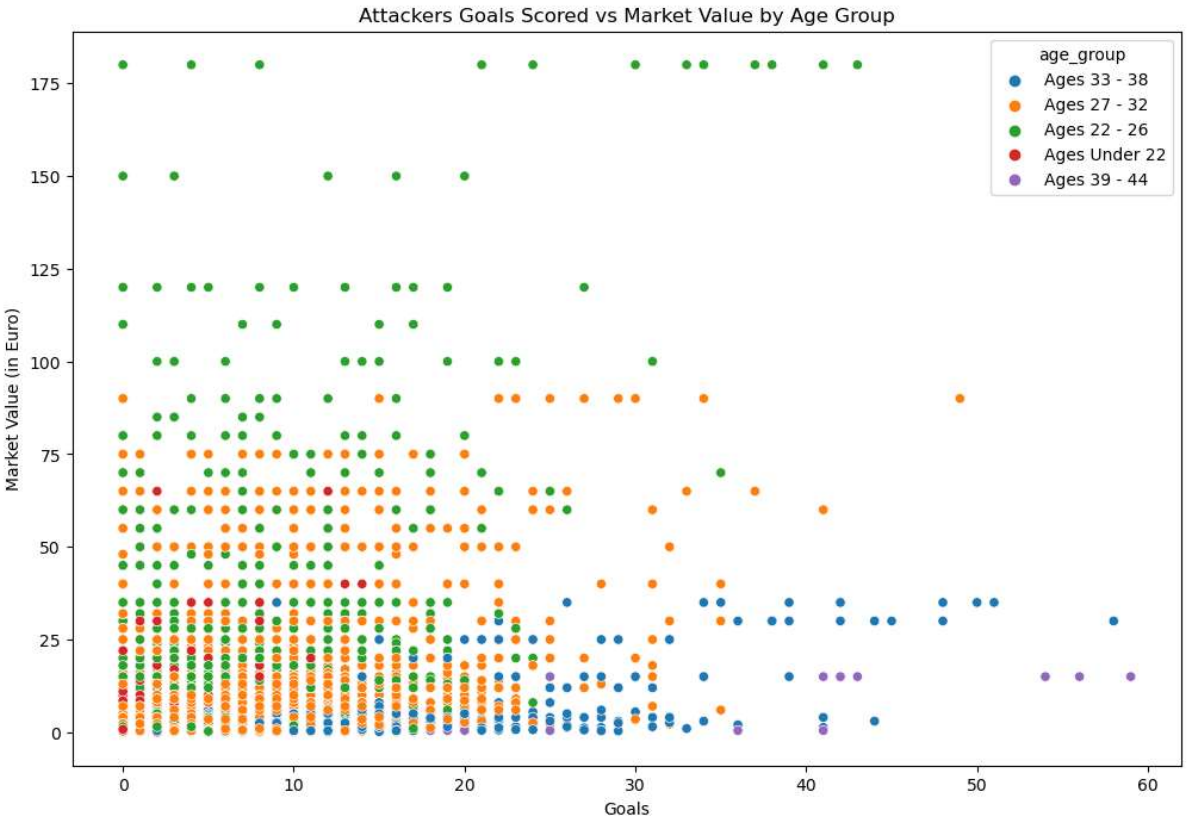
In [172…
```
current_players_df_with_stats['goals'] = current_players_df_with_stats['goals'].fil
```

In [193…
```
attacker_grouped = current_players_df_with_stats[current_players_df_with_stats['pos
midfielder_grouped = current_players_df_with_stats[current_players_df_with_stats['p
defender_grouped = current_players_df_with_stats[current_players_df_with_stats['pos
```

Individual Player Values

Attackers

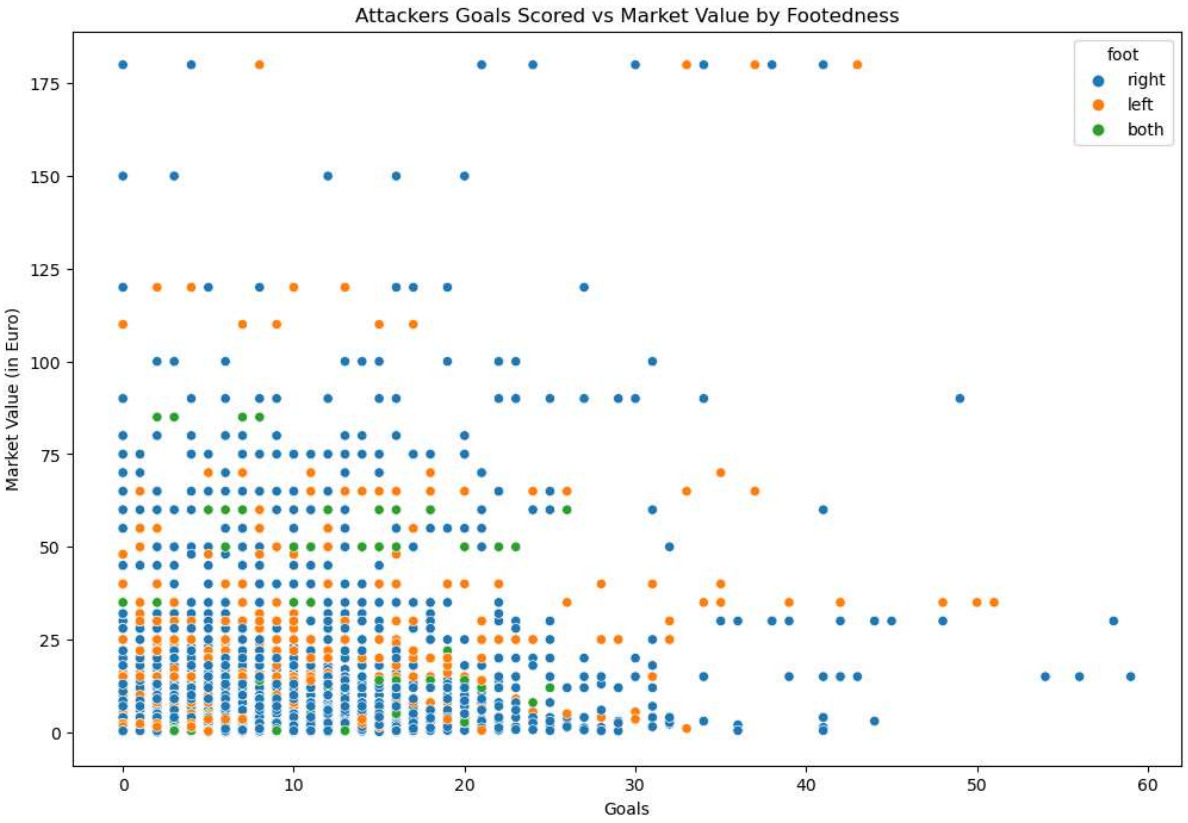
In [189…
```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='goals', y= attacker_grouped['market_value_in_eur']/1000000, hue=
plt.title('Attackers Goals Scored vs Market Value by Age Group')
plt.xlabel('Goals')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



The dataset contains goals within a year for indivual players. The same player may be displayed more than once depending on the goals scored each year. We see that based on this visual there is no direct correlation between goals and market value,

In [190…

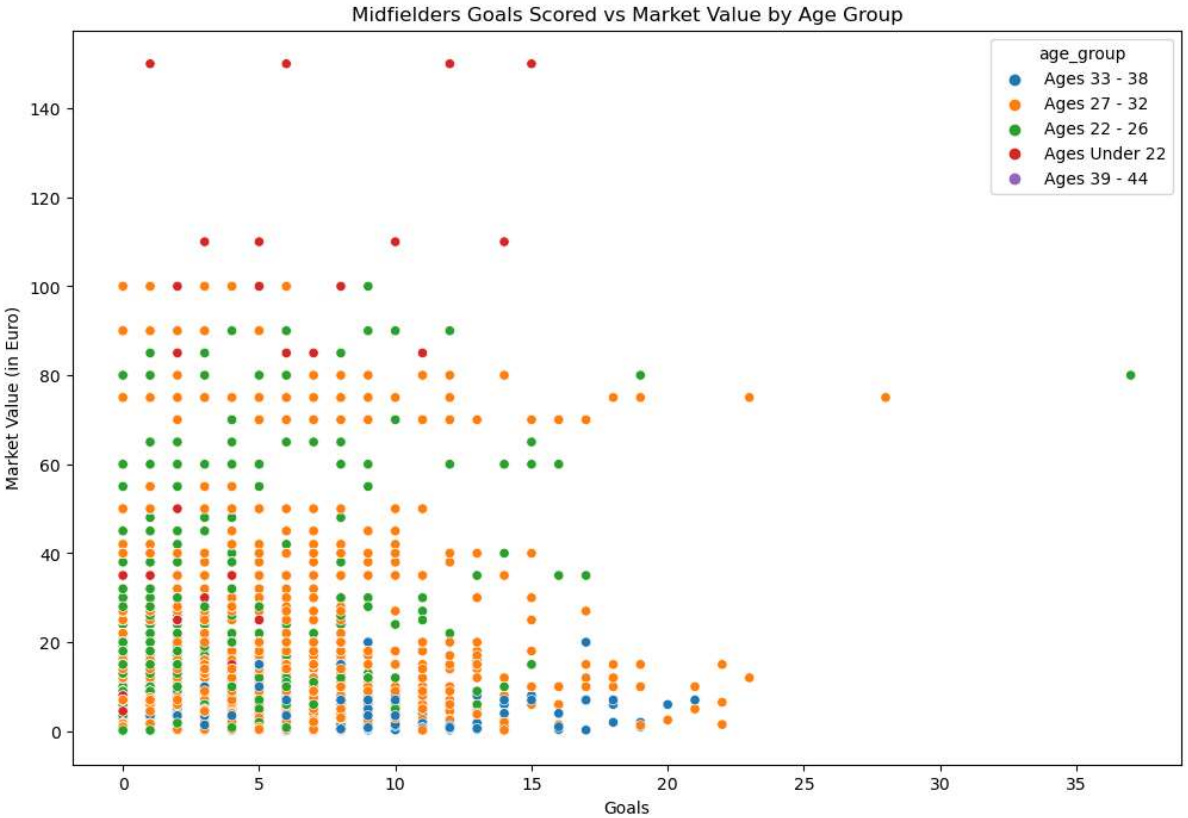
```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='goals', y= attacker_grouped['market_value_in_eur']/1000000, hue=
plt.title('Attackers Goals Scored vs Market Value by Footedness')
plt.xlabel('Goals')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



Midfielders

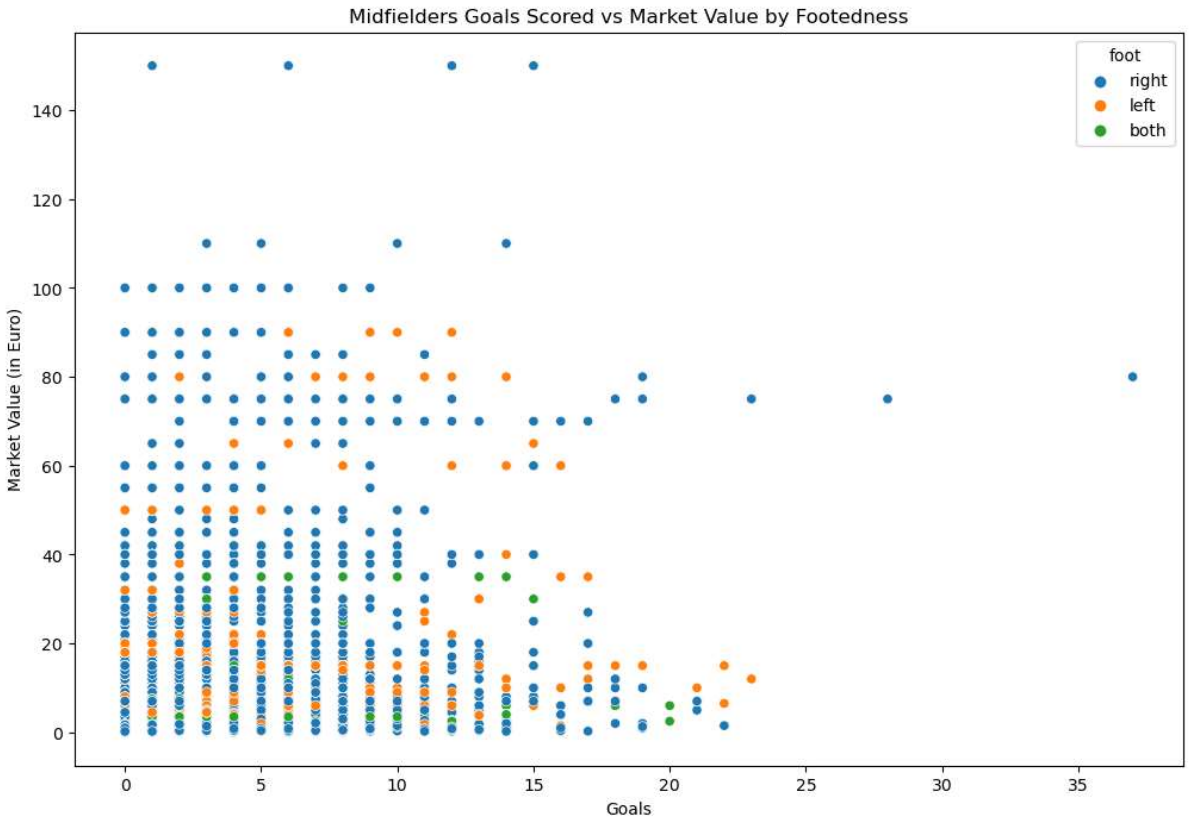
In [199…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='goals', y= midfielder_grouped['market_value_in_eur']/1000000, hu
plt.title('Midfielders Goals Scored vs Market Value by Age Group')
plt.xlabel('Goals')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



In [200…

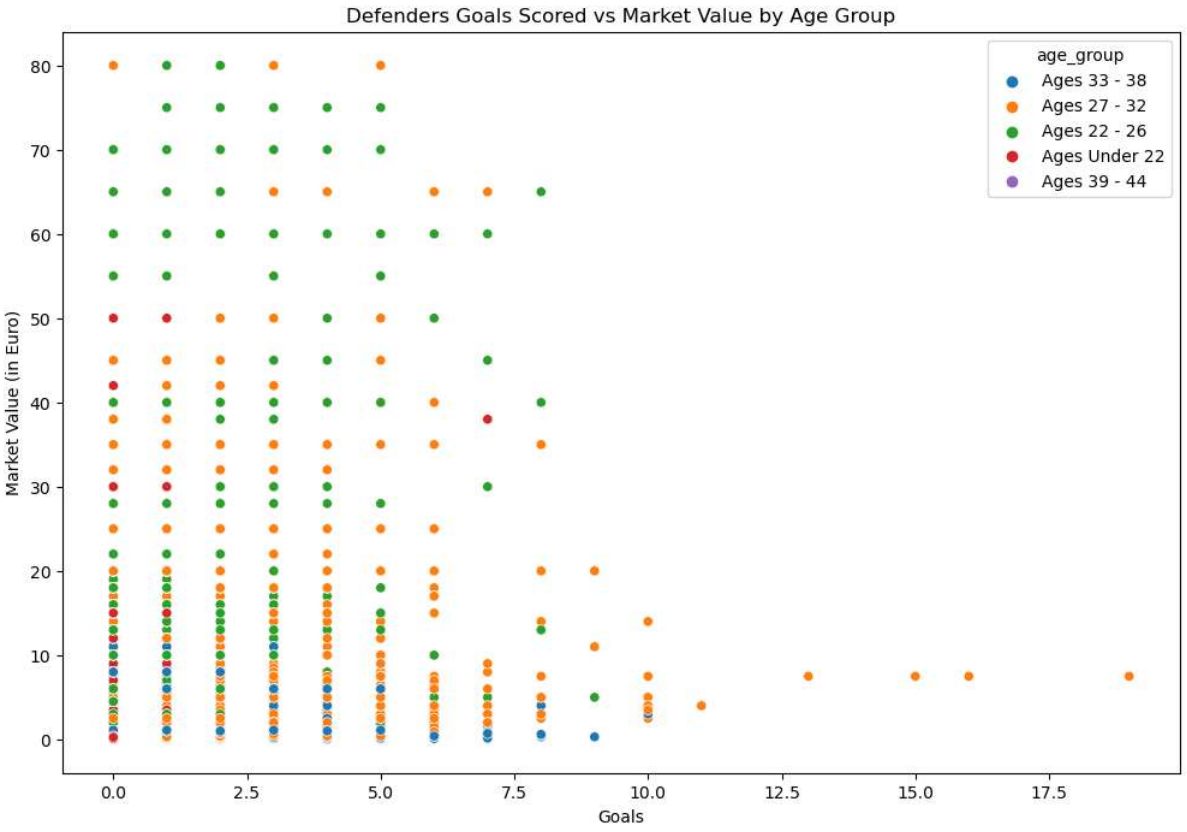
```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='goals', y= midfielder_grouped['market_value_in_eur']/1000000, hu
plt.title('Midfielders Goals Scored vs Market Value by Footedness')
plt.xlabel('Goals')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



Defenders

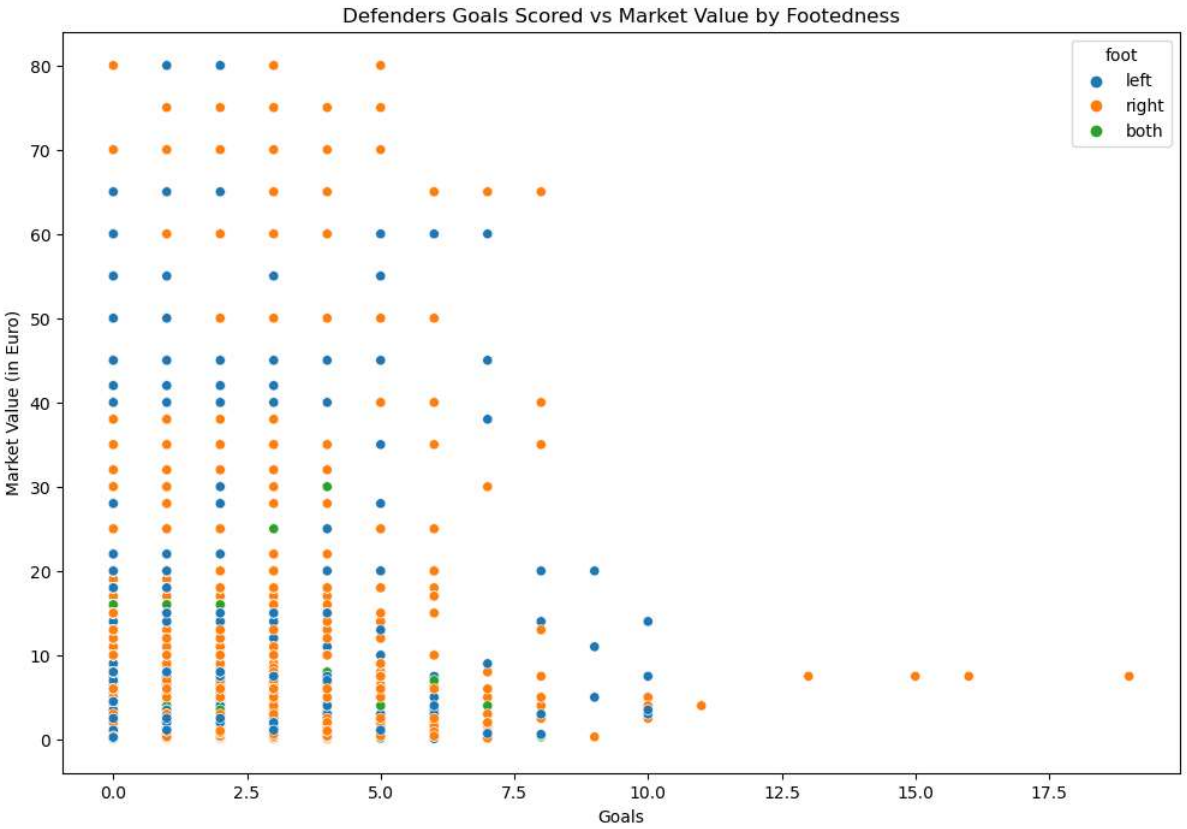
In [201…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='goals', y= defender_grouped['market_value_in_eur']/1000000, hue=
plt.title('Defenders Goals Scored vs Market Value by Age Group')
plt.xlabel('Goals')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



In [202…

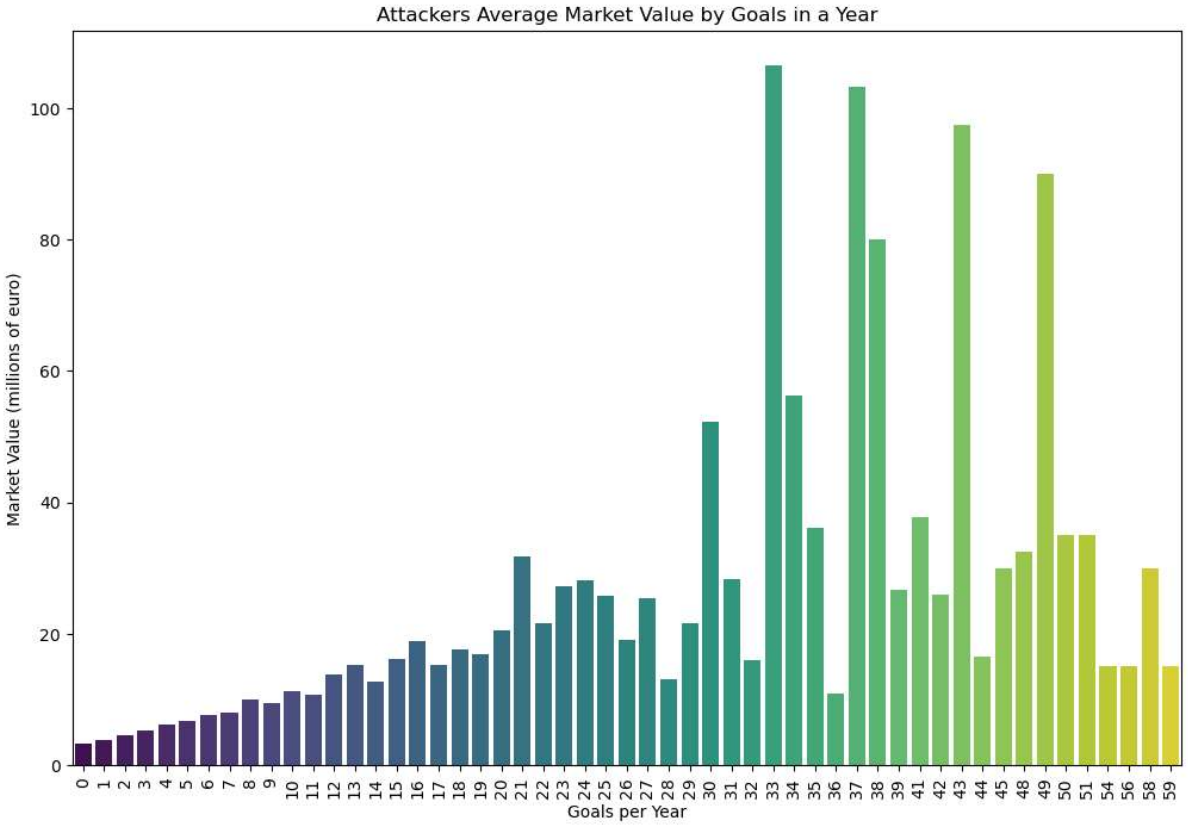
```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='goals', y= defender_grouped['market_value_in_eur']/1000000, hue=
plt.title('Defenders Goals Scored vs Market Value by Footedness')
plt.xlabel('Goals')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



