Data analysis Project-football market value analysis

November 21, 2023

1 Project-Football Market Value Analysis

1.0.1 Objective:

Perform the analysis on the given dataset and write the observed insights about the relationship between the market value and the following categories:

Position

Country

Age

Club

Players

Also, compare the two most valuable players based on various parameters.

2 MetaData

2.1 Dataset Name: Most Expensive Footballers 2021

2.1.1 Description:

This dataset encompasses statistical data of football players, detailing their performance and attributes across different countries and clubs.

2.1.2 Columns:

1. `Name` - Player's name (string)
2. `Position` - Player's position on the field (string)
3. `Age` - Player's age (integer)
4. `Market_value` - Estimated market value of the player (float)
5. `Country` - Country where the player is active (string)
6. `Club` - Football club the player belongs to (string)
7. `Matches` - Total matches played by the player (integer)
8. `Goals` - Total goals scored by the player (integer)
9. `Own_goals` - Number of own goals scored by the player (integer)
10. `Assists` - Total assists provided by the player (integer)
11. `Yellow_cards` - Total yellow cards received by the player (integer)
12. `Second_yellow_cards` - Total second yellow cards received by the player (integer)

13. `Red_cards` - Total red cards received by the player (integer)
14. `Substitute_in` - Number of times the player was substituted in (integer)
15. `Substitute_out` - Number of times the player was substituted out (integer)

2.1.3 Data Types:

- `Name`: String
- `Position`: String
- `Age`: Integer
- `Market_value`: Float
- `Country`: String
- `Club`: String
- `Matches`: Integer
- `Goals`: Integer
- `Own_goals`: Integer
- `Assists`: Integer
- `Yellow_cards`: Integer
- `Second_yellow_cards`: Integer
- `Red_cards`: Integer
- `Substitute_in`: Integer
- `Substitute_out`: Integer

2.1.4 Notes:

- The dataset contains 500 entries for various football players.
- 'Position' denotes the player's specific role on the field (e.g., Forward, Midfielder, Defender, Goalkeeper, etc.).
- 'Market_value' provides an estimated value of the player in the football market.
- The dataset contains no missing values (all columns have 500 non-null entries).
- Ensure consistency in data recording and categorization for accurate analysis across countries and clubs.

3 1. Loading the libraries and reading the csv

[152]:
```
# importing libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from mlxtend.frequent_patterns import apriori, association_rules
from mlxtend.preprocessing import TransactionEncoder
```

[2]:
```
#reading csv file
df=pd.read_csv("players.csv")
df
```

[2]:

```
     Unnamed: 0                     Name             Position  Age  \
0             0            Kylian Mbappé       Centre-Forward   22
1             1           Erling Haaland       Centre-Forward   21
2             2               Harry Kane       Centre-Forward   28
3             3            Jack Grealish          Left Winger   26
4             4            Mohamed Salah         Right Winger   29
..          …                        …                  …  …
495         495  Giorgian de Arrascaeta  Attacking Midfield   27
496         496              Ayoze Pérez       Second Striker   28
497         497               Alex Meret           Goalkeeper   24
498         498         Duje Caleta-Car          Centre-Back   25
499         499         Aritz Elustondo          Centre-Back   27

     Markey Value In Millions(£)  Country                         Club  \
0                          144.0   France          Paris Saint-Germain
1                          135.0   Norway            Borussia Dortmund
2                          108.0  England            Tottenham Hotspur
3                           90.0  England              Manchester City
4                           90.0    Egypt                 Liverpool FC
..                           …      …                            …
495                         16.2  Uruguay  Clube de Regatas do Flamengo
496                         16.2    Spain               Leicester City
497                         16.2    Italy                   SSC Napoli
498                         16.2  Croatia          Olympique Marseille
499                         16.2    Spain                Real Sociedad

     Matches  Goals  Own Goals  Assists  Yellow Cards  Second Yellow Cards  \
0         16      7          0       11             3                    0
1         10     13          0        4             1                    0
2         16      7          0        2             2                    0
3         15      2          0        3             1                    0
4         15     15          0        6             1                    0
..       …    …        …      …           …                  …
495        0      0          0        0             0                    0
496        8      1          0        3             0                    0
497        5      0          0        0             0                    0
498        8      0          0        0             2                    0
499       15      3          0        1             4                    0

     Red Cards  Number Of Substitute In  Number Of Substitute Out
0            0                        0                         8
1            0                        0                         1
2            0                        2                         2
3            0                        2                         8
4            0                        0                         3
..         …                      …                       …
495          0                        0                         0
```

```
496          1                    2                    5
497          0                    0                    0
498          0                    0                    2
499          0                    1                    1

[500 rows x 16 columns]
```