Averaged Player Values

In [182…

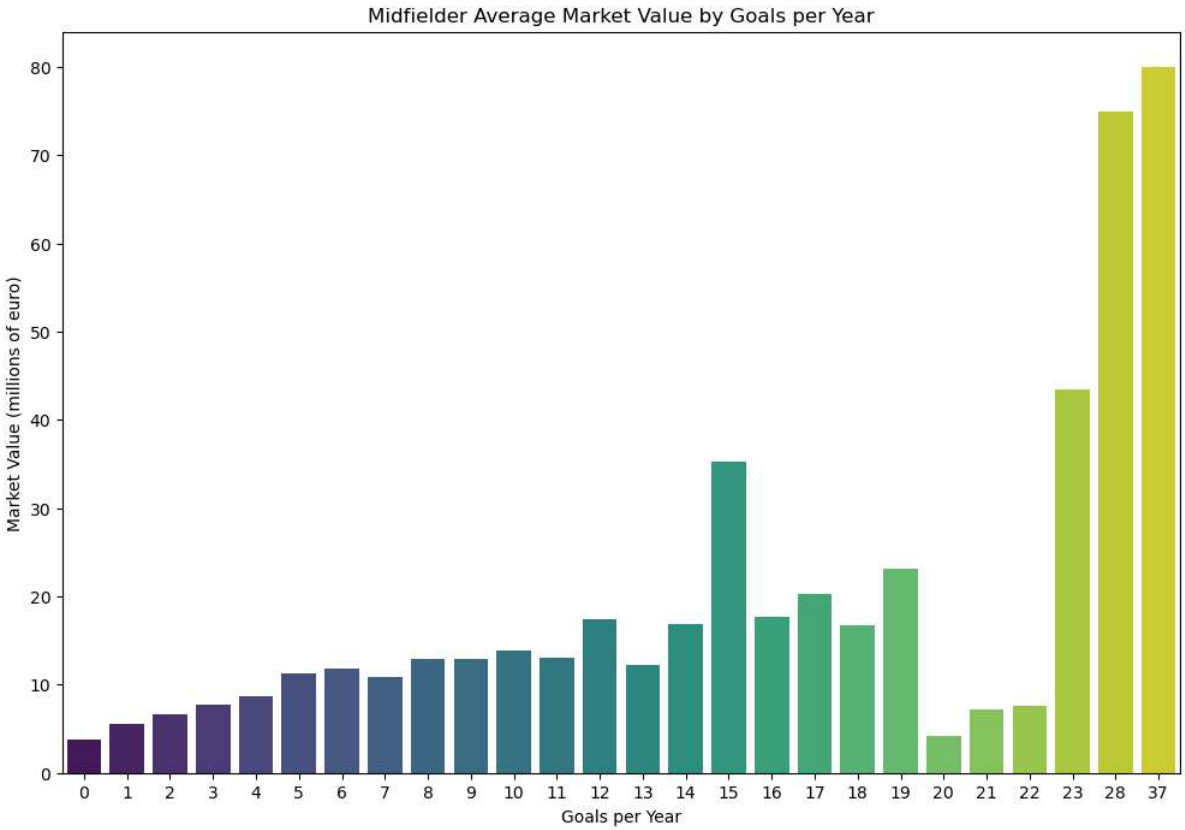
```
plt.figure(figsize=(12, 8))
sns.barplot(x = 'goals' , y=attacker_grouped['market_value_in_eur']/1000000, data =
plt.title("Attackers Average Market Value by Goals in a Year")
plt.ylabel('Market Value (millions of euro)')
plt.xlabel('Goals per Year')
plt.xticks(rotation = 'vertical')
plt.show()
```



As one would expect, there is an increase in market value with an increase in goals scored. There are some exceptions would could be due to factors such as the age of the scoring player or the standard/ domestic league that the goals have been scored in. Naturally, scoring goals in a more competetive league will is rated higher and reflective in market value.

In [196…

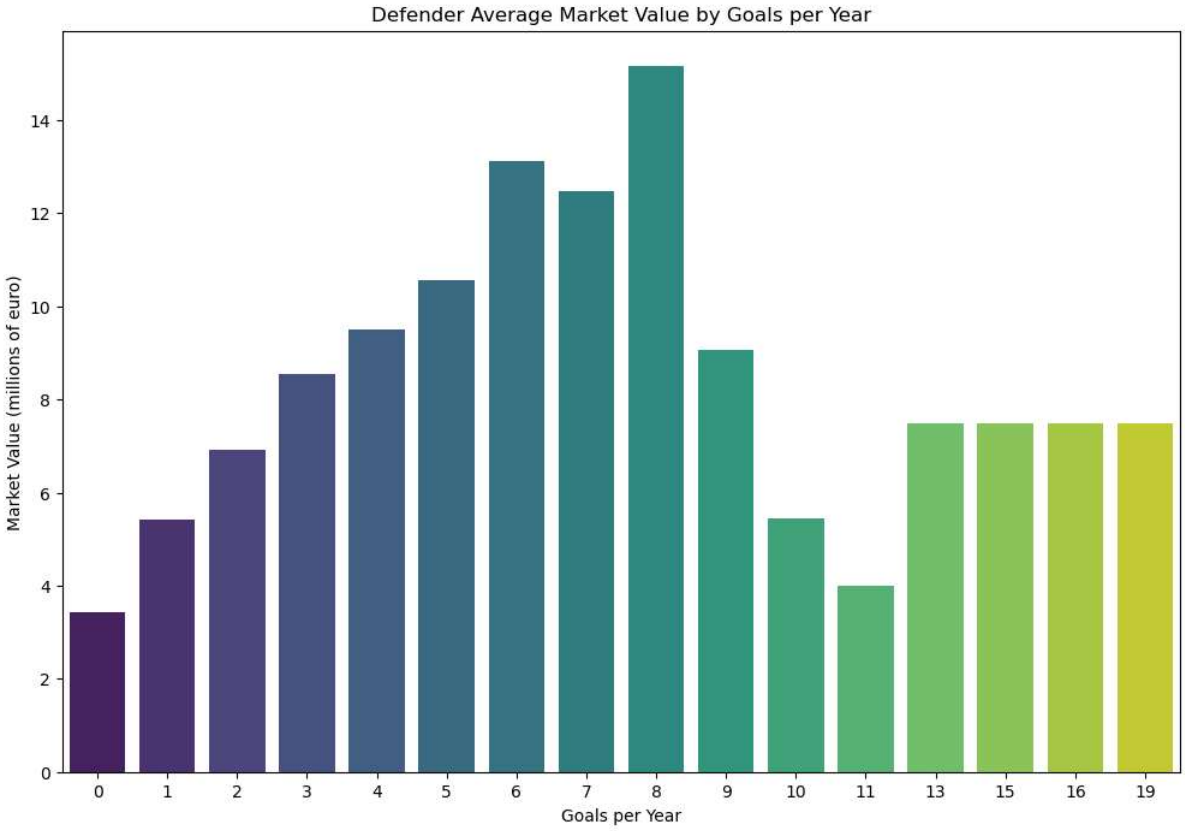
```
plt.figure(figsize=(12, 8))
sns.barplot(x = 'goals' , y=midfielder_grouped['market_value_in_eur']/1000000, data
plt.title("Midfielder Average Market Value by Goals per Year")
plt.ylabel('Market Value (millions of euro)')
plt.xlabel('Goals per Year')
plt.show()
```



In [ ]:

In [192…

```
plt.figure(figsize=(12, 8))
sns.barplot(x = 'goals' , y=defender_grouped['market_value_in_eur']/1000000, data =
plt.title("Defender Average Market Value by Goals per Year")
plt.ylabel('Market Value (millions of euro)')
plt.xlabel('Goals per Year')
plt.show()
```



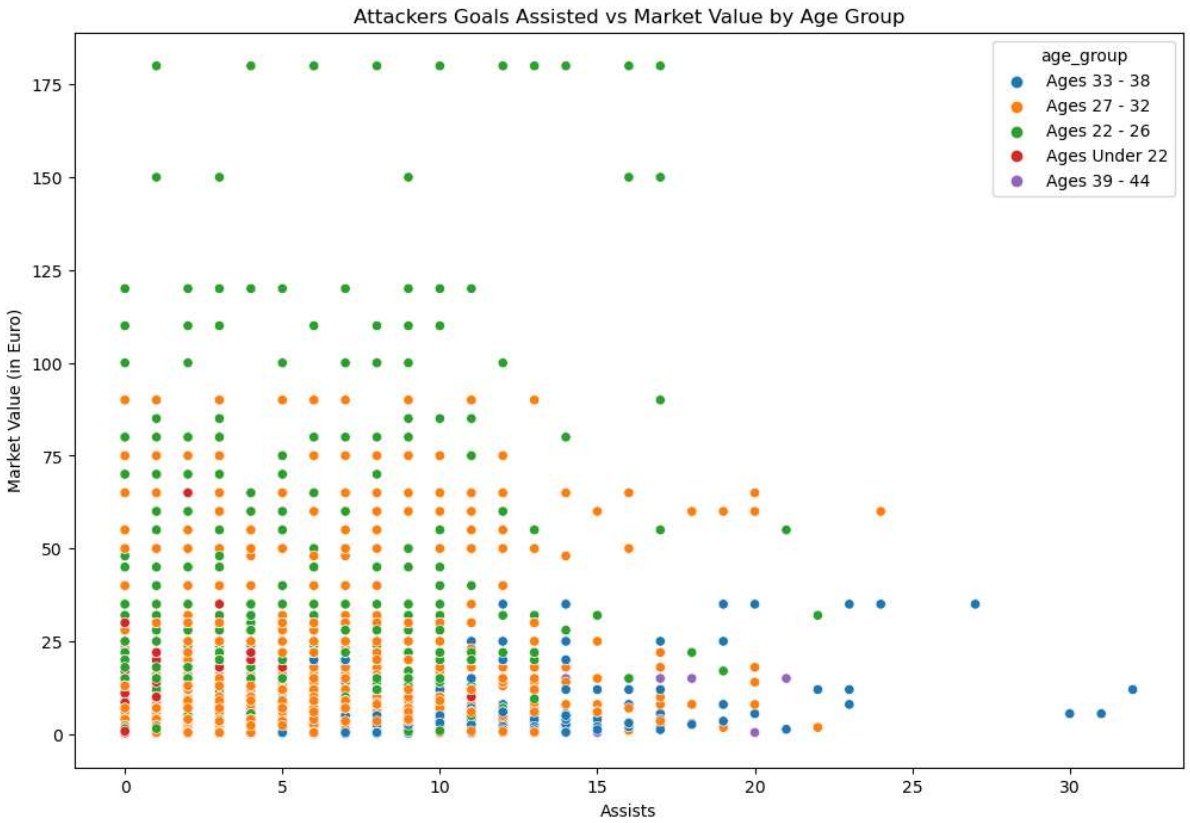
Analysis of Market Value by Goals Assisted in a Year

Individual Player Values

Attackers

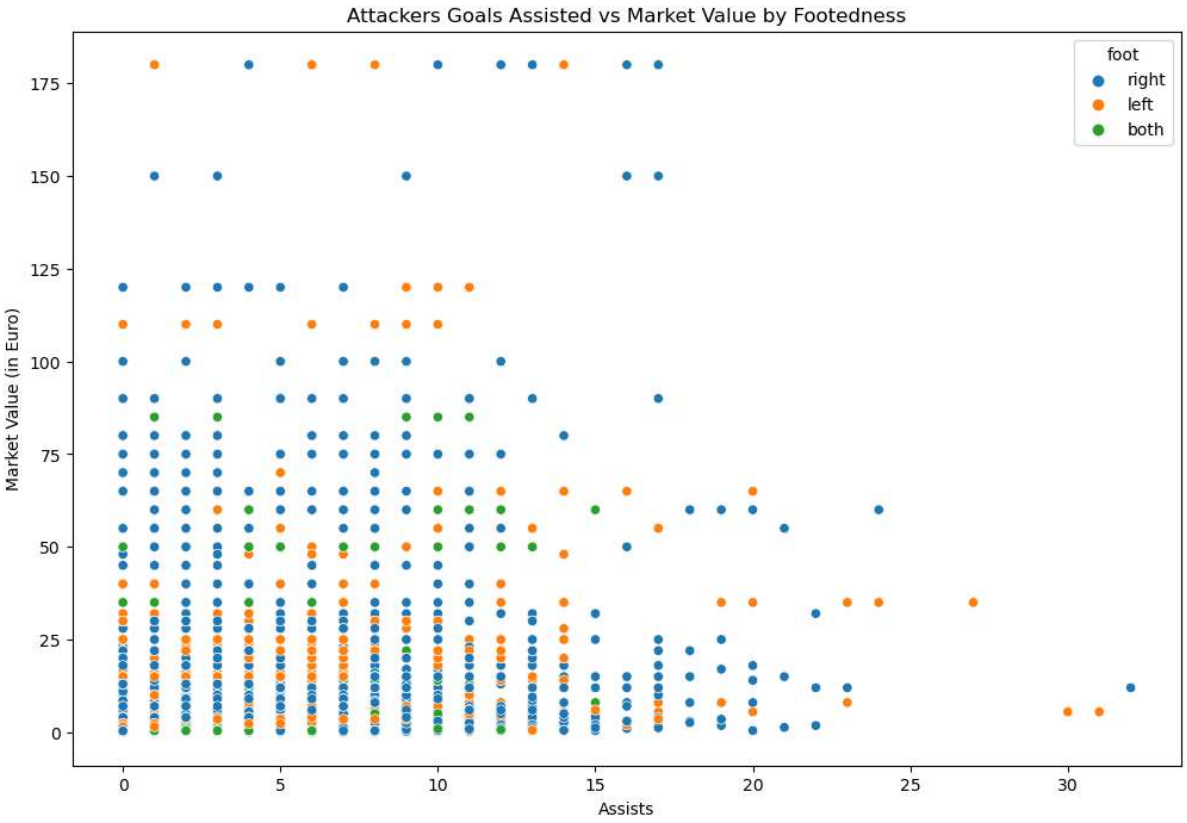
In [203…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='assists', y= attacker_grouped['market_value_in_eur']/1000000, hu
plt.title('Attackers Goals Assisted vs Market Value by Age Group')
plt.xlabel('Assists')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



In [204…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='assists', y= attacker_grouped['market_value_in_eur']/1000000, hu
plt.title('Attackers Goals Assisted vs Market Value by Footedness')
plt.xlabel('Assists')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



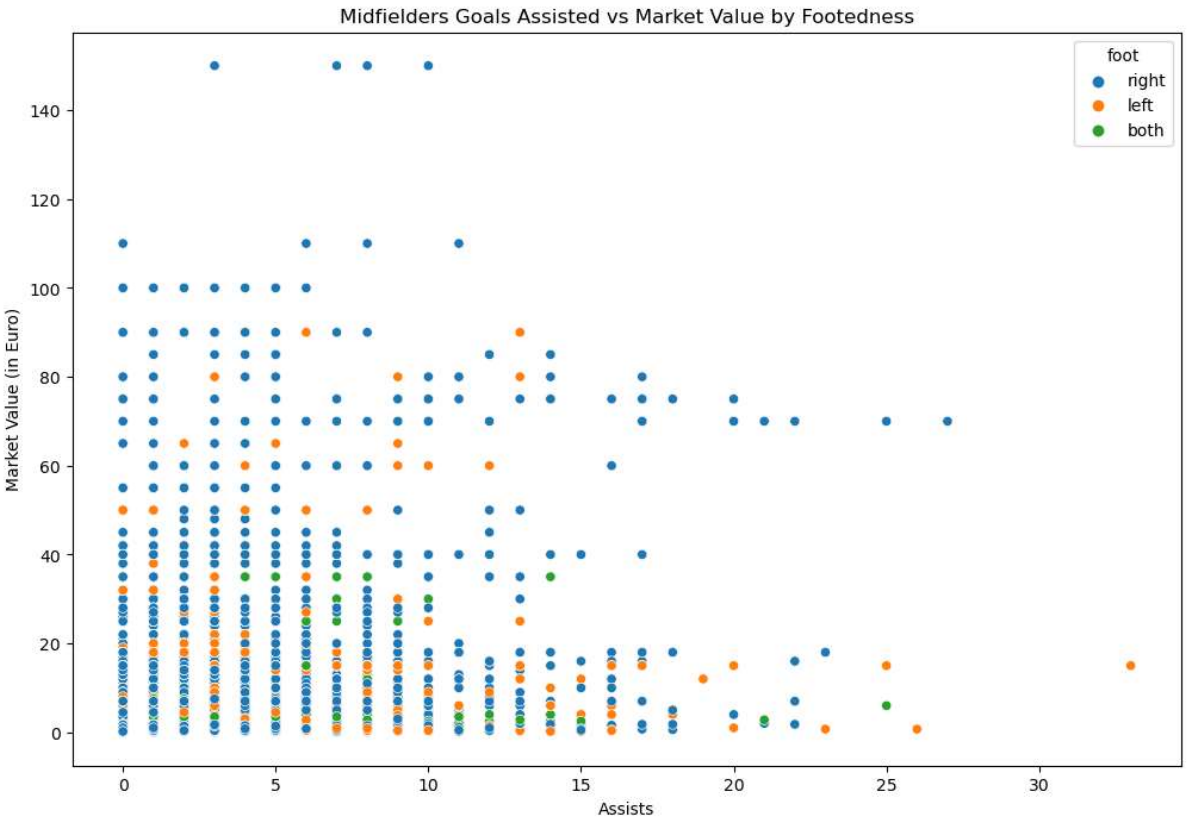
Midfielders

In [205…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='assists', y= midfielder_grouped['market_value_in_eur']/1000000,
plt.title('Midfielders Goals Assisted vs Market Value by Age Group')
plt.xlabel('Assists')
plt.ylabel('Market Value (in Euro)')
plt.show()
```

In [206…

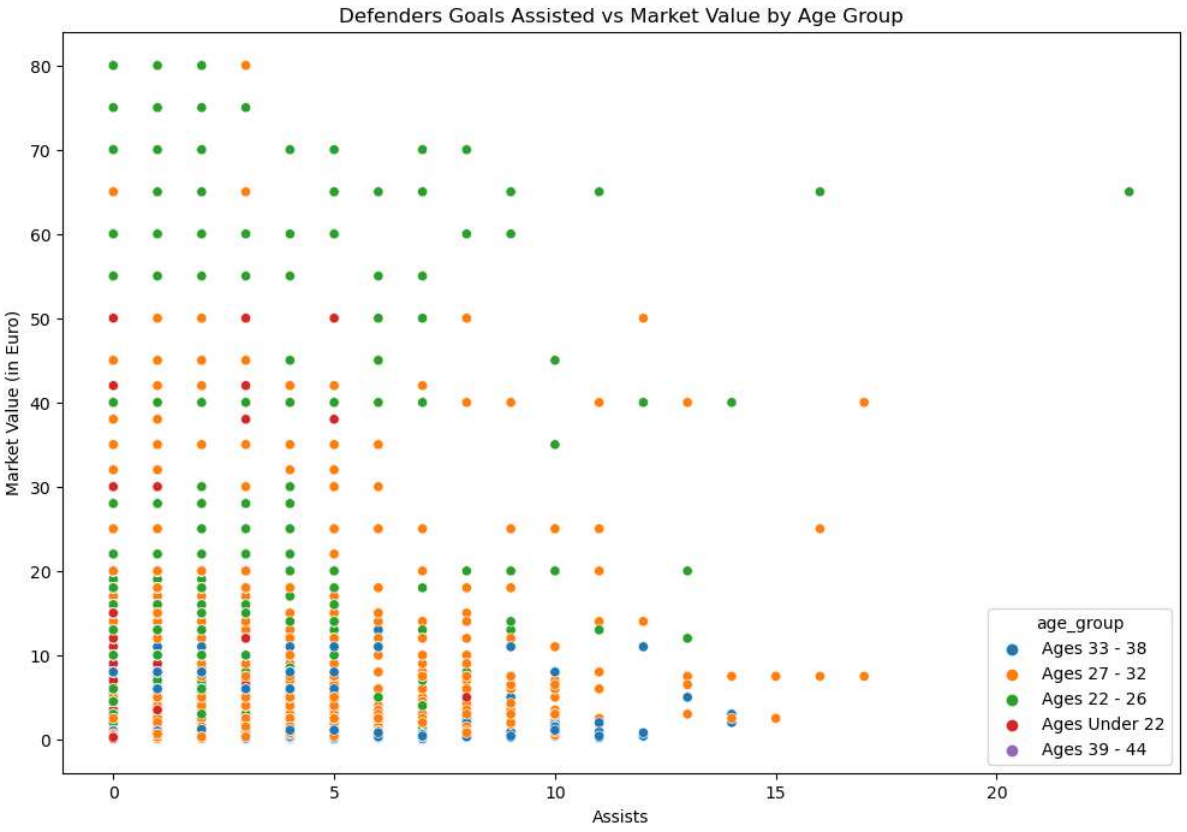
```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='assists', y= midfielder_grouped['market_value_in_eur']/1000000,
plt.title('Midfielders Goals Assisted vs Market Value by Footedness')
plt.xlabel('Assists')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



Defenders

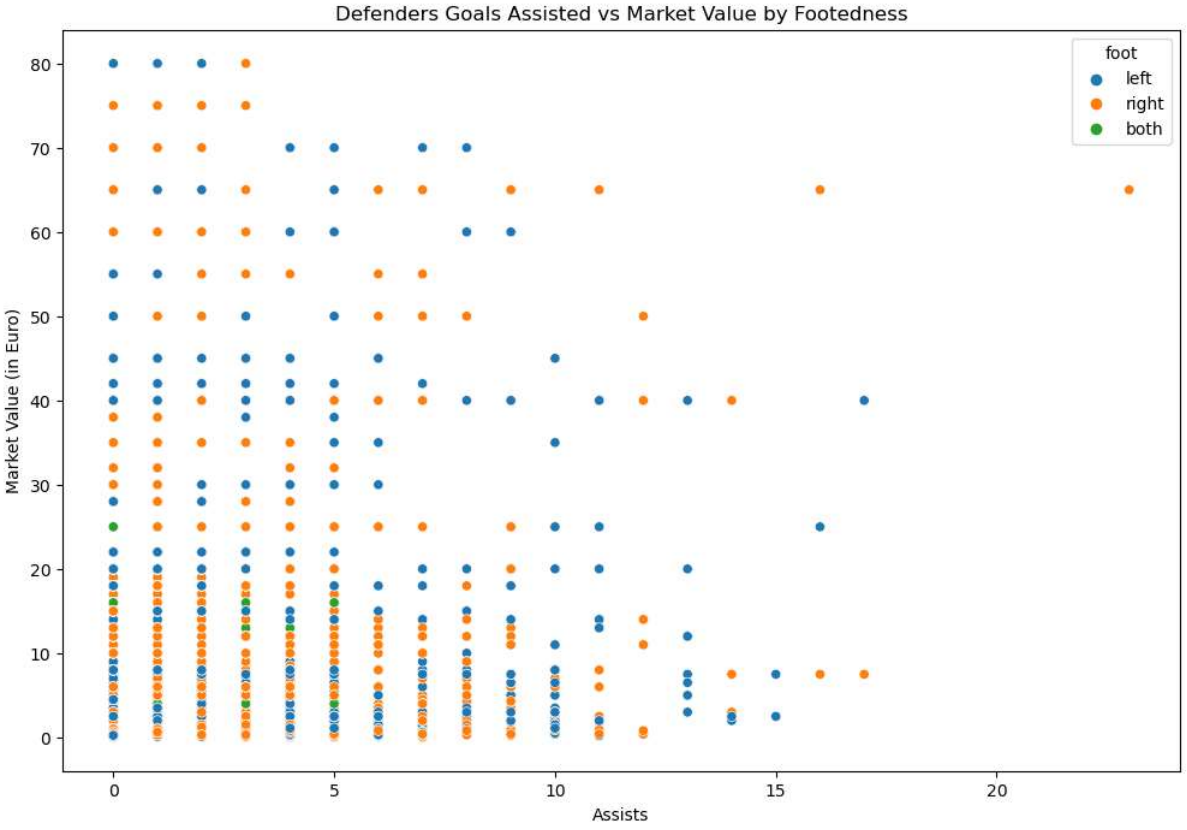
In [207…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='assists', y= defender_grouped['market_value_in_eur']/1000000, hu
plt.title('Defenders Goals Assisted vs Market Value by Age Group')
plt.xlabel('Assists')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



In [208…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='assists', y= defender_grouped['market_value_in_eur']/1000000, hu
plt.title('Defenders Goals Assisted vs Market Value by Footedness')
plt.xlabel('Assists')
plt.ylabel('Market Value (in Euro)')
plt.show()
```

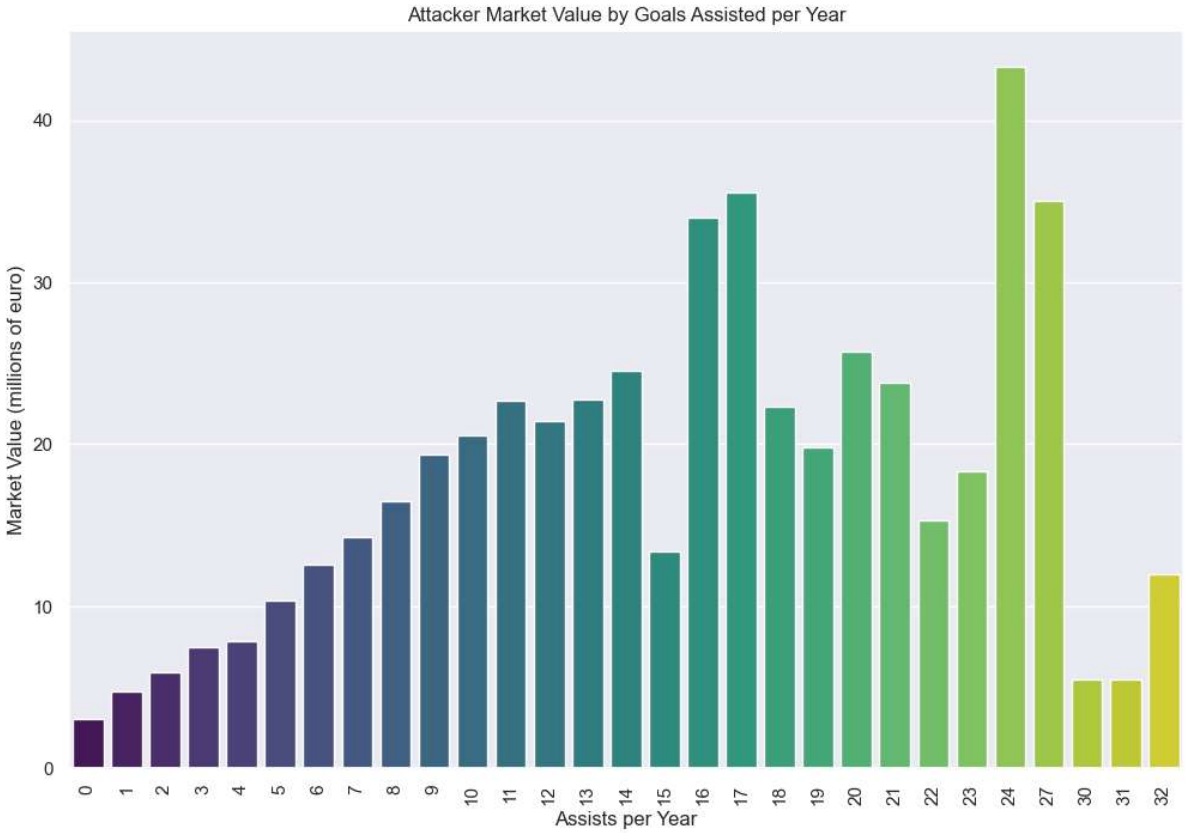


Average Player Values

Attackers

In [621…

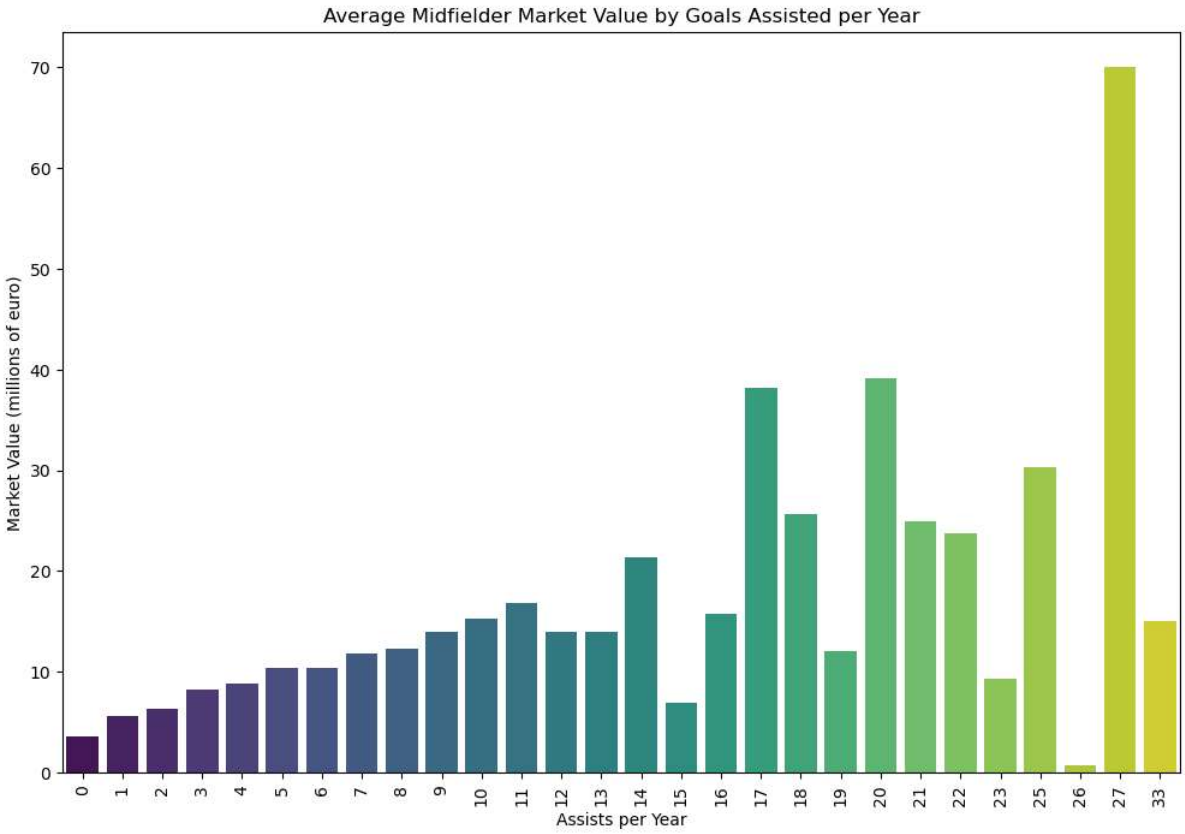
```
plt.figure(figsize=(12, 8))
sns.barplot(x = 'assists' , y=attacker_grouped['market_value_in_eur']/1000000, data
plt.title("Attacker Market Value by Goals Assisted per Year")
plt.ylabel('Market Value (millions of euro)')
plt.xlabel('Assists per Year')
plt.xticks(rotation = 'vertical')
plt.show()
```



Midfielders

In [197…

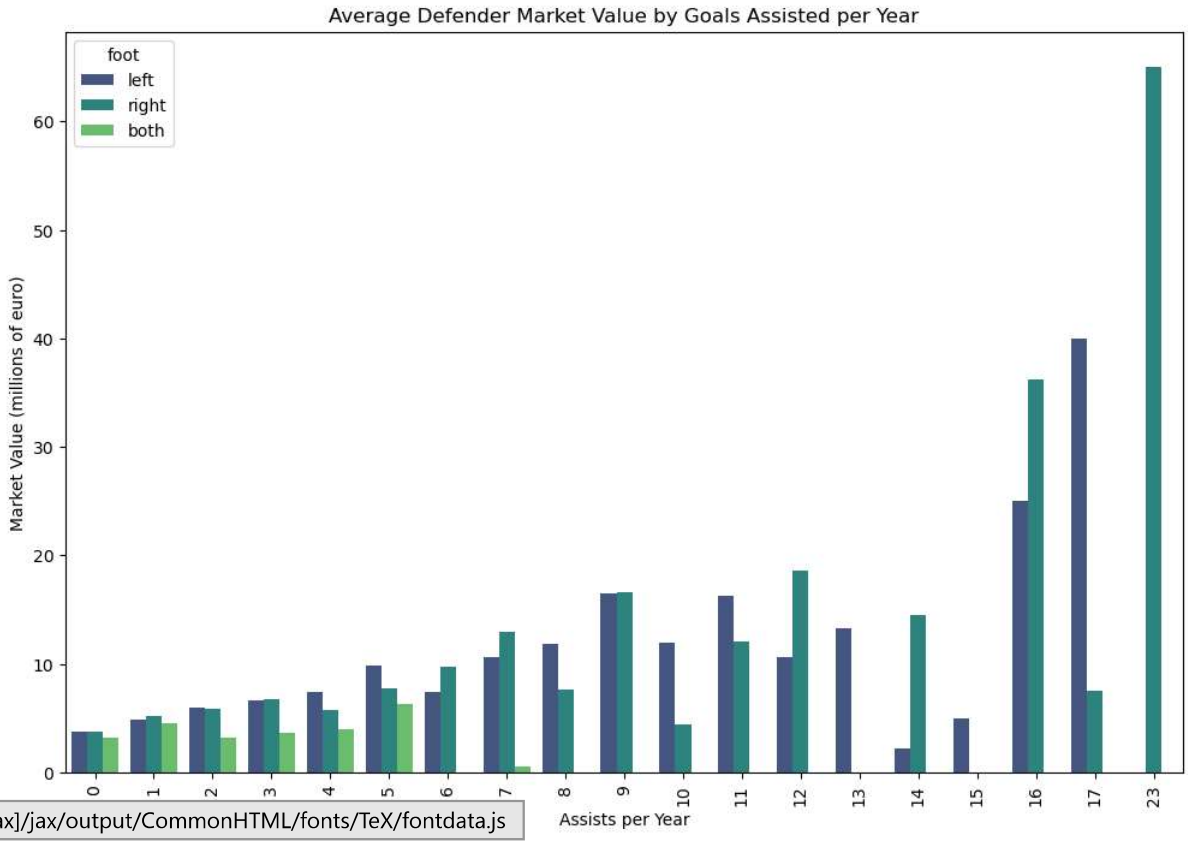
```
plt.figure(figsize=(12, 8))
sns.barplot(x = 'assists' , y=midfielder_grouped['market_value_in_eur']/1000000, da
plt.title("Average Midfielder Market Value by Goals Assisted per Year")
plt.ylabel('Market Value (millions of euro)')
plt.xlabel('Assists per Year')
plt.xticks(rotation = 'vertical')
plt.show()
```



Defenders

In [184…

```
plt.figure(figsize=(12, 8))
sns.barplot(x = 'assists' , y=defender_grouped['market_value_in_eur']/1000000, data
plt.title("Average Defender Market Value by Goals Assisted per Year")
plt.ylabel('Market Value (millions of euro)')
plt.xlabel('Assists per Year')
plt.xticks(rotation = 'vertical')
plt.show()
```



Analysis of last season 2022-2023

In [210…
```
merged_players_df['goals_2022'] = merged_players_df['goals_2022'].fillna(0).astype(
```

In [211…
```
merged_players_df['assists_2022'] = merged_players_df['assists_2022'].fillna(0).ast
```

In [292…
```
merged_players_df['red_cards_2022'] = merged_players_df['red_cards_2022'].fillna(0)
```

In [293…
```
merged_players_df['yellow_cards_2022'] = merged_players_df['yellow_cards_2022'].fil
```

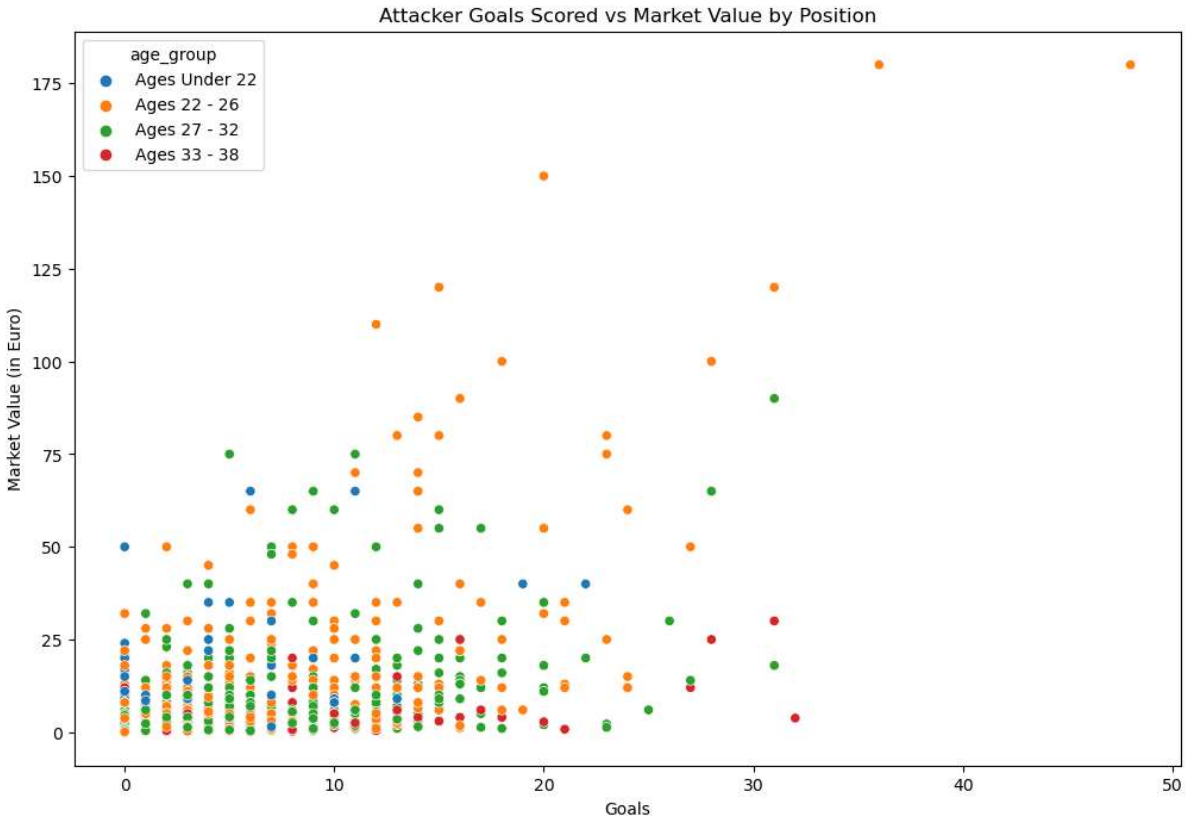
In [294…
```
merged_players_df['clean_sheet_2022'] = merged_players_df['clean_sheet_2022'].fillr
```

In [216…
```
attacker_grouped_22 = merged_players_df[merged_players_df['position'] == 'Attack']
midfielder_grouped_22 = merged_players_df[merged_players_df['position'] == 'Midfiel
defender_grouped_22 = merged_players_df[merged_players_df['position'] == 'Defender'
goalkeeper_grouped_22 = merged_players_df[merged_players_df['position'] == 'Goalkee
```

Goals Scored 22/23

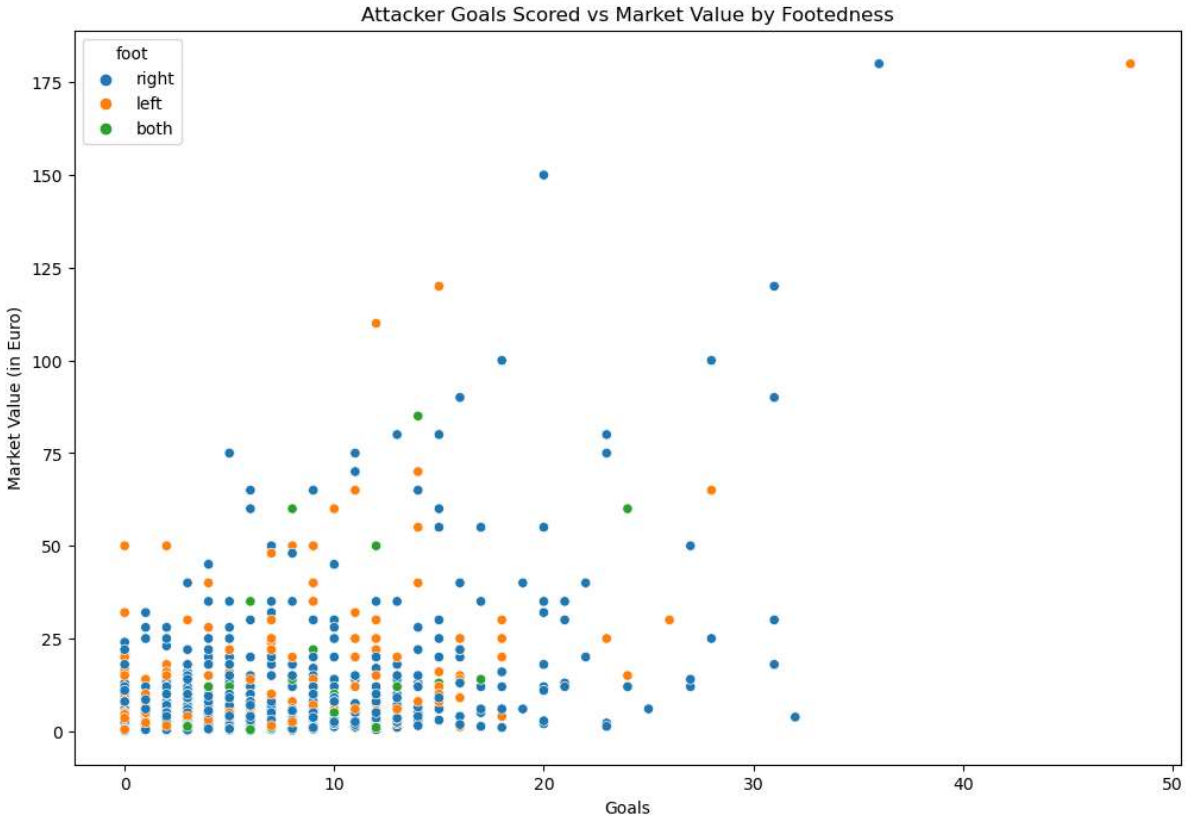
Attackers

In [219…
```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='goals_2022', y= attacker_grouped_22['market_value_in_eur']/10000
plt.title('Attacker Goals Scored vs Market Value by Position')
plt.xlabel('Goals')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