4 2. Exploratory Analysis

[5]:
```
df.head()
```

[5]:
```
   Unnamed: 0            Name        Position  Age  \
0           0   Kylian Mbappé  Centre-Forward   22
1           1  Erling Haaland  Centre-Forward   21
2           2      Harry Kane  Centre-Forward   28
3           3   Jack Grealish     Left Winger   26
4           4   Mohamed Salah    Right Winger   29

   Markey Value In Millions(£)  Country                 Club  Matches  Goals  \
0                        144.0   France  Paris Saint-Germain       16      7
1                        135.0   Norway    Borussia Dortmund       10     13
2                        108.0  England    Tottenham Hotspur       16      7
3                         90.0  England      Manchester City       15      2
4                         90.0    Egypt         Liverpool FC       15     15

   Own Goals  Assists  Yellow Cards  Second Yellow Cards  Red Cards  \
0          0       11             3                    0          0
1          0        4             1                    0          0
2          0        2             2                    0          0
3          0        3             1                    0          0
4          0        6             1                    0          0

   Number Of Substitute In  Number Of Substitute Out
0                        0                         8
1                        0                         1
2                        2                         2
3                        2                         8
4                        0                         3
```

[4]:
```
df.tail()
```

[4]:
```
     Unnamed: 0                     Name            Position  Age  \
495         495  Giorgian de Arrascaeta  Attacking Midfield   27
496         496             Ayoze Pérez      Second Striker   28
497         497              Alex Meret          Goalkeeper   24
498         498          Duje Caleta-Car         Centre-Back   25
```

```
499         499         Aritz Elustondo         Centre-Back   27

     Markey Value In Millions(£)  Country                         Club  \
495                         16.2  Uruguay  Clube de Regatas do Flamengo
496                         16.2    Spain                Leicester City
497                         16.2    Italy                    SSC Napoli
498                         16.2  Croatia           Olympique Marseille
499                         16.2    Spain                 Real Sociedad

     Matches  Goals  Own Goals  Assists  Yellow Cards  Second Yellow Cards  \
495        0      0          0        0             0                    0
496        8      1          0        3             0                    0
497        5      0          0        0             0                    0
498        8      0          0        0             2                    0
499       15      3          0        1             4                    0

     Red Cards  Number Of Substitute In  Number Of Substitute Out
495          0                        0                         0
496          1                        2                         5
497          0                        0                         0
498          0                        0                         2
499          0                        1                         1
```

[6]:
```
#removing the unnamed columns
df.drop(columns="Unnamed: 0",axis=1,inplace=True)
df
```

[6]:
```
                     Name           Position  Age  \
0           Kylian Mbappé     Centre-Forward   22
1          Erling Haaland     Centre-Forward   21
2              Harry Kane     Centre-Forward   28
3            Jack Grealish        Left Winger   26
4            Mohamed Salah       Right Winger   29
..                     …                  …  …
495  Giorgian de Arrascaeta  Attacking Midfield   27
496             Ayoze Pérez     Second Striker   28
497              Alex Meret         Goalkeeper   24
498         Duje Caleta-Car        Centre-Back   25
499         Aritz Elustondo        Centre-Back   27

     Markey Value In Millions(£)  Country                         Club  \
0                          144.0   France           Paris Saint-Germain
1                          135.0   Norway             Borussia Dortmund
2                          108.0  England             Tottenham Hotspur
3                           90.0  England               Manchester City
4                           90.0    Egypt                  Liverpool FC
..                           …        …                           …
```

```
495                        16.2  Uruguay  Clube de Regatas do Flamengo
496                        16.2    Spain               Leicester City
497                        16.2    Italy                   SSC Napoli
498                        16.2  Croatia          Olympique Marseille
499                        16.2    Spain                Real Sociedad

     Matches  Goals  Own Goals  Assists  Yellow Cards  Second Yellow Cards  \
0         16      7          0       11             3                    0
1         10     13          0        4             1                    0
2         16      7          0        2             2                    0
3         15      2          0        3             1                    0
4         15     15          0        6             1                    0
..       …      …          …        …             …                    …
495        0      0          0        0             0                    0
496        8      1          0        3             0                    0
497        5      0          0        0             0                    0
498        8      0          0        0             2                    0
499       15      3          0        1             4                    0

     Red Cards  Number Of Substitute In  Number Of Substitute Out
0            0                        0                         8
1            0                        1                         1
2            0                        2                         2
3            0                        2                         8
4            0                        0                         3
..         …                      …                         …
495          0                        0                         0
496          1                        2                         5
497          0                        0                         0
498          0                        0                         2
499          0                        1                         1

[500 rows x 15 columns]
```