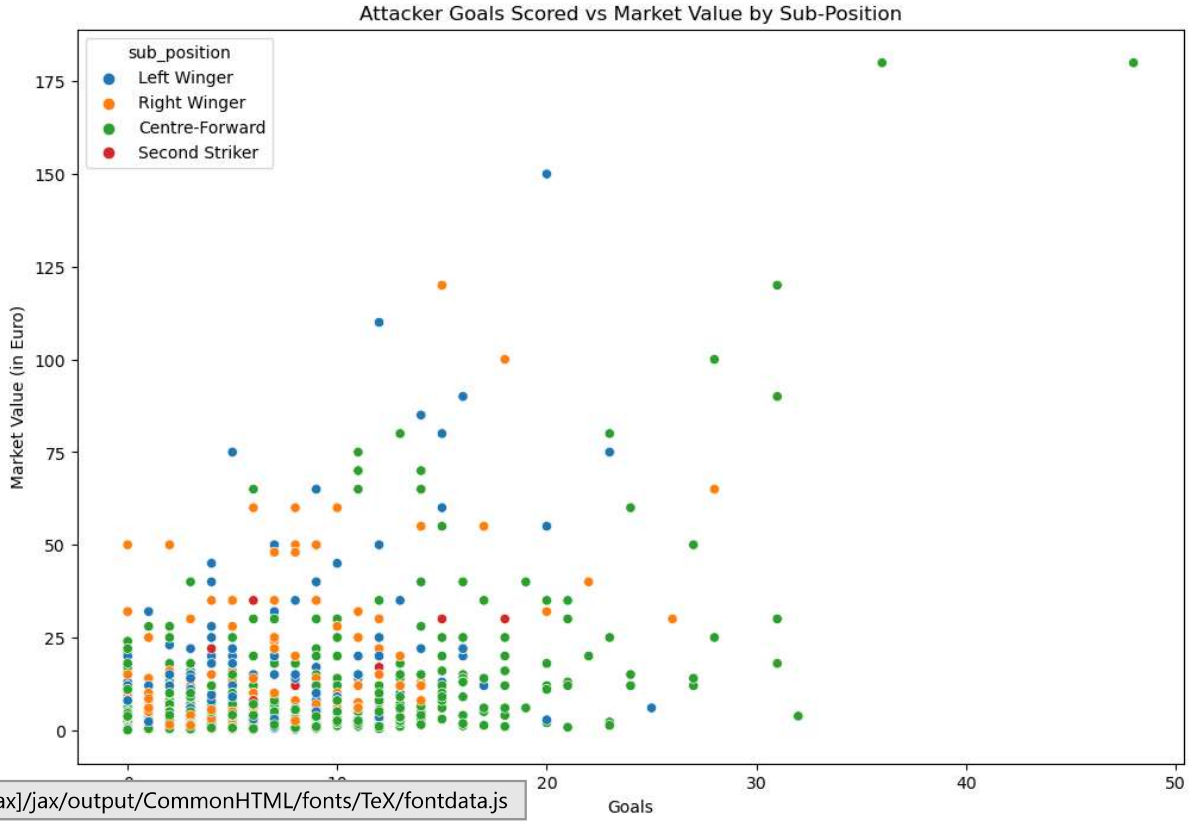

```
sns.scatterplot(x='goals_2022', y=attacker_grouped_22['market_value_in_eur']/100000
```

```
plt.title('Attacker Goals Scored vs Market Value by Footedness')
plt.xlabel('Goals')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



In [222…

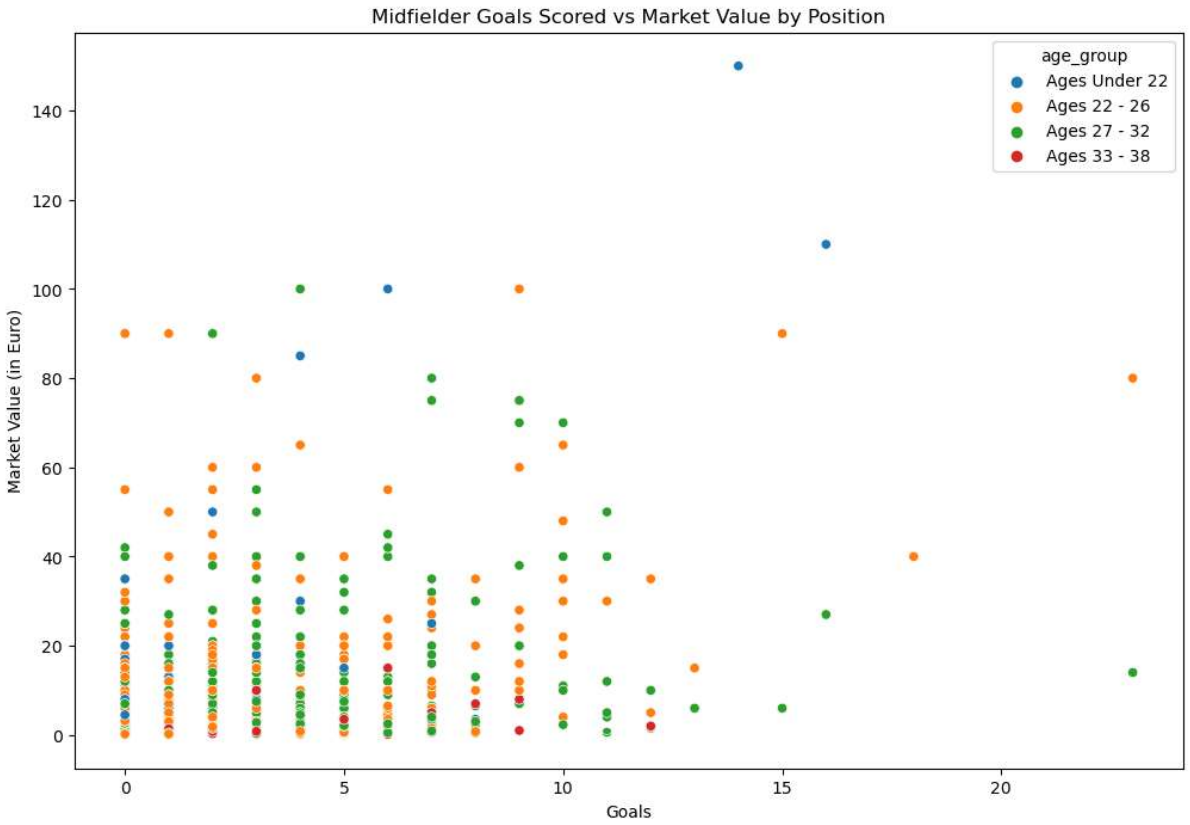
```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='goals_2022', y=attacker_grouped_22['market_value_in_eur']/100000
plt.title('Attacker Goals Scored vs Market Value by Sub-Position')
plt.xlabel('Goals')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



Midfielders

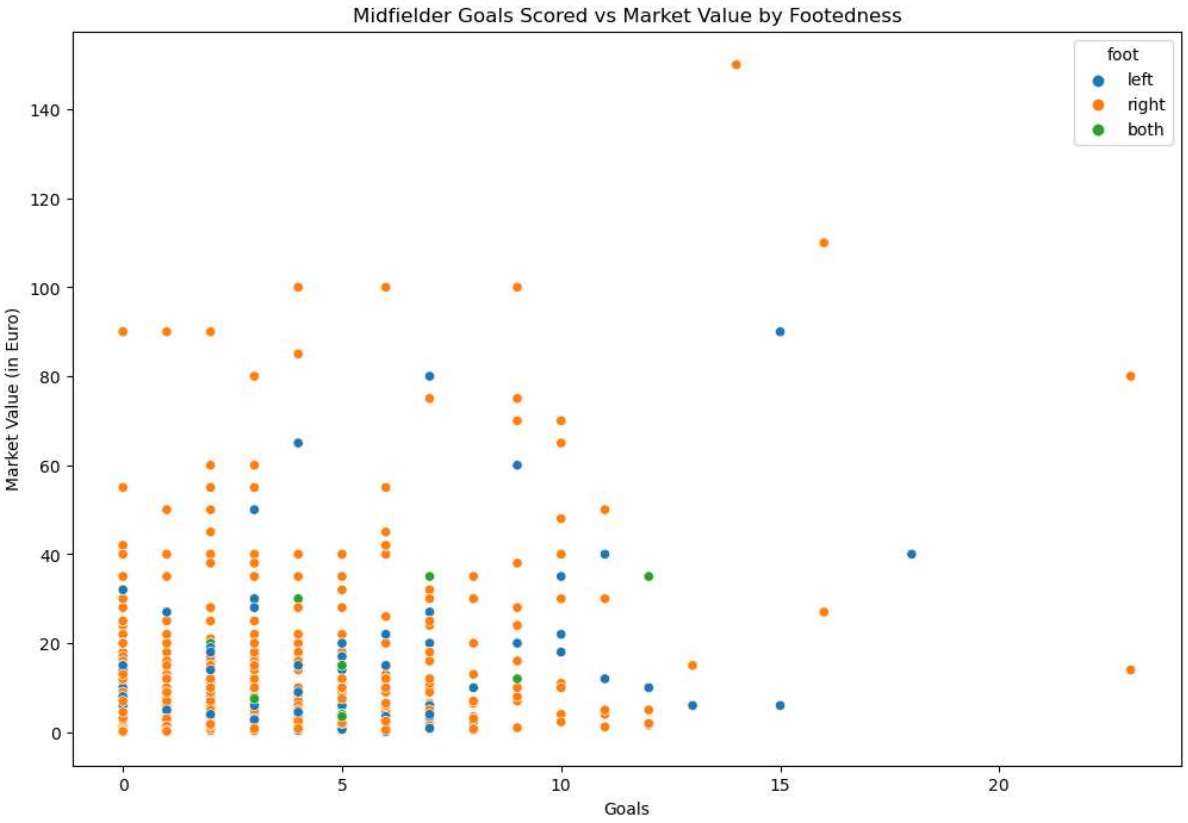
In [228…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='goals_2022', y= midfielder_grouped_22['market_value_in_eur']/100
plt.title('Midfielder Goals Scored vs Market Value by Position')
plt.xlabel('Goals')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



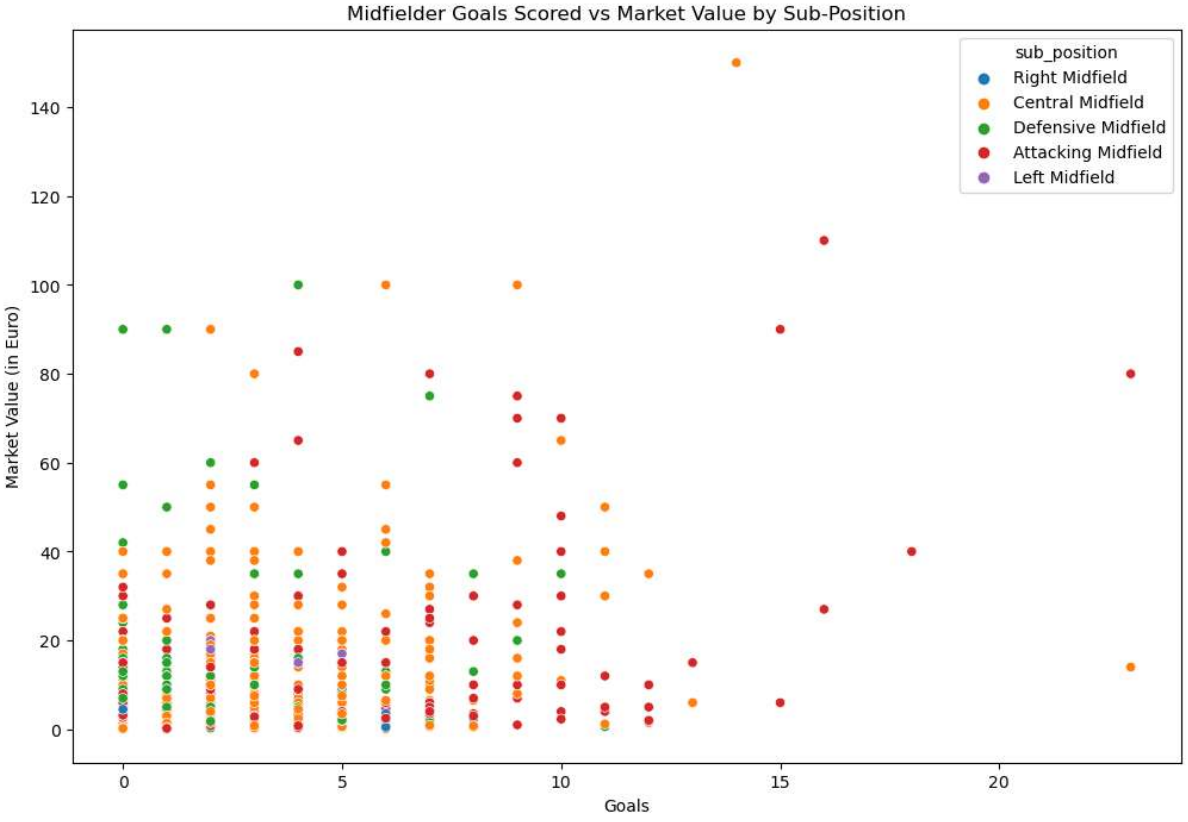
In [227…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='goals_2022', y=midfielder_grouped_22['market_value_in_eur']/1000
plt.title('Midfielder Goals Scored vs Market Value by Footedness')
plt.xlabel('Goals')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



In [226…

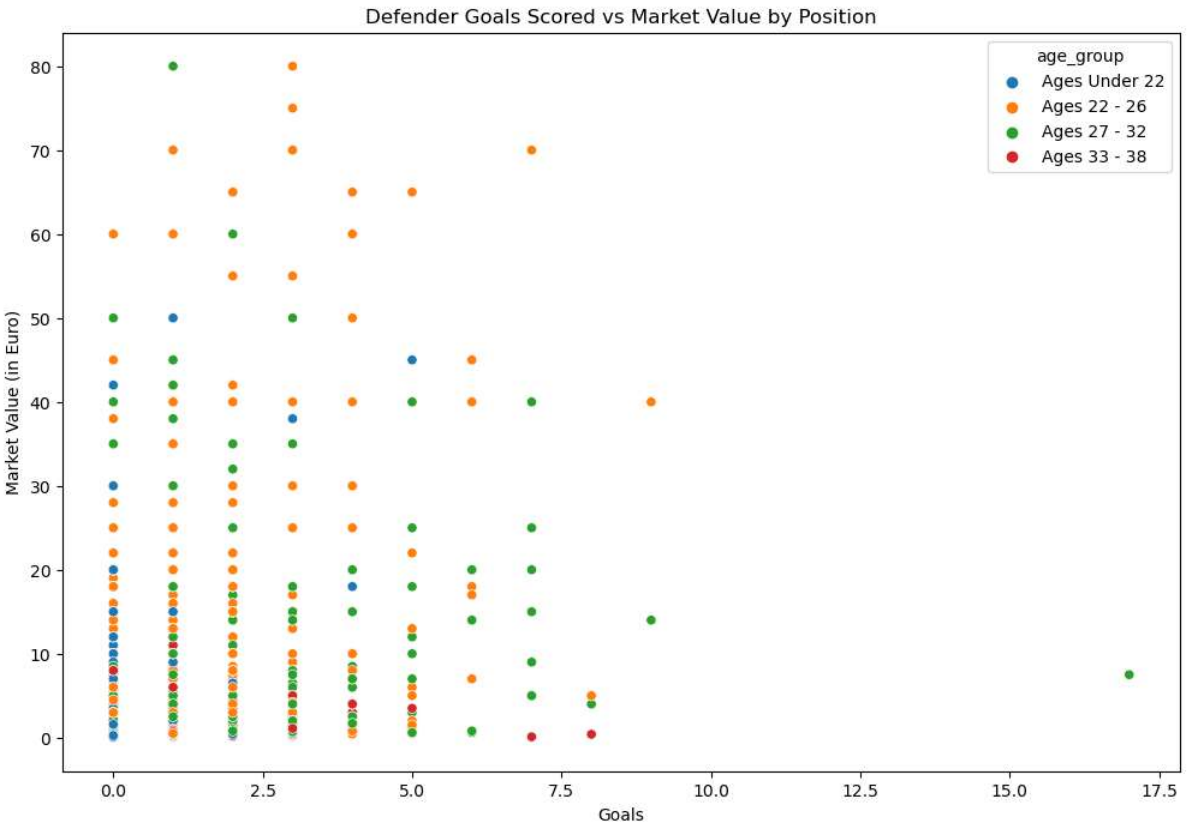
```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='goals_2022', y=midfielder_grouped_22['market_value_in_eur']/1000
plt.title('Midfielder Goals Scored vs Market Value by Sub-Position')
plt.xlabel('Goals')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



Defenders

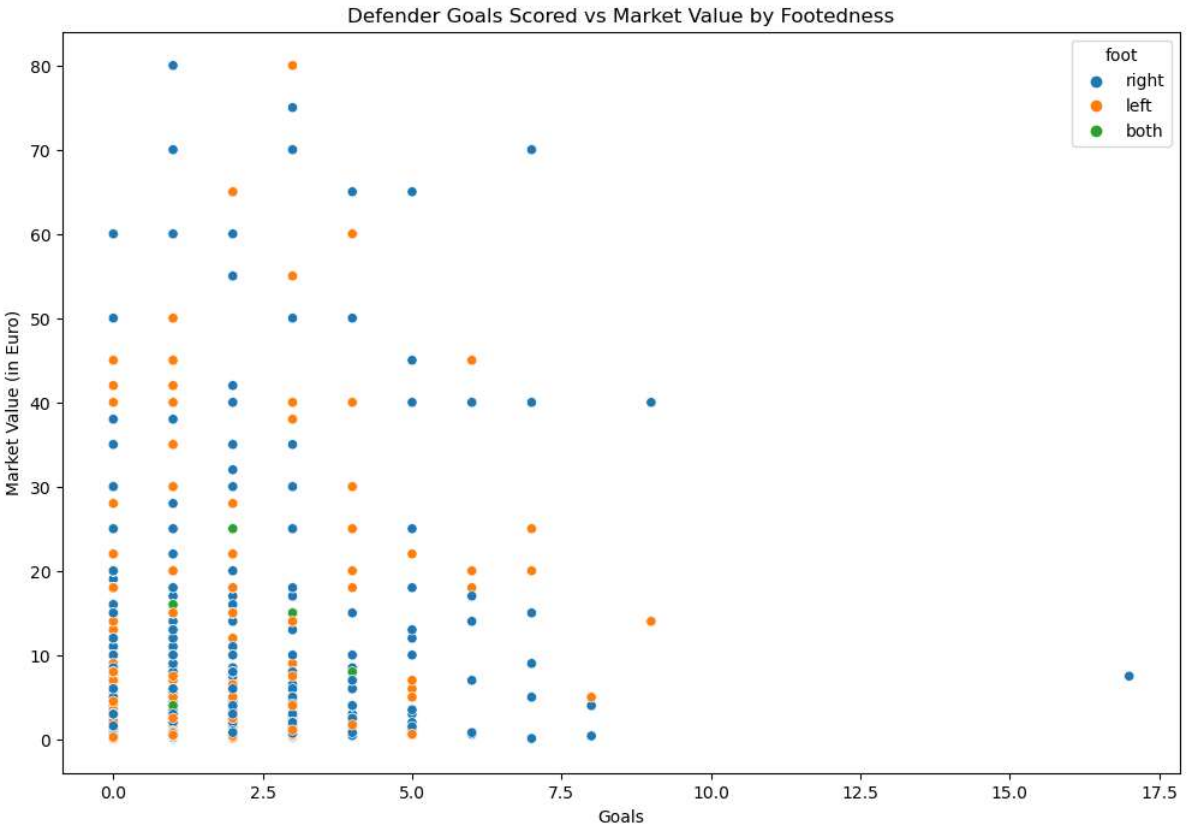
In [229…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='goals_2022', y= defender_grouped_22['market_value_in_eur']/10000
plt.title('Defender Goals Scored vs Market Value by Position')
plt.xlabel('Goals')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



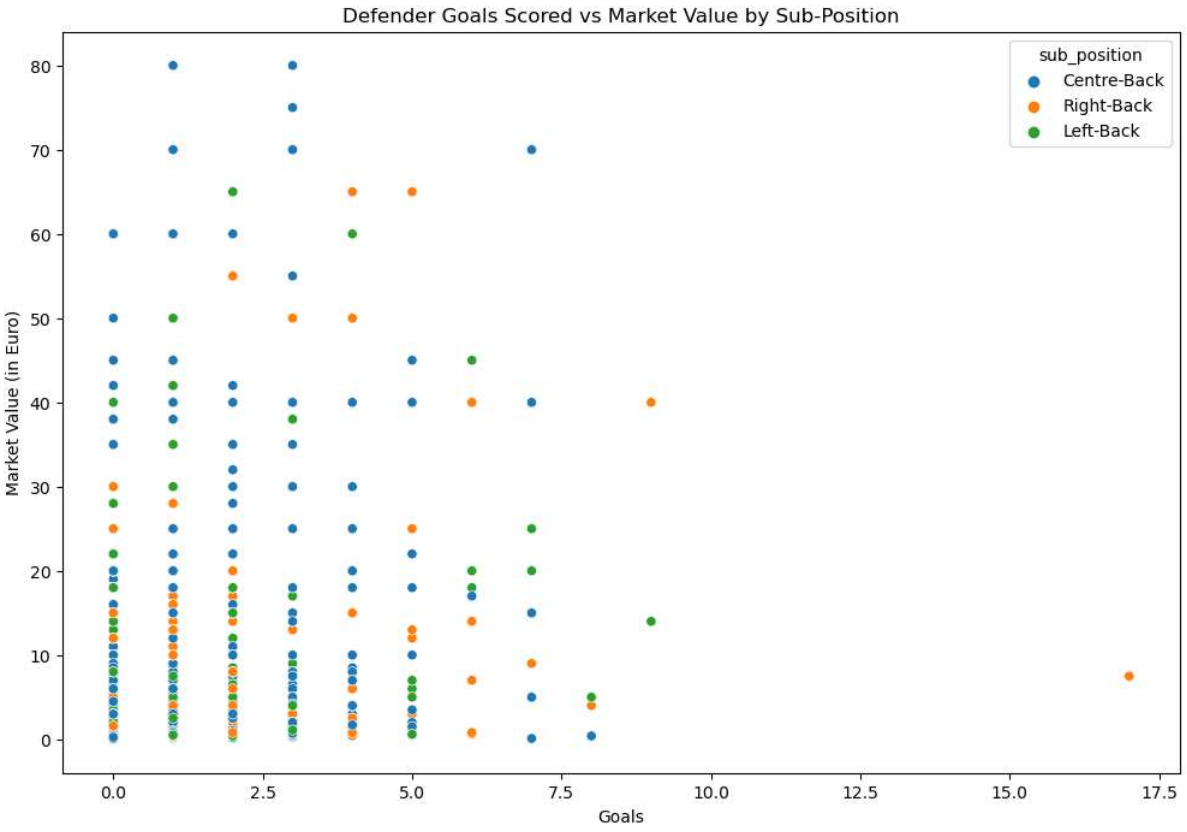
In [230…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='goals_2022', y=defender_grouped_22['market_value_in_eur']/100000
plt.title('Defender Goals Scored vs Market Value by Footedness')
plt.xlabel('Goals')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



In [231…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='goals_2022', y=defender_grouped_22['market_value_in_eur']/100000
plt.title('Defender Goals Scored vs Market Value by Sub-Position')
plt.xlabel('Goals')
plt.ylabel('Market Value (in Euro)')
plt.show()
```