[7]:

```python
#renaming some columns for better use
df = df.rename(columns=
            {'Markey Value In Millions(£)': 'Market_value','Own Goals':␣
 ↪'Own_goals',
             'Yellow Cards': 'Yellow_cards','Second Yellow Cards':␣
 ↪'Second_yellow_cards',
             'Red Cards': 'Red_cards', 'Number Of Substitute In':␣
 ↪'Substitute_in',
             'Number Of Substitute Out': 'Substitute_out'})
df
```

[7]:

```
                     Name          Position  Age  Market_value  Country  \
0           Kylian Mbappé     Centre-Forward   22         144.0   France
```

```
1          Erling Haaland      Centre-Forward   21         135.0   Norway
2              Harry Kane      Centre-Forward   28         108.0  England
3           Jack Grealish         Left Winger   26          90.0  England
4           Mohamed Salah        Right Winger   29          90.0    Egypt
..                    …                 …   …            …       …
495  Giorgian de Arrascaeta  Attacking Midfield   27          16.2  Uruguay
496            Ayoze Pérez      Second Striker   28          16.2    Spain
497             Alex Meret          Goalkeeper   24          16.2    Italy
498        Duje Caleta-Car          Centre-Back   25          16.2  Croatia
499        Aritz Elustondo          Centre-Back   27          16.2    Spain

                          Club  Matches  Goals  Own_goals  Assists  \
0          Paris Saint-Germain       16      7          0       11
1            Borussia Dortmund       10     13          0        4
2            Tottenham Hotspur       16      7          0        2
3              Manchester City       15      2          0        3
4                 Liverpool FC       15     15          0        6
..                         …        …      …          …        …
495  Clube de Regatas do Flamengo        0      0          0        0
496              Leicester City        8      1          0        3
497                 SSC Napoli        5      0          0        0
498         Olympique Marseille        8      0          0        0
499               Real Sociedad       15      3          0        1

     Yellow_cards  Second_yellow_cards  Red_cards  Substitute_in  \
0               3                    0          0              0
1               1                    0          0              0
2               2                    0          0              2
3               1                    0          0              2
4               1                    0          0              0
..            …                    …          …              …
495             0                    0          0              0
496             0                    0          1              2
497             0                    0          0              0
498             2                    0          0              0
499             4                    0          0              1

     Substitute_out
0                 8
1                 1
2                 2
3                 8
4                 3
..              …
495               0
496               5
497               0
```

```
498               2
499               1

[500 rows x 15 columns]
```

[8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 15 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Name                 500 non-null    object
 1   Position             500 non-null    object
 2   Age                  500 non-null    int64
 3   Market_value         500 non-null    float64
 4   Country              500 non-null    object
 5   Club                 500 non-null    object
 6   Matches              500 non-null    int64
 7   Goals                500 non-null    int64
 8   Own_goals            500 non-null    int64
 9   Assists              500 non-null    int64
 10  Yellow_cards         500 non-null    int64
 11  Second_yellow_cards  500 non-null    int64
 12  Red_cards            500 non-null    int64
 13  Substitute_in        500 non-null    int64
 14  Substitute_out       500 non-null    int64
dtypes: float64(1), int64(10), object(4)
memory usage: 58.7+ KB
```

[9]:
```
df.dtypes
```

[9]:
```
Name                    object
Position                object
Age                      int64
Market_value           float64
Country                 object
Club                    object
Matches                  int64
Goals                    int64
Own_goals                int64
Assists                  int64
Yellow_cards             int64
Second_yellow_cards      int64
Red_cards                int64
Substitute_in            int64
Substitute_out           int64
dtype: object
```

[10]:
```
#checking for any null values
df.isna().sum()
```

[10]:
```
Name                   0
Position               0
Age                    0
Market_value           0
Country                0
Club                   0
Matches                0
Goals                  0
Own_goals              0
Assists                0
Yellow_cards           0
Second_yellow_cards    0
Red_cards              0
Substitute_in          0
Substitute_out         0
dtype: int64
```

[11]:
```
#descriptive statistics of the dataset
df.describe()
```

[11]:
```
              Age  Market_value     Matches       Goals   Own_goals  \
count  500.000000    500.000000  500.000000  500.000000  500.000000
mean    24.968000     31.537800   12.396000    2.160000    0.030000
std      3.165916     17.577697    4.342453    2.880102    0.170758
min     16.000000     16.200000    0.000000    0.000000    0.000000
25%     23.000000     19.800000   10.000000    0.000000    0.000000
50%     25.000000     25.200000   13.000000    1.000000    0.000000
75%     27.000000     36.000000   16.000000    3.000000    0.000000
max     36.000000    144.000000   24.000000   23.000000    1.000000

         Assists  Yellow_cards  Second_yellow_cards   Red_cards  \
count  500.00000    500.000000           500.000000  500.000000
mean     1.51200      1.592000             0.036000    0.046000
std      1.85276      1.445585             0.186477    0.209695
min      0.00000      0.000000             0.000000    0.000000
25%      0.00000      0.000000             0.000000    0.000000
50%      1.00000      1.000000             0.000000    0.000000
75%      2.00000      2.000000             0.000000    0.000000
max     12.00000      7.000000             1.000000    1.000000

       Substitute_in  Substitute_out
count     500.000000      500.000000
mean        2.394000        3.744000
std         2.517825        3.293046
```

```
min         0.000000        0.000000
25%         0.000000        1.000000
50%         2.000000        3.000000
75%         3.250000        6.000000
max        13.000000       20.000000
```

[13]:
```
df.shape
```

[13]:
```
(500, 15)
```

[15]:
```
df['Club'].unique()
```

[15]:
```
array(['Paris Saint-Germain', 'Borussia Dortmund', 'Tottenham Hotspur',
       'Manchester City', 'Liverpool FC', 'Chelsea FC',
       'Manchester United', 'FC Barcelona', 'Bayern Munich',
       'Inter Milan', 'Atlético de Madrid', 'West Ham United',
       'Real Sociedad', 'Juventus FC', 'SS Lazio', 'Real Madrid',
       'Bayer 04 Leverkusen', 'Arsenal FC', 'Sevilla FC', 'SSC Napoli',
       'Leicester City', 'Everton FC', 'AC Milan', 'Villarreal CF',
       'ACF Fiorentina', 'RB Leipzig', 'AS Roma', 'Crystal Palace',
       'Wolverhampton Wanderers', 'Valencia CF', 'Leeds United',
       'LOSC Lille', 'Aston Villa', 'Olympique Lyon', 'OGC Nice',
       'AS Monaco', 'Atalanta BC', 'US Sassuolo', 'Torino FC',
       'Ajax Amsterdam', 'Brentford FC', 'Southampton FC',
       'Newcastle United', 'VfL Wolfsburg', 'FC Porto',
       'Olympique Marseille', 'Eintracht Frankfurt',
       'Borussia Mönchengladbach', 'Watford FC', 'Stade Rennais FC',
       'Clube de Regatas do Flamengo', 'VfB Stuttgart', 'Club Brugge KV',
       'Sporting CP', 'Brighton & Hove Albion', 'Dynamo Kyiv',
       'Athletic Bilbao', 'Real Betis Balompié', 'Zenit St. Petersburg',
       'Burnley FC', 'SL Benfica', 'TSG 1899 Hoffenheim', 'Norwich City',
       'PSV Eindhoven', 'KRC Genk', 'Club Atlético Vélez Sarsfield',
       'Club Atlético River Plate', 'FC Metz', 'UC Sampdoria',
       'Red Bull Salzburg', 'Bologna FC 1909', 'Shakhtar Donetsk',
       'Cagliari Calcio', 'Getafe CF', 'Al-Rayyan SC', 'Rubin Kazan',
       'Feyenoord Rotterdam', 'RCD Espanyol Barcelona', 'UD Almería',
       'Sheffield United', 'Celta de Vigo'], dtype=object)
```

[16]:
```
df['Club'].nunique()
```

[16]:
```
81
```

[17]:
```
df.Country.unique()
```

[17]:
```
array(['France', 'Norway', 'England', 'Egypt', 'Belgium', 'Brazil',
       'Netherlands', 'Portugal', 'Germany', 'Senegal', 'Korea, South',
       'Spain', 'Argentina', 'Canada', 'Morocco', 'Italy', 'Serbia',
       'Slovenia', 'Uruguay', 'Scotland', 'Nigeria', 'Slovakia', 'Poland',
```