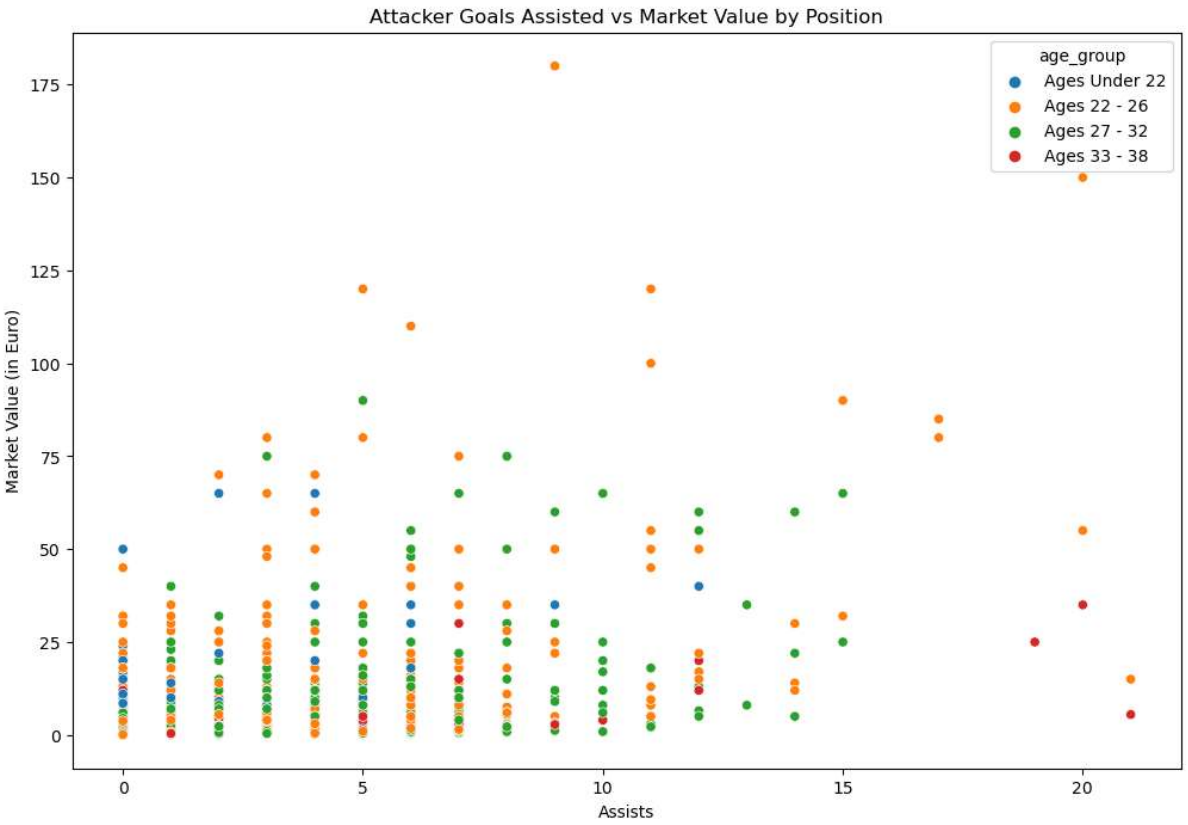


Goals Assisted 22/23

Attackers

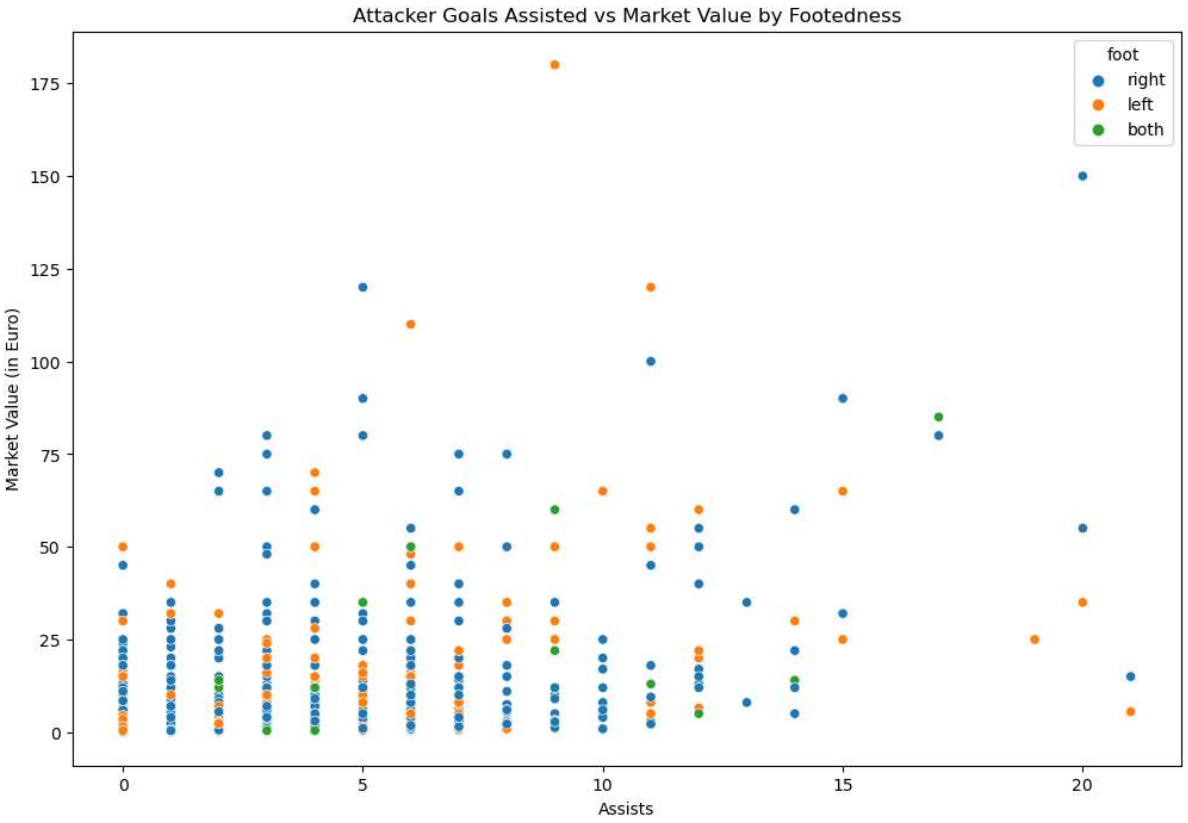
In [233…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='assists_2022', y= attacker_grouped_22['market_value_in_eur']/100
plt.title('Attacker Goals Assisted vs Market Value by Position')
plt.xlabel('Assists')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



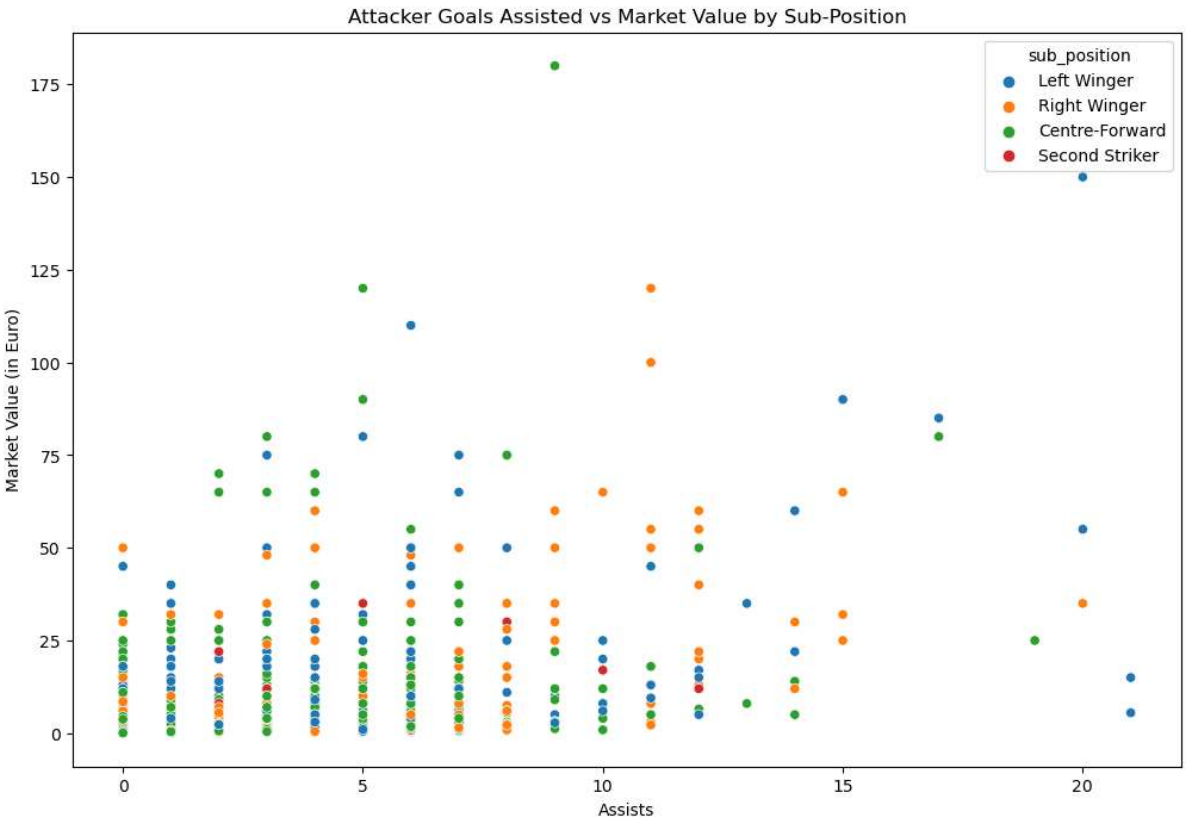
In [234…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='assists_2022', y=attacker_grouped_22['market_value_in_eur']/1000
plt.title('Attacker Goals Assisted vs Market Value by Footedness')
plt.xlabel('Assists')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



In [235…

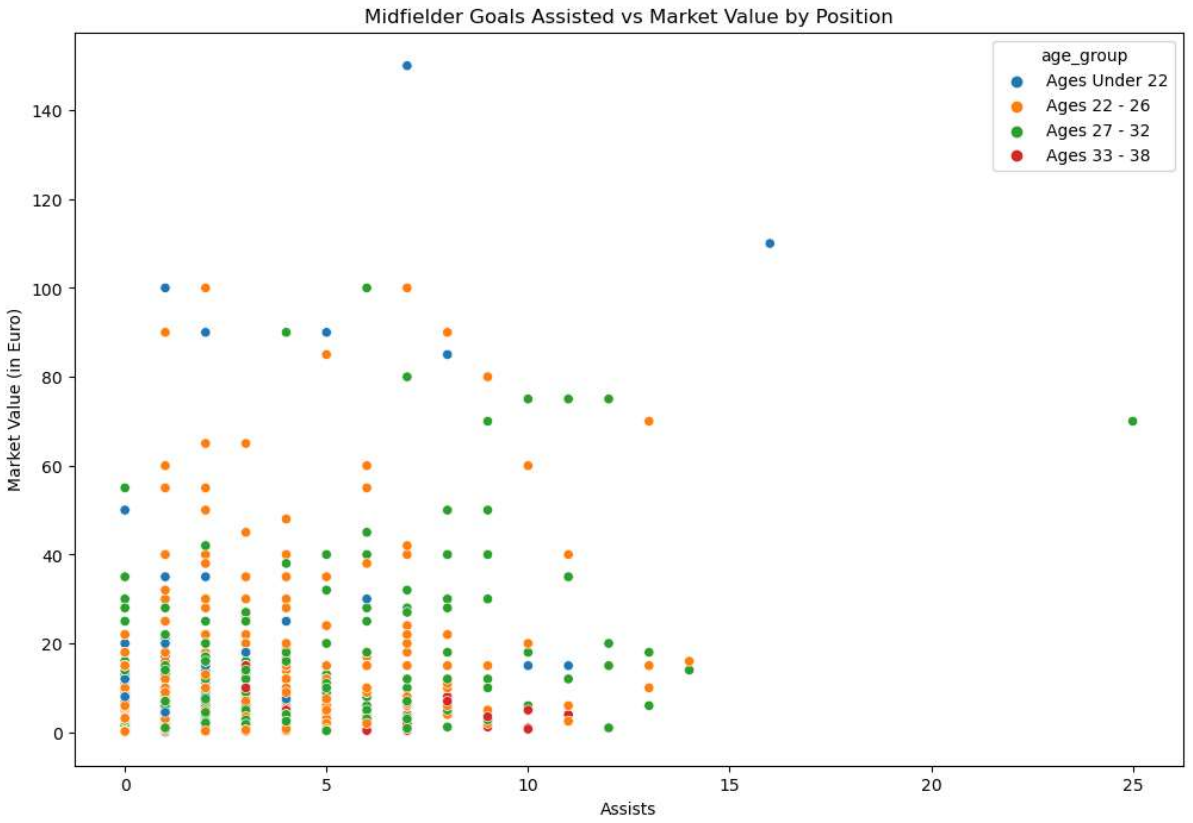
```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='assists_2022', y=attacker_grouped_22['market_value_in_eur']/1000
plt.title('Attacker Goals Assisted vs Market Value by Sub-Position')
plt.xlabel('Assists')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



Midfielders

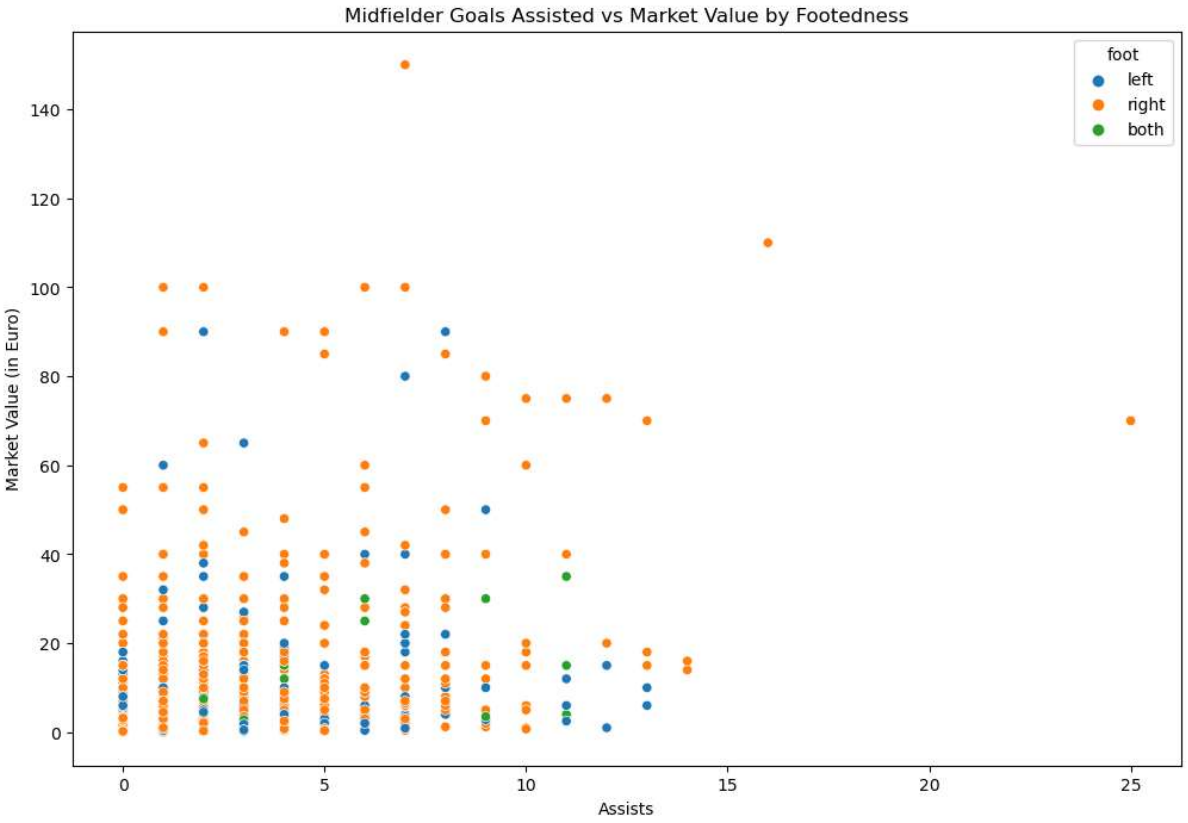
In [236…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='assists_2022', y= midfielder_grouped_22['market_value_in_eur']/1
plt.title('Midfielder Goals Assisted vs Market Value by Position')
plt.xlabel('Assists')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



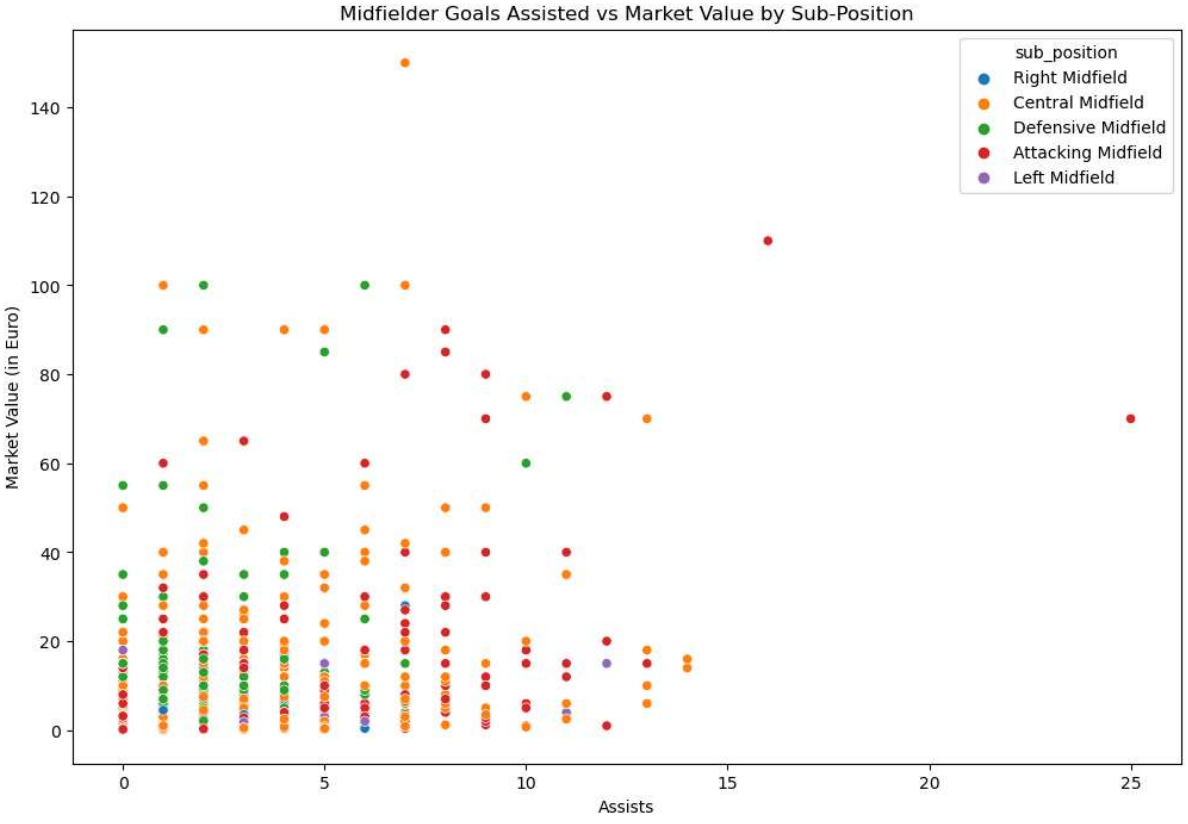
In [237…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='assists_2022', y=midfielder_grouped_22['market_value_in_eur']/10
plt.title('Midfielder Goals Assisted vs Market Value by Footedness')
plt.xlabel('Assists')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



In [238…

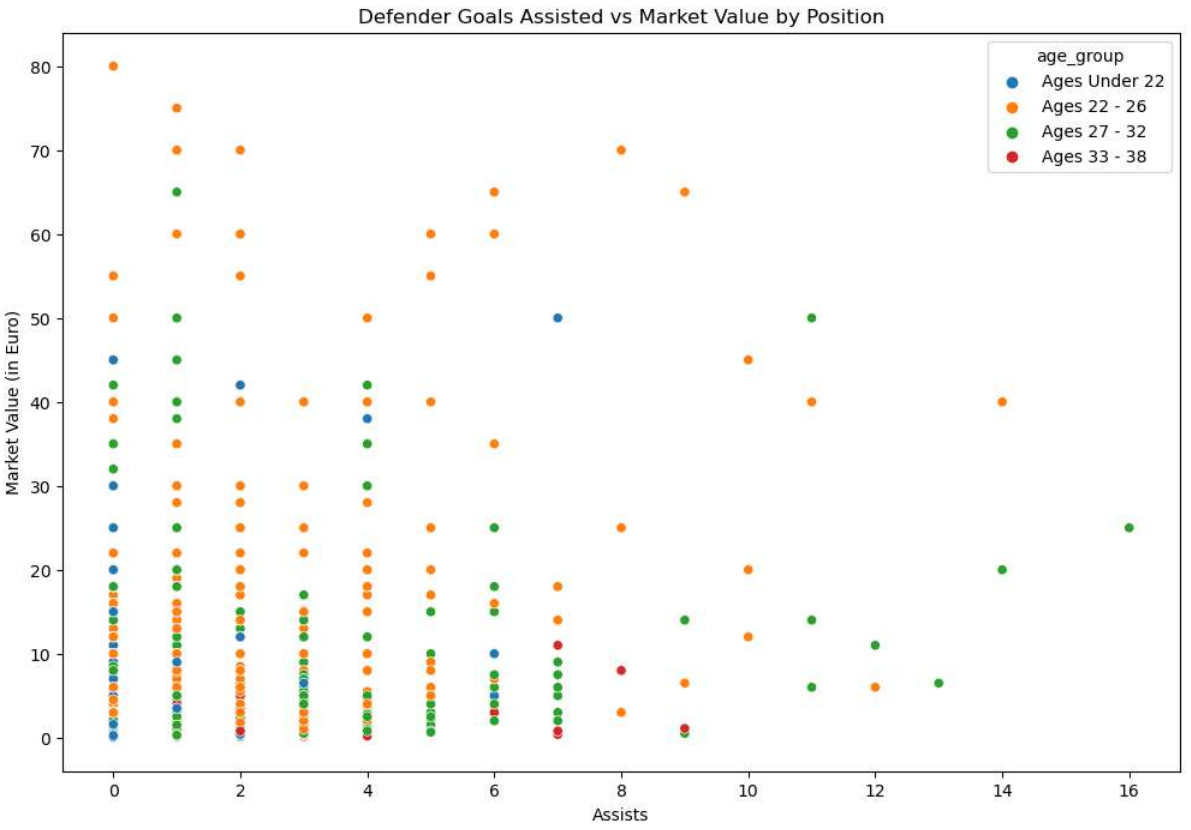
```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='assists_2022', y=midfielder_grouped_22['market_value_in_eur']/10
plt.title('Midfielder Goals Assisted vs Market Value by Sub-Position')
plt.xlabel('Assists')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



Defenders

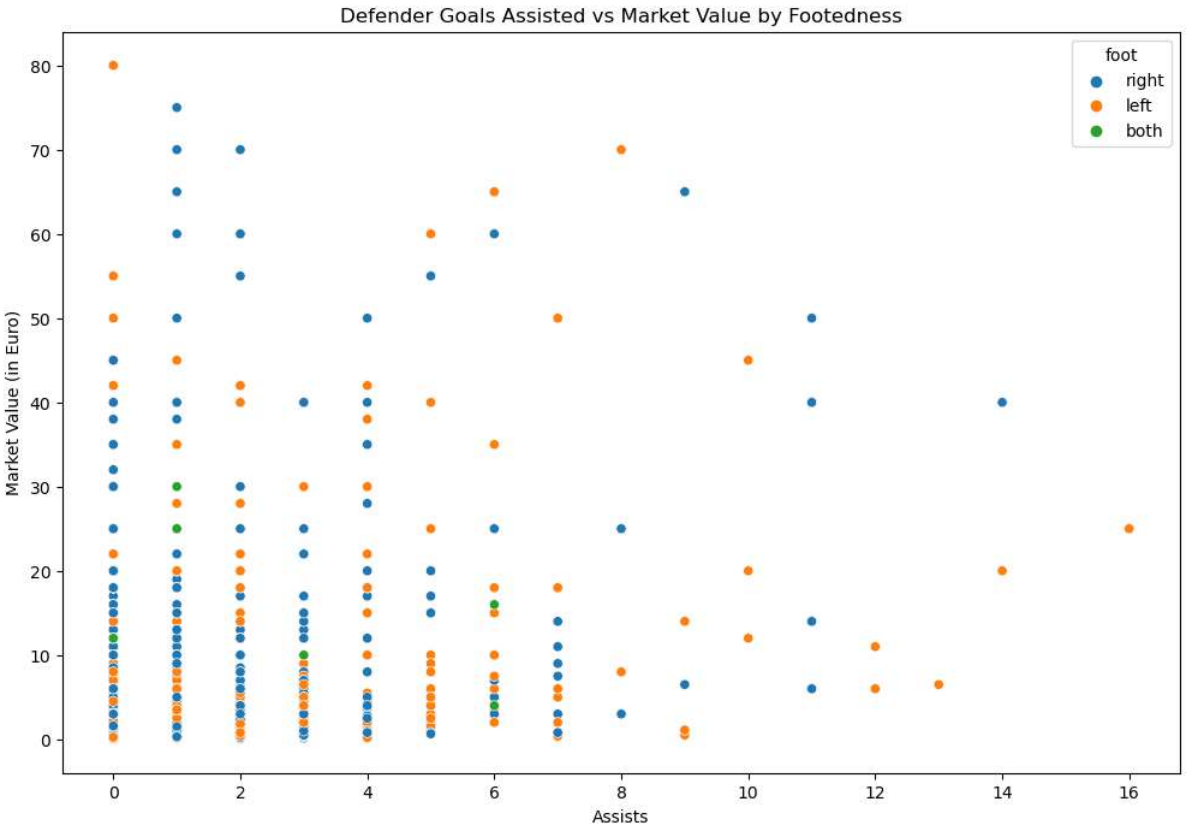
In [239…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='assists_2022', y= defender_grouped_22['market_value_in_eur']/100
plt.title('Defender Goals Assisted vs Market Value by Position')
plt.xlabel('Assists')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



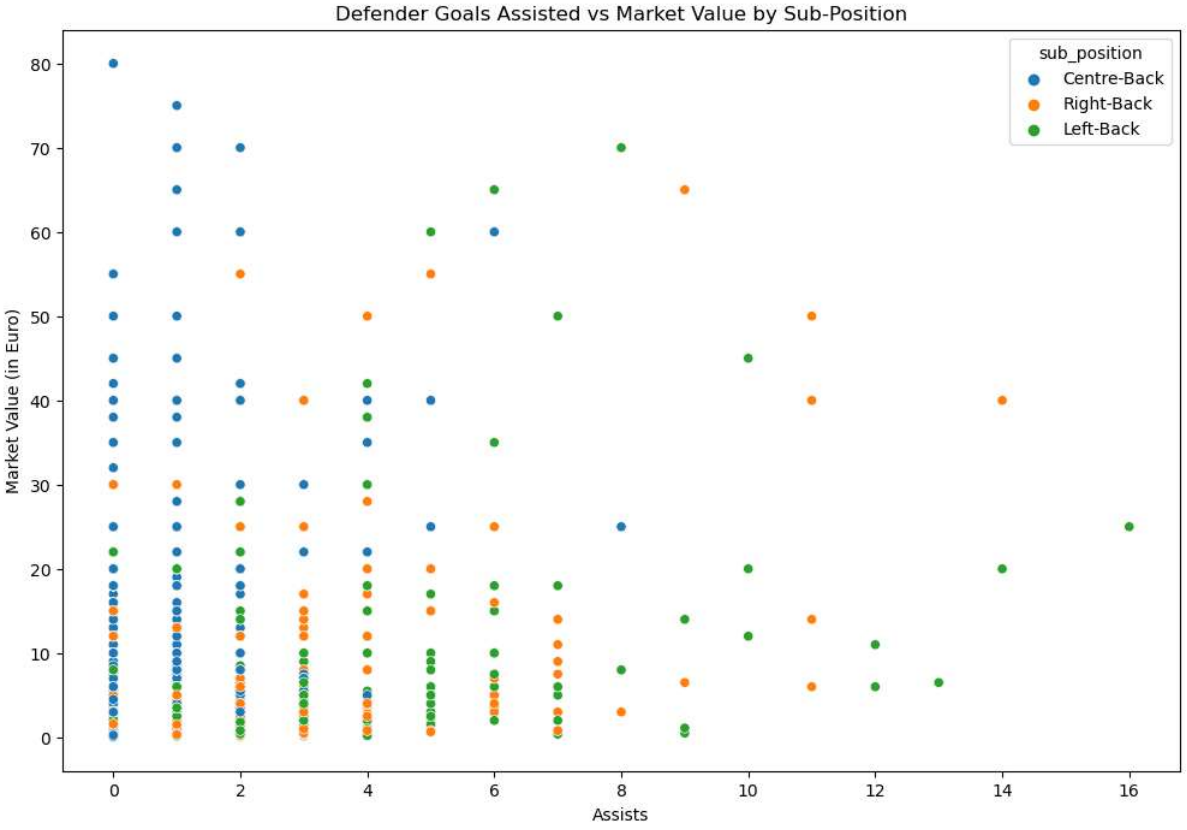
In [240…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='assists_2022', y=defender_grouped_22['market_value_in_eur']/1000
plt.title('Defender Goals Assisted vs Market Value by Footedness')
plt.xlabel('Assists')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



In [241…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='assists_2022', y=defender_grouped_22['market_value_in_eur']/1000
plt.title('Defender Goals Assisted vs Market Value by Sub-Position')
plt.xlabel('Assists')
plt.ylabel('Market Value (in Euro)')
plt.show()
```

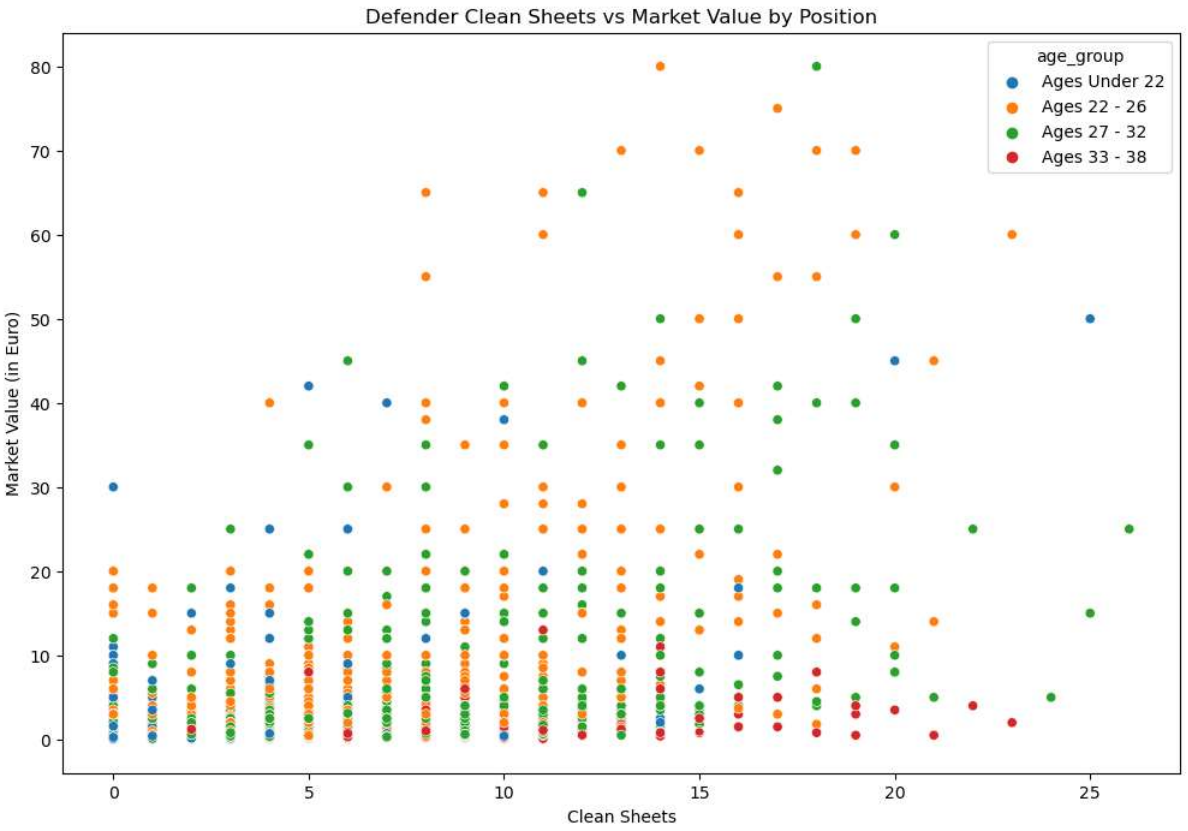


Clean Sheets 22/23

Defenders

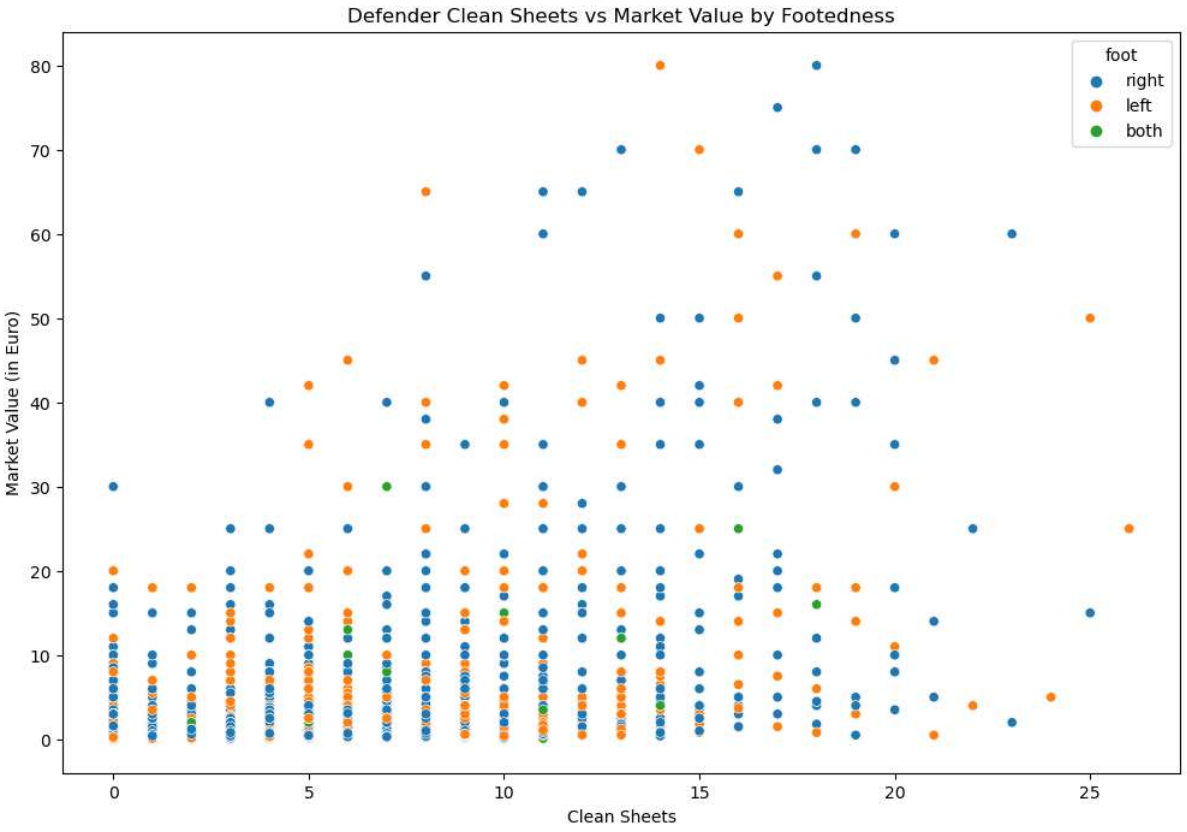
In [242…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='clean_sheet_2022', y= defender_grouped_22['market_value_in_eur']
plt.title('Defender Clean Sheets vs Market Value by Position')
plt.xlabel('Clean Sheets')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



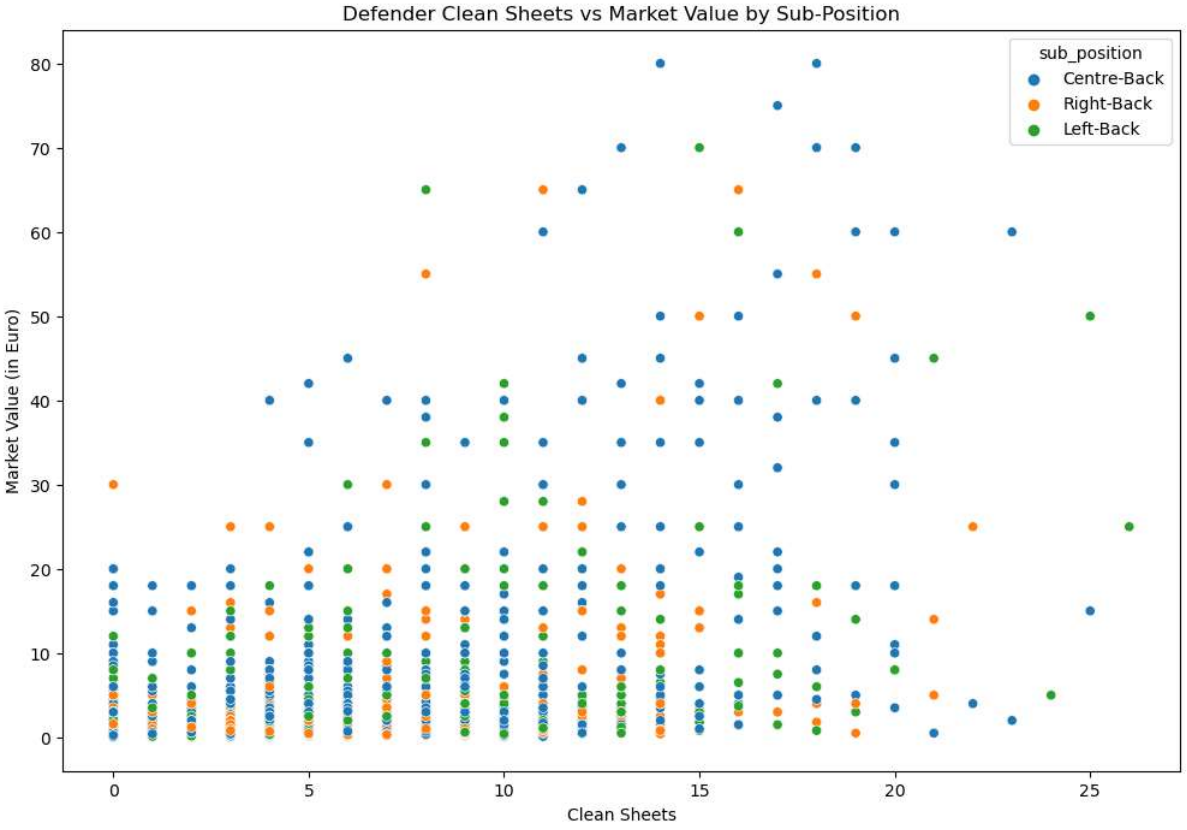
In [243…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='clean_sheet_2022', y=defender_grouped_22['market_value_in_eur']/
plt.title('Defender Clean Sheets vs Market Value by Footedness')
plt.xlabel('Clean Sheets')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



In [244…

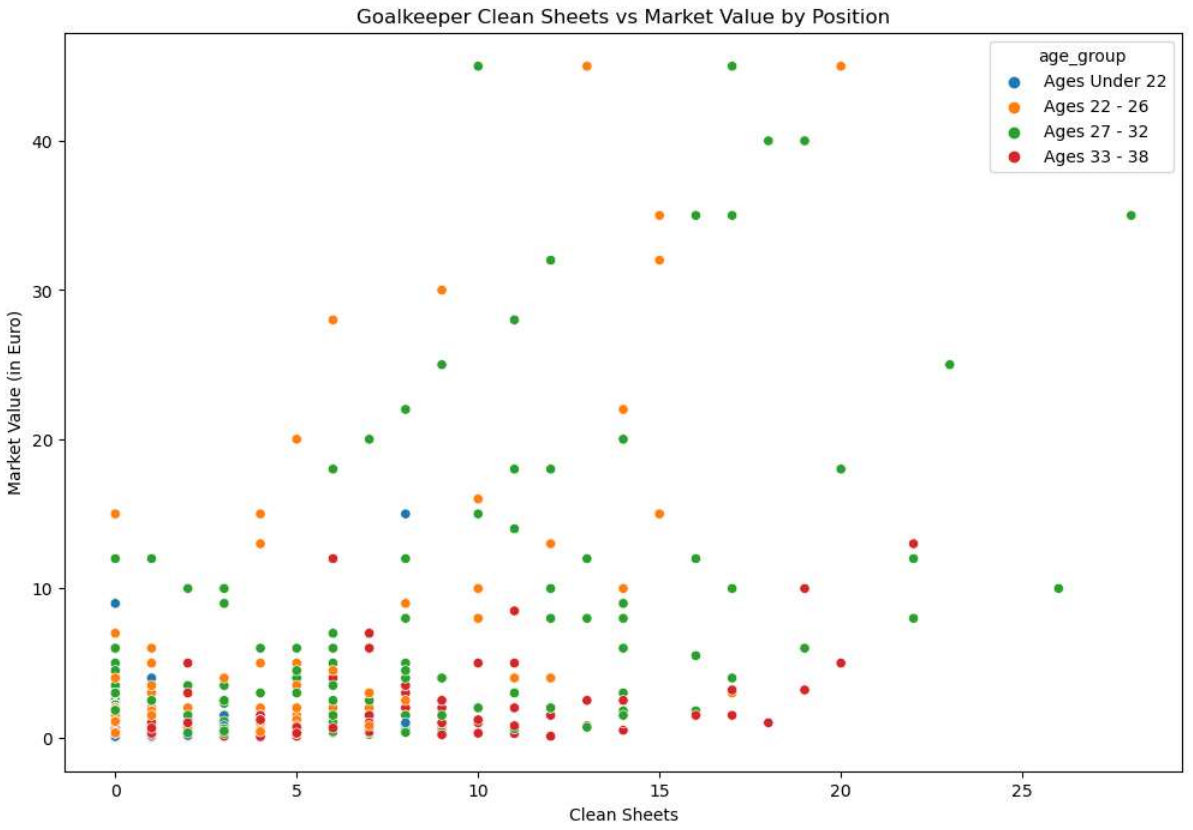
```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='clean_sheet_2022', y=defender_grouped_22['market_value_in_eur']/
plt.title('Defender Clean Sheets vs Market Value by Sub-Position')
plt.xlabel('Clean Sheets')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



Goalkeepers

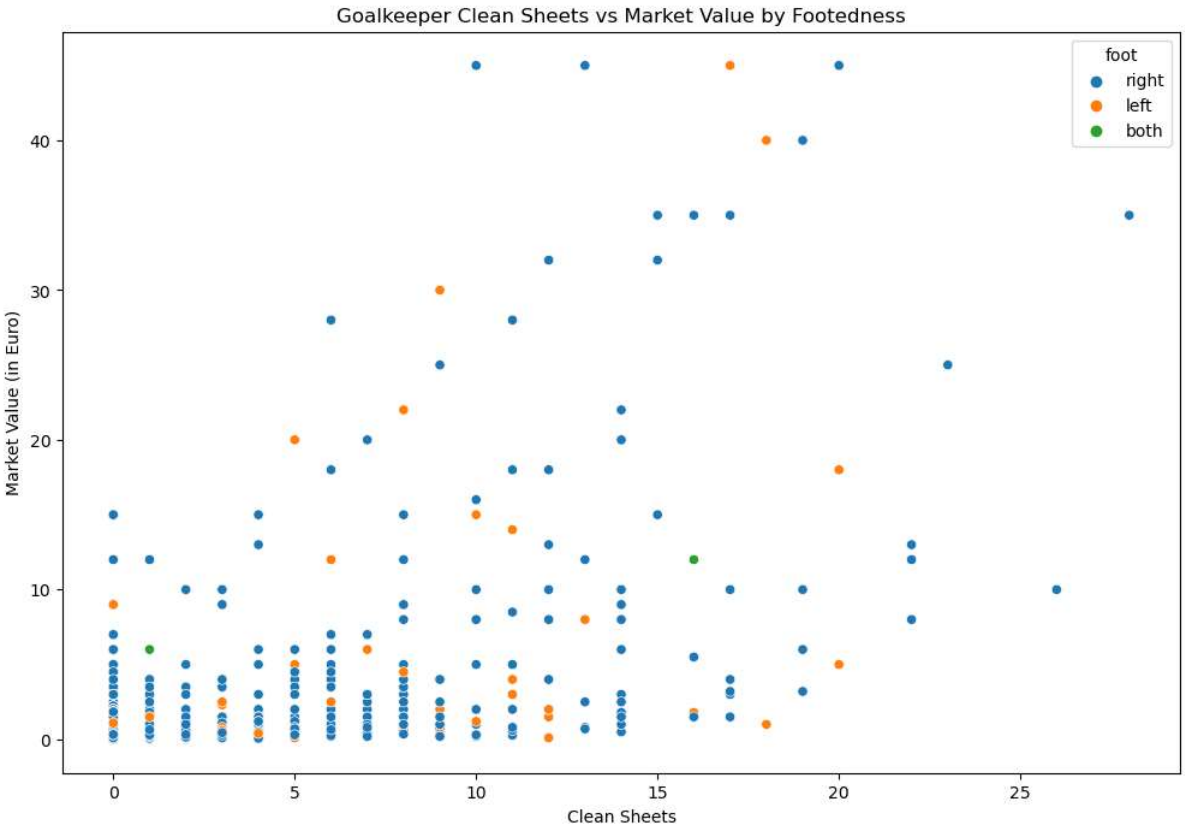
In [248…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='clean_sheet_2022', y= goalkeeper_grouped_22['market_value_in_eur
plt.title('Goalkeeper Clean Sheets vs Market Value by Position')
plt.xlabel('Clean Sheets')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



In [249…

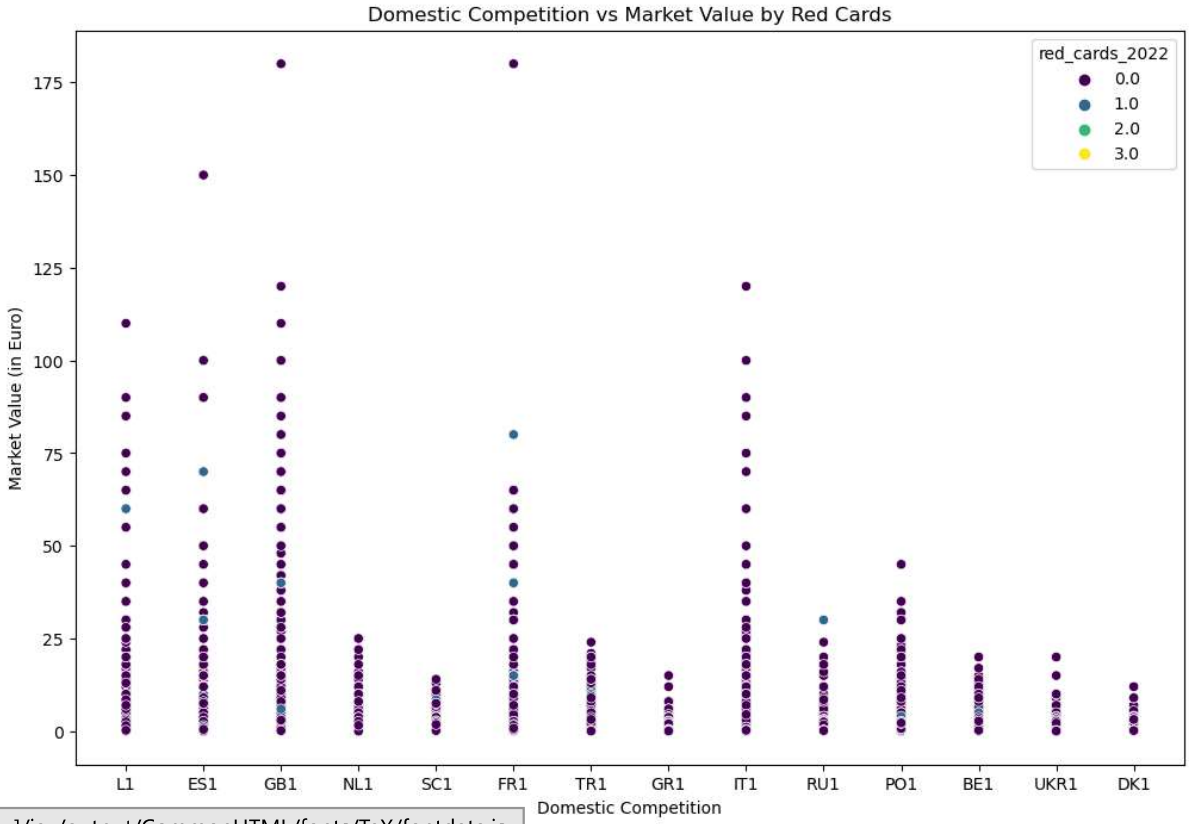
```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='clean_sheet_2022', y=goalkeeper_grouped_22['market_value_in_eur'
plt.title('Goalkeeper Clean Sheets vs Market Value by Footedness')
plt.xlabel('Clean Sheets')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



Red Cards 22/23

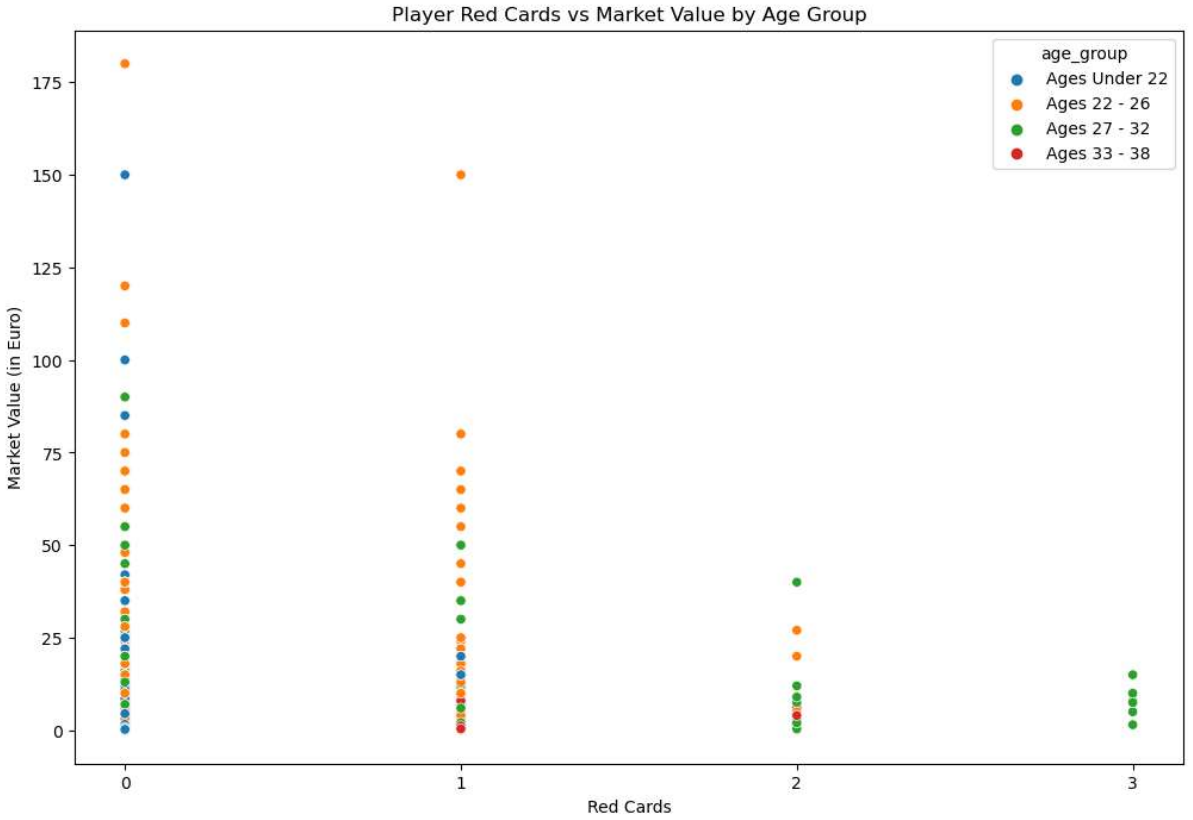
In [255…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='current_club_domestic_competition_id', y= merged_players_df['mar
plt.title('Domestic Competition vs Market Value by Red Cards')
plt.xlabel('Domestic Competition')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



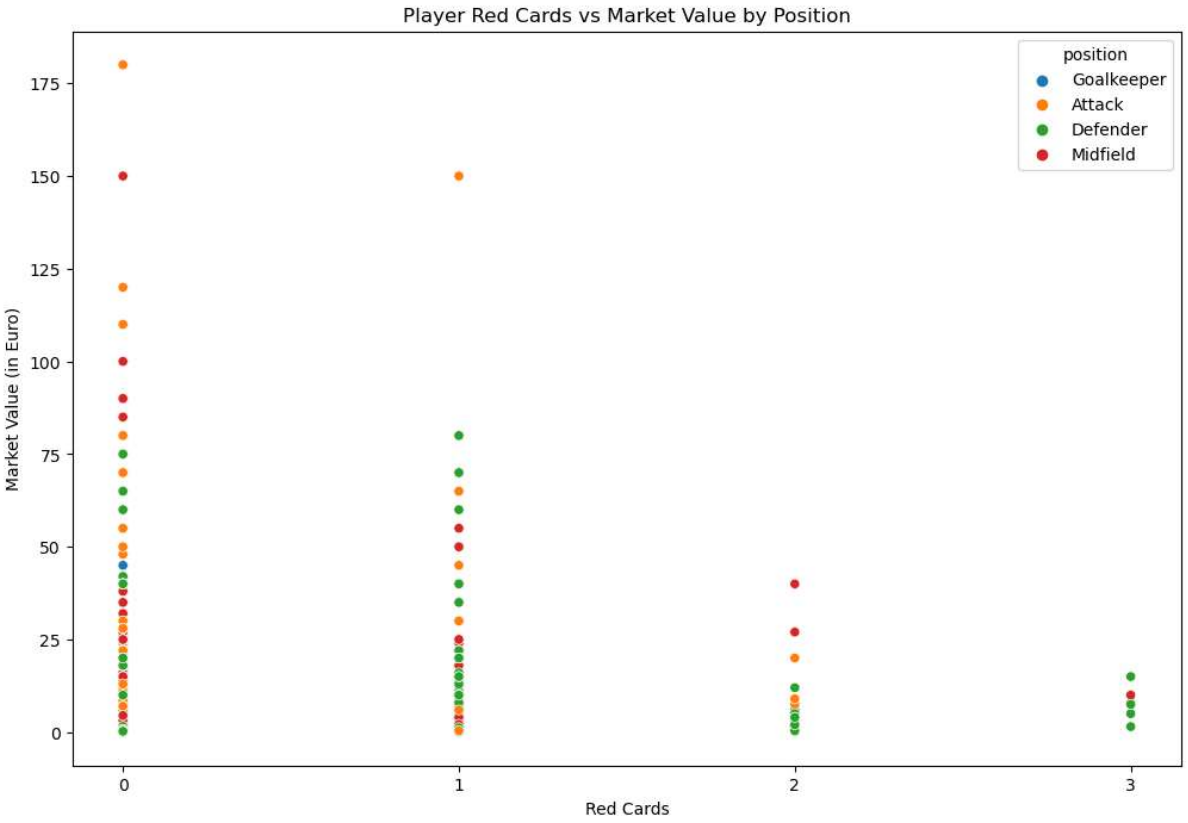
In [265…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='red_cards_2022', y= merged_players_df['market_value_in_eur']/100
plt.title('Player Red Cards vs Market Value by Age Group')
plt.xlabel('Red Cards')
plt.ylabel('Market Value (in Euro)')
plt.xticks( ticks=merged_players_df['red_cards_2022'].unique())
plt.show()
```



In [261…

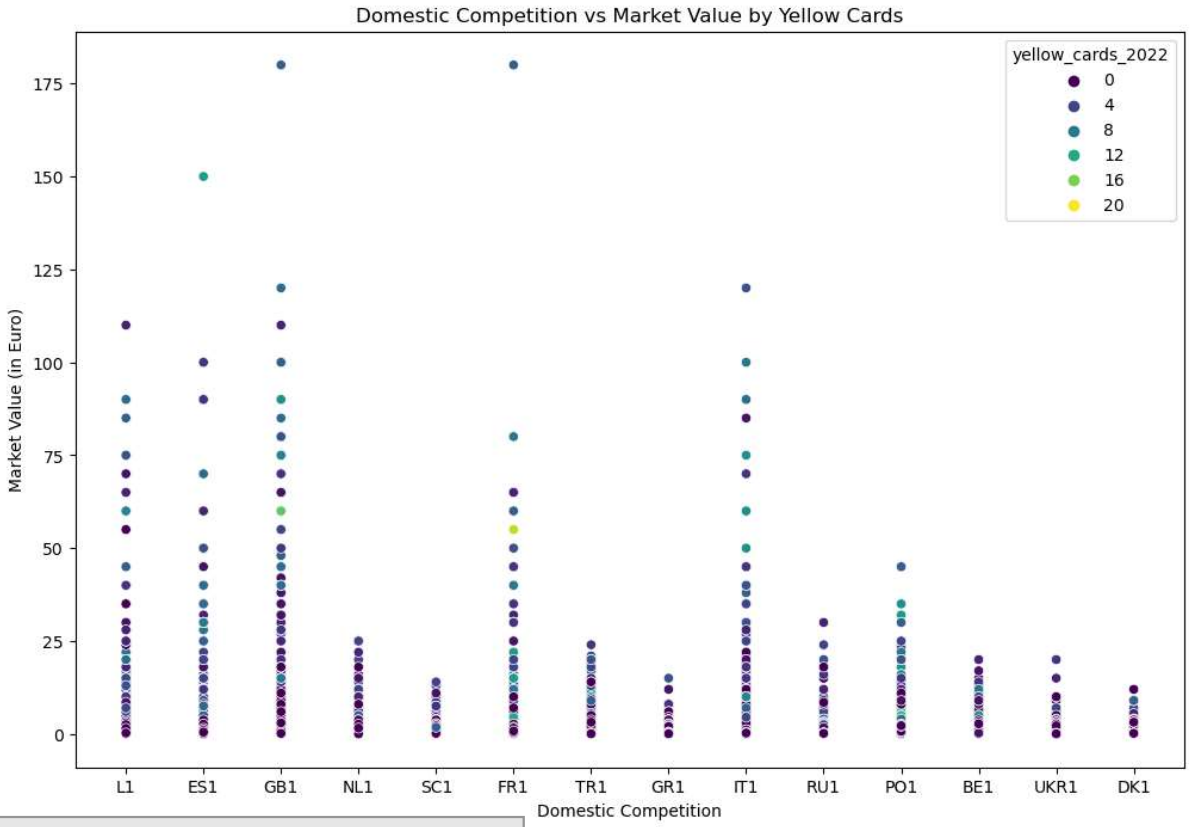
```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='red_cards_2022', y= merged_players_df['market_value_in_eur']/100
plt.title('Player Red Cards vs Market Value by Position')
plt.xlabel('Red Cards')
plt.ylabel('Market Value (in Euro)')
plt.xticks( ticks=merged_players_df['red_cards_2022'].unique())
plt.show()
```



Yellow Cards 22/23

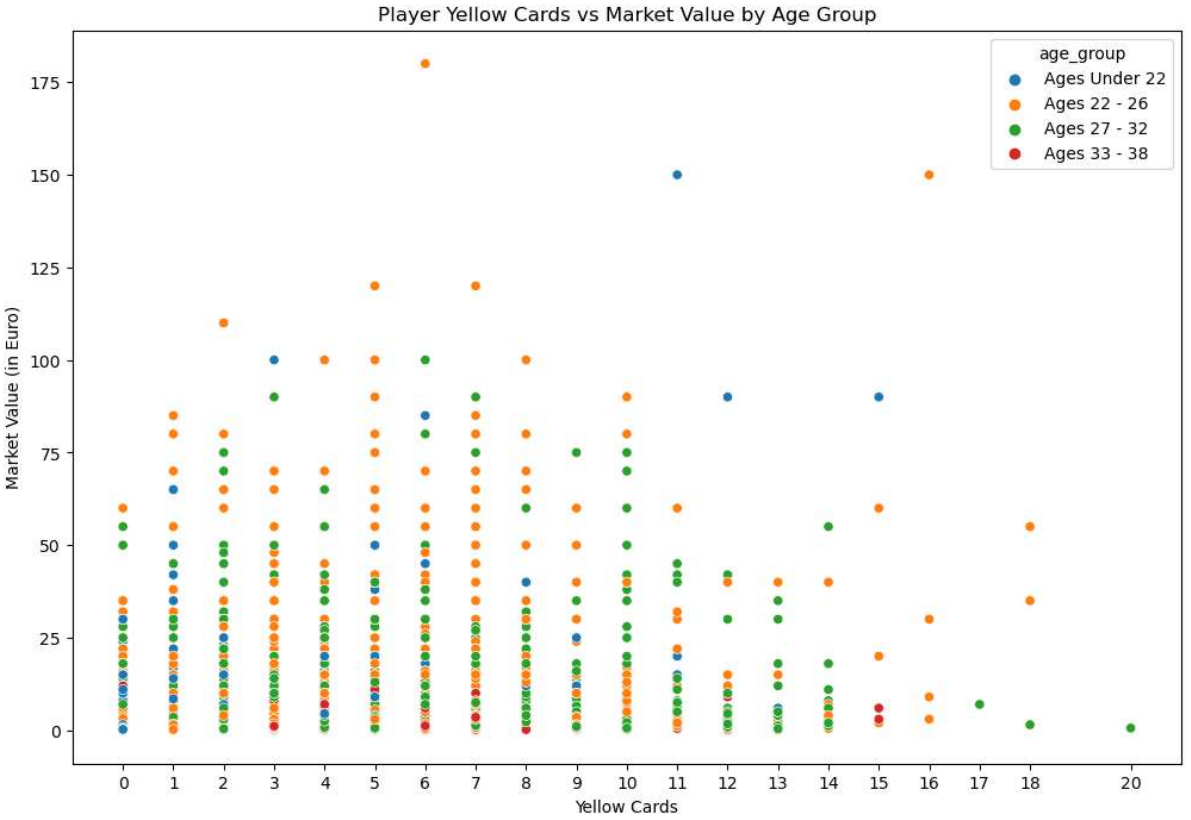
In [259…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='current_club_domestic_competition_id', y= merged_players_df['mar
plt.title('Domestic Competition vs Market Value by Yellow Cards')
plt.xlabel('Domestic Competition')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



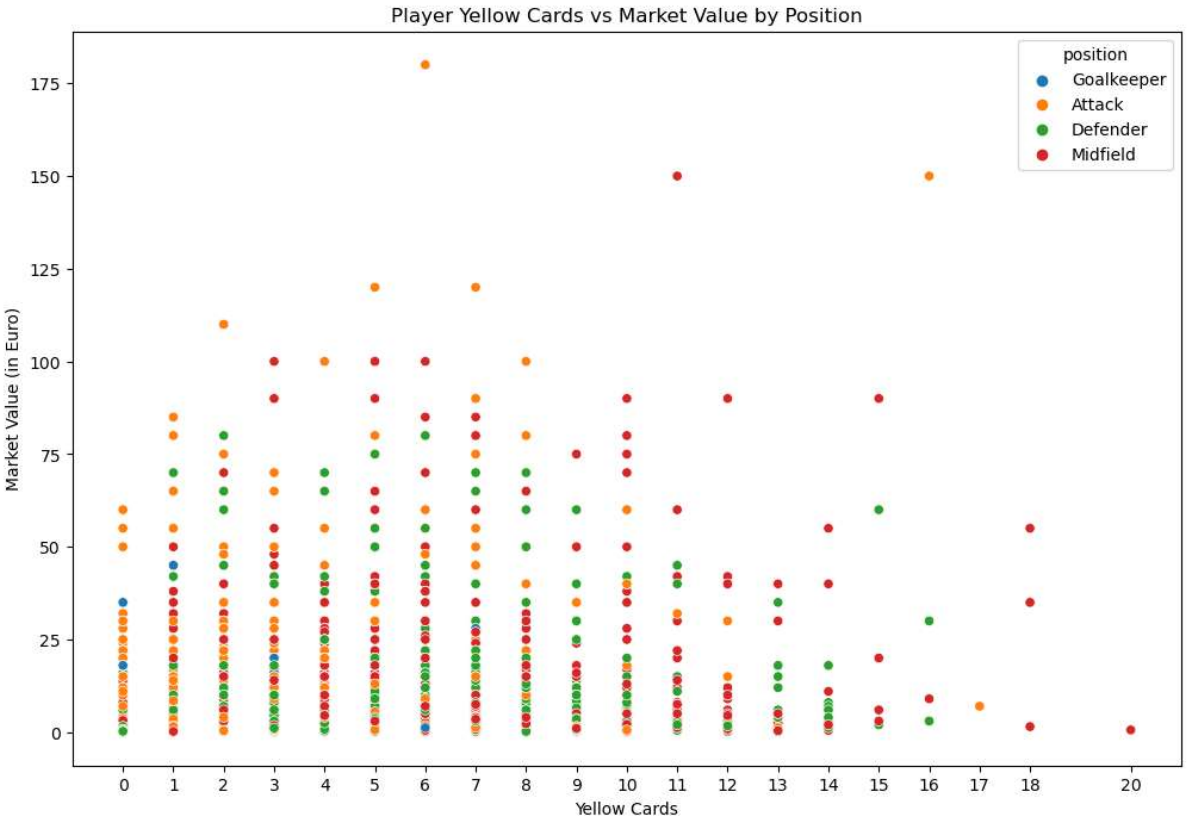
In [264…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='yellow_cards_2022', y= merged_players_df['market_value_in_eur']/
plt.title('Player Yellow Cards vs Market Value by Age Group')
plt.xlabel('Yellow Cards')
plt.ylabel('Market Value (in Euro)')
plt.xticks( ticks=merged_players_df['yellow_cards_2022'].unique())
plt.show()
```



In [263…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='yellow_cards_2022', y= merged_players_df['market_value_in_eur']/
plt.title('Player Yellow Cards vs Market Value by Position')
plt.xlabel('Yellow Cards')
plt.ylabel('Market Value (in Euro)')
plt.xticks( ticks=merged_players_df['yellow_cards_2022'].unique())
plt.show()
```

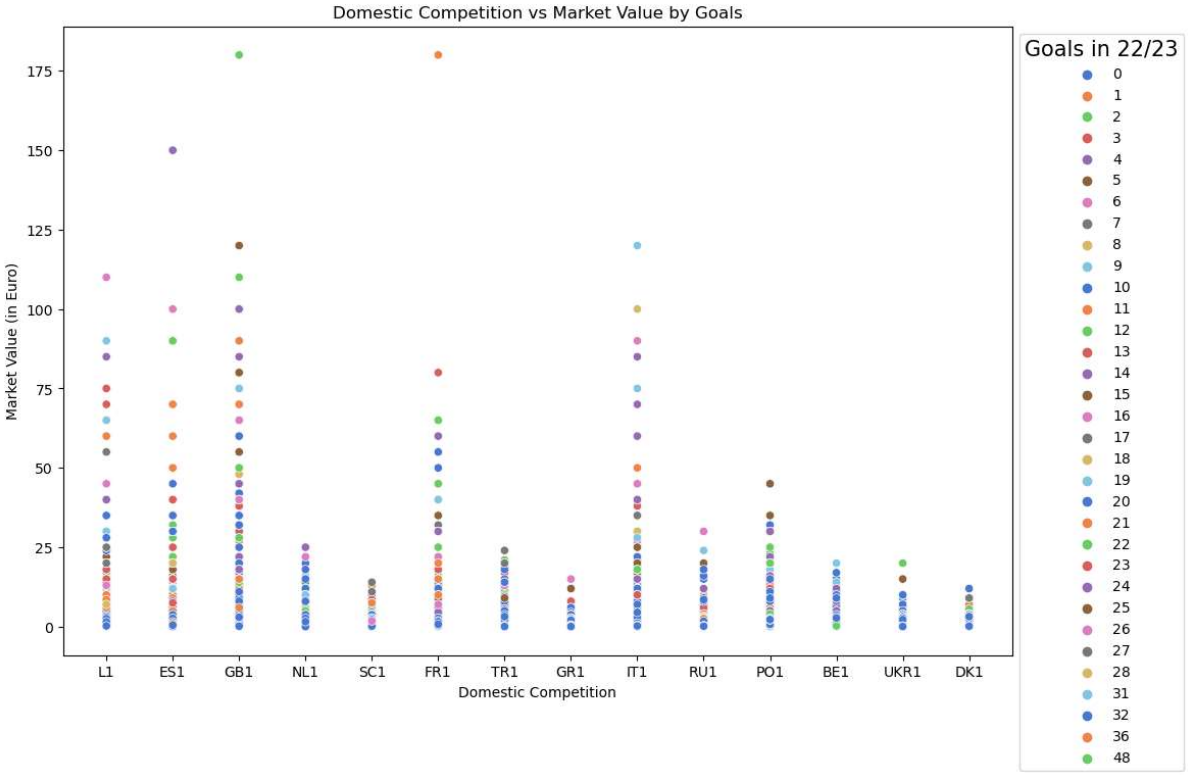


Statistics by Domestic League 22/23

Goals

In [280…

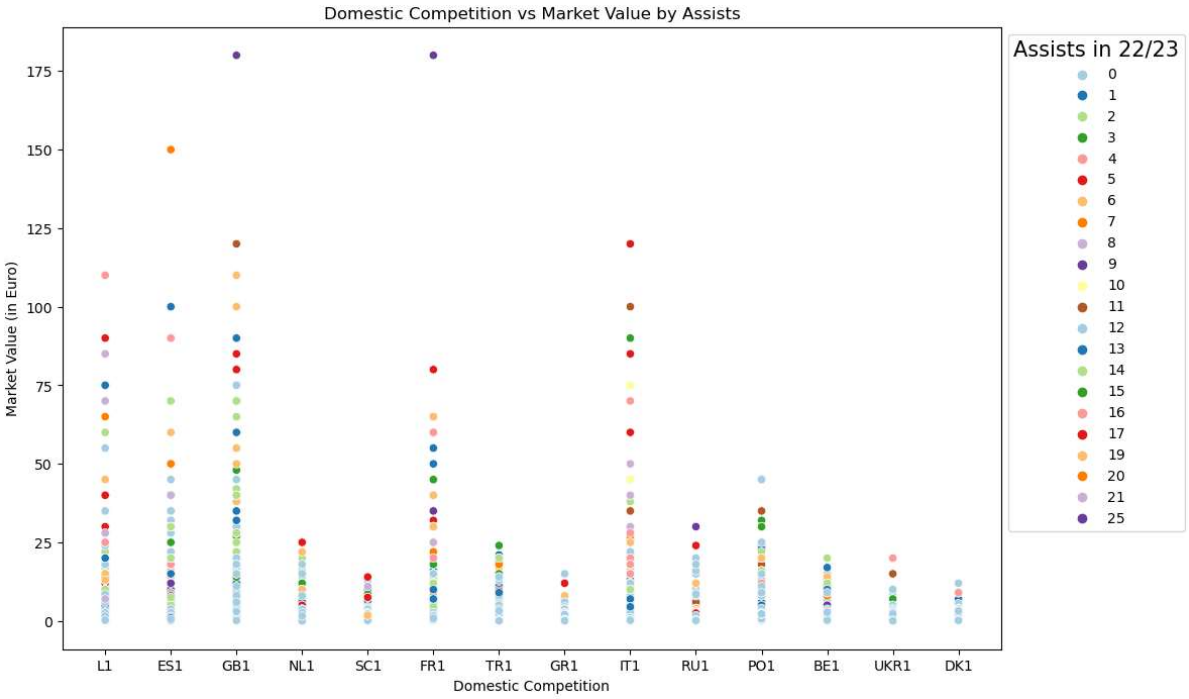
```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='current_club_domestic_competition_id', y= merged_players_df['mar
plt.title('Domestic Competition vs Market Value by Goals')
plt.xlabel('Domestic Competition')
plt.ylabel('Market Value (in Euro)')
plt.legend(title='Goals in 22/23', title_fontsize='15', loc='upper left', bbox_to_a

plt.show()
```



Assists

In [284…

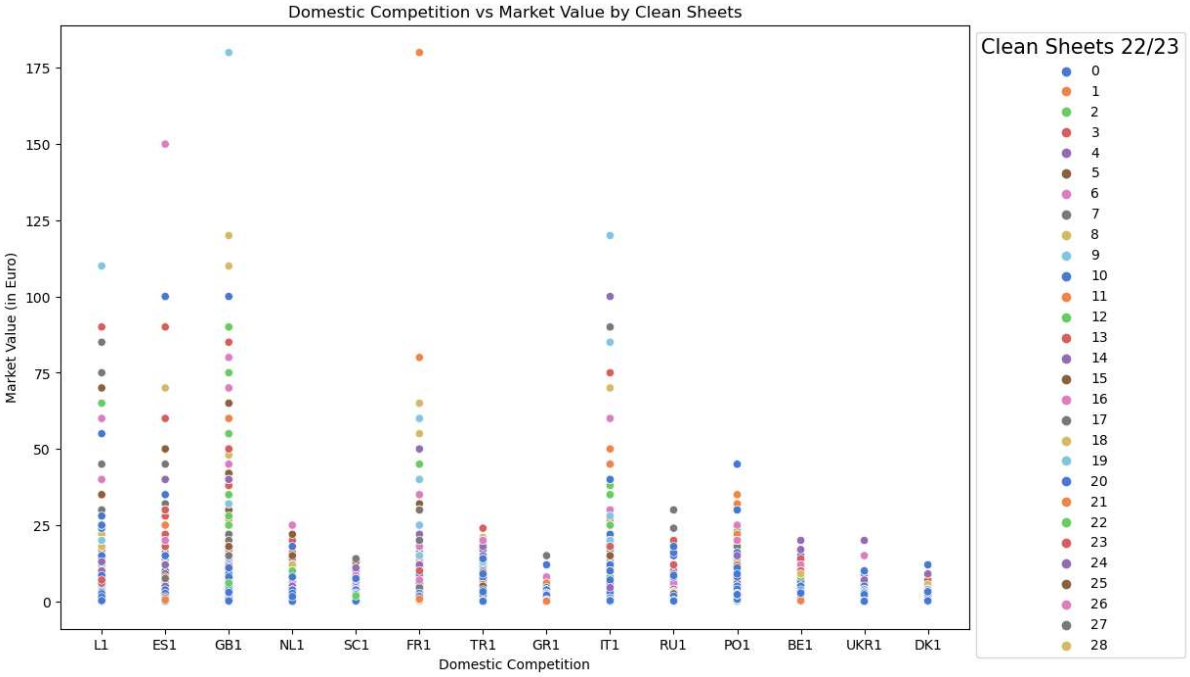
```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='current_club_domestic_competition_id', y= merged_players_df['mar
plt.title('Domestic Competition vs Market Value by Assists')
plt.xlabel('Domestic Competition')
plt.ylabel('Market Value (in Euro)')
plt.legend(title='Assists in 22/23', title_fontsize='15', loc='upper left', bbox_to
plt.show()
```



Clean Sheets

```
sns.scatterplot(x='current_club_domestic_competition_id', y= merged_players_df['mar
```

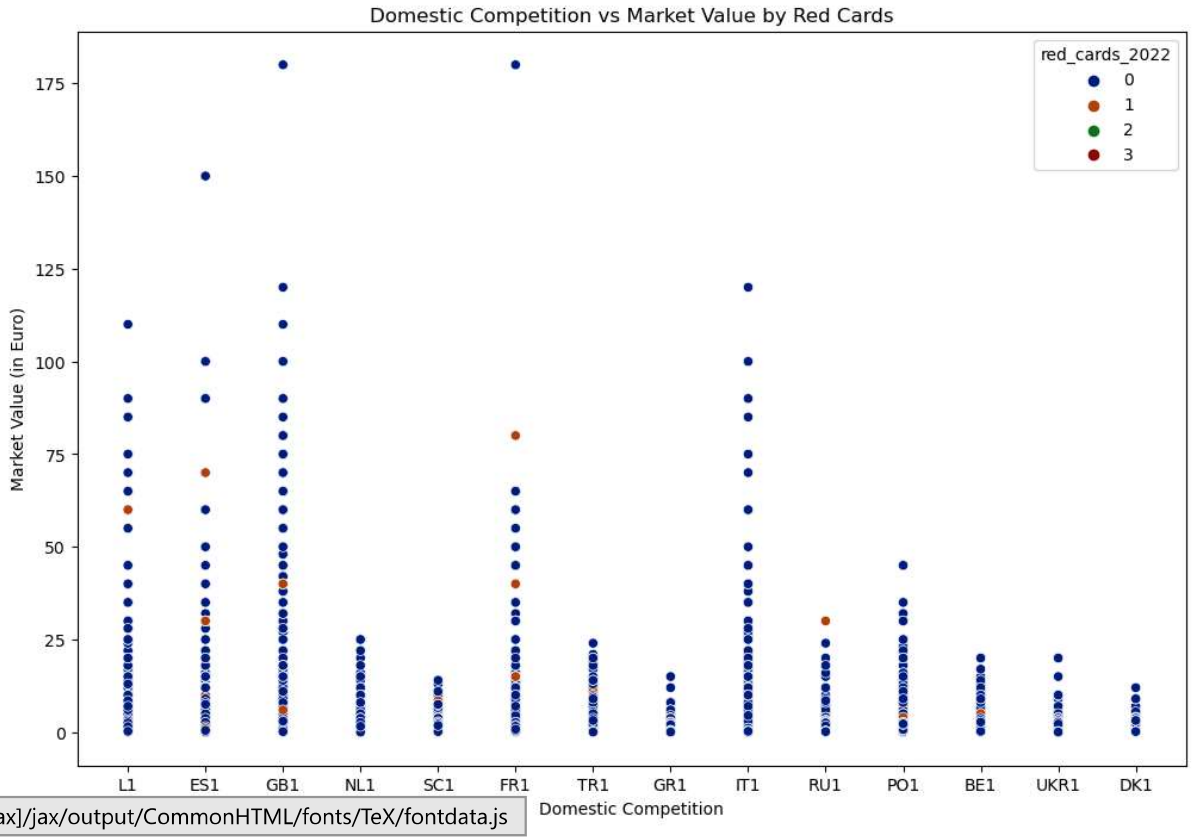
```
plt.title('Domestic Competition vs Market Value by Clean Sheets')
plt.xlabel('Domestic Competition')
plt.ylabel('Market Value (in Euro)')
plt.legend(title='Clean Sheets 22/23', title_fontsize='15', loc='upper left', bbox_
plt.show()
```



Red Cards

In [296…

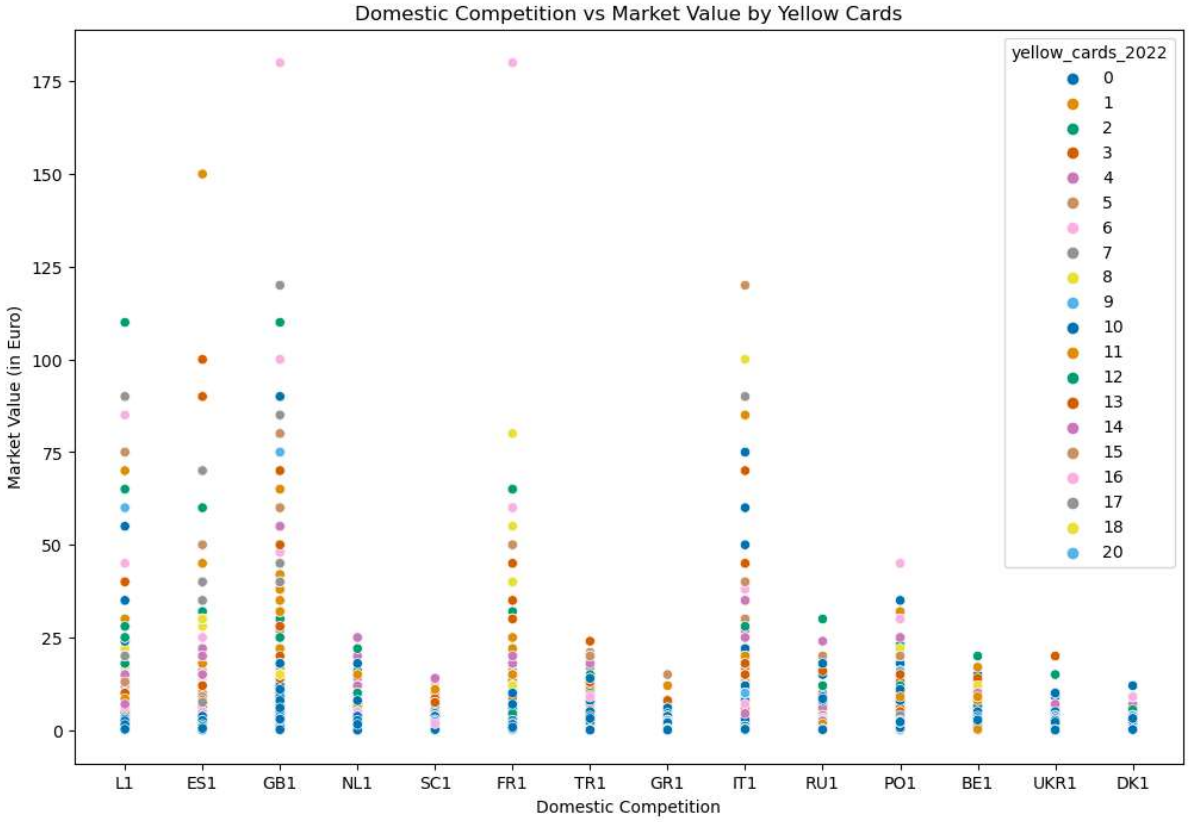
```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='current_club_domestic_competition_id', y= merged_players_df['mar
plt.title('Domestic Competition vs Market Value by Red Cards')
plt.xlabel('Domestic Competition')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



Yellow Cards

In [297…

```
plt.figure(figsize=(12, 8))
sns.scatterplot(x='current_club_domestic_competition_id', y= merged_players_df['mar
plt.title('Domestic Competition vs Market Value by Yellow Cards')
plt.xlabel('Domestic Competition')
plt.ylabel('Market Value (in Euro)')
plt.show()
```



In [298…

```
merged_players_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 30130 entries, 0 to 30301
Data columns (total 79 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   player_id                             30130 non-null  int64
 1   first_name                            28173 non-null  object
 2   last_name                             30130 non-null  object
 3   name                                  30130 non-null  object
 4   last_season                           30130 non-null  int64
 5   current_club_id                       30130 non-null  int64
 6   player_code                           30130 non-null  object
 7   country_of_birth                      27528 non-null  object
 8   city_of_birth                         28013 non-null  object
 9   country_of_citizenship                29587 non-null  object
 10  date_of_birth                         30130 non-null  datetime64[ns]
 11  sub_position                          30130 non-null  object
 12  position                              30130 non-null  object
 13  foot                                  27844 non-null  object
 14  height_in_cm                          28099 non-null  float64
 15  market_value_in_eur                   30130 non-null  float64
 16  highest_market_value_in_eur           30130 non-null  float64
 17  contract_expiration_date              18799 non-null  datetime64[ns]
 18  image_url                             30130 non-null  object
 19  url                                   30130 non-null  object
 20  current_club_domestic_competition_id  30130 non-null  object
 21  current_club_name                     30130 non-null  object
 22  age                                   30130 non-null  int32
 23  remaining_contract_days               18799 non-null  float64
 24  age_group                             30130 non-null  object
 25  games_total                           30130 non-null  int64
 26  goals_total                           30130 non-null  int64
 27  assists_total                         30130 non-null  int64
 28  minutes_played_total                  30130 non-null  int64
 29  goals_for_total                       30130 non-null  int64
 30  goals_against_total                   30130 non-null  int64
 31  clean_sheet_total                     30130 non-null  int64
 32  yellow_cards_total                    30130 non-null  int64
 33  red_cards_total                       30130 non-null  int64
 34  games_2019                            30130 non-null  float64
 35  goals_2019                            30130 non-null  float64
 36  assists_2019                          30130 non-null  float64
 37  minutes_played_2019                   30130 non-null  float64
 38  goals_for_2019                        30130 non-null  float64
 39  goals_against_2019                    30130 non-null  float64
 40  clean_sheet_2019                      30130 non-null  float64
 41  yellow_cards_2019                     30130 non-null  float64
 42  red_cards_2019                        30130 non-null  float64
 43  games_2020                            30130 non-null  float64
 44  goals_2020                            30130 non-null  float64
 45  assists_2020                          30130 non-null  float64
 46  minutes_played_2020                   30130 non-null  float64
 47  goals_for_2020                        30130 non-null  float64
 48  goals_against_2020                    30130 non-null  float64
 49  clean_sheet_2020                      30130 non-null  float64
 50  yellow_cards_2020                     30130 non-null  float64
 51  red_cards_2020                        30130 non-null  float64
 52  games_2021                            30130 non-null  float64
 53  goals_2021                            30130 non-null  float64
 54  assists_2021                          30130 non-null  float64
 55  minutes_played_2021                   30130 non-null  float64
 56  goals_for_2021                        30130 non-null  float64
                                           30130 non-null  float64
 58  clean_sheet_2021                      30130 non-null  float64
```

```
 59  yellow_cards_2021                     30130 non-null  float64
 60  red_cards_2021                        30130 non-null  float64
 61  games_2022                            30130 non-null  float64
 62  goals_2022                            30130 non-null  int32
 63  assists_2022                          30130 non-null  int32
 64  minutes_played_2022                   30130 non-null  float64
 65  goals_for_2022                        30130 non-null  float64
 66  goals_against_2022                    30130 non-null  float64
 67  clean_sheet_2022                      30130 non-null  int32
 68  yellow_cards_2022                     30130 non-null  int32
 69  red_cards_2022                        30130 non-null  int32
 70  games_2023                            30130 non-null  float64
 71  goals_2023                            30130 non-null  float64
 72  assists_2023                          30130 non-null  float64
 73  minutes_played_2023                   30130 non-null  float64
 74  goals_for_2023                        30130 non-null  float64
 75  goals_against_2023                    30130 non-null  float64
 76  clean_sheet_2023                      30130 non-null  float64
 77  yellow_cards_2023                     30130 non-null  float64
 78  red_cards_2023                        30130 non-null  float64
dtypes: datetime64[ns](2), float64(44), int32(6), int64(12), object(15)
memory usage: 18.7+ MB
```

Saving merged datasets for later use

In [301…

```
merged_players_df.to_csv('Merged_Data_4_Pre_Processing.csv', index=False)
```

References

www.kaggle.com. (n.d.). Football Data from Transfermarkt. [online] Available at: https://www.kaggle.com/datasets/davidcariboo/player-scores/versions/284/data [Accessed 29 Oct. 2023].

www.footballbenchmark.com. (n.d.). Football Benchmark - Methodology and limitations of published information. [online] Available at: https://www.footballbenchmark.com/methodology/player_valuation.

www.transfermarkt.co.uk. (n.d.). Aïssa Boudechicha - Player profile 23/24. [online] Available at: https://www.transfermarkt.co.uk/aissa-boudechicha/profil/spieler/592398 [Accessed 16 Dec. 2023].

www.transfermarkt.co.uk. (n.d.). Genar Fornés - Player profile 23/24. [online] Available at: https://www.transfermarkt.co.uk/genar-fornes/profil/spieler/628490 [Accessed 5 Dec. 2023].

kaggle.com. (n.d.). Football Transfer Market EDA & Basic Modelling. [online] Available at: https://www.kaggle.com/code/davidcoxon/football-transfer-market-eda-basic-modelling [Accessed 19 Dec. 2023].

Wirth, R. and Hipp, J. (2000). CRISP-DM: Towards a Standard Process Model for Data Mining . Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining, (Vol. 1, pp. 29-39).