```
       "Cote d'Ivoire", 'Austria', 'United States', 'Turkey', 'Mexico',
       'Croatia', 'Czech Republic', 'Algeria', 'Burkina Faso', 'Sweden',
       'Ghana', 'Denmark', 'Jamaica', 'Colombia', 'Guinea', 'Switzerland',
       'Ukraine', 'Russia', 'DR Congo', 'Hungary', 'Mali', 'Japan',
       'Cameroon', 'Iran', 'Montenegro', 'Gabon', 'Albania', 'Zambia',
       'The Gambia', 'Israel', 'Georgia', 'Venezuela', 'Wales', 'Peru'],
      dtype=object)
```

[18]:

```
df.Country.nunique()
```

[18]:

```
57
```

[19]:

```
df.Position.unique()
```

[19]:

```
array(['Centre-Forward', 'Left Winger', 'Right Winger',
       'Attacking Midfield', 'Central Midfield', 'Defensive Midfield',
       'Right-Back', 'Centre-Back', 'Second Striker', 'Left-Back',
       'Goalkeeper', 'Left Midfield', 'Right Midfield'], dtype=object)
```

[20]:

```
df.Position.nunique()
```

[20]:

```
13
```

[23]:

```
# Creating new categories for Position into 3␣
 ↪categories(attacker,midfielder,defender,goalkeeper)

df['New_Position'] = np.where((df['Position'] == 'Centre-Forward') |
                              (df['Position'] == 'Left Winger') |
                              (df['Position'] == 'Right Winger')|
                              (df['Position'] == 'Second Striker'),'Attacker',
                              np.where((df['Position'] == 'Attacking␣
 ↪Midfield') |
                                       (df['Position'] == 'Central Midfield')␣
 ↪|
                                       (df['Position'] == 'Defensive␣
 ↪Midfield') |
                                       (df['Position'] == 'Left Midfield') |
                                       (df['Position'] == 'Right Midfield'),␣
 ↪'Midfielder',
                                       np.where(df['Position'] ==␣
 ↪'Goalkeeper',
                                                'Goalkeeper','Defender')))
df
```

[23]:

```
                 Name           Position  Age  Market_value  Country  \
0           Kylian Mbappé      Centre-Forward   22         144.0   France
1           Erling Haaland     Centre-Forward   21         135.0   Norway
```

```
2                 Harry Kane      Centre-Forward   28         108.0  England
3              Jack Grealish         Left Winger   26          90.0  England
4              Mohamed Salah        Right Winger   29          90.0    Egypt
..                       …                   …   …            …       …
495  Giorgian de Arrascaeta  Attacking Midfield   27          16.2  Uruguay
496             Ayoze Pérez      Second Striker   28          16.2    Spain
497              Alex Meret          Goalkeeper   24          16.2    Italy
498         Duje Caleta-Car         Centre-Back   25          16.2  Croatia
499         Aritz Elustondo         Centre-Back   27          16.2    Spain

                           Club  Matches  Goals  Own_goals  Assists  \
0            Paris Saint-Germain       16      7          0       11
1              Borussia Dortmund       10     13          0        4
2              Tottenham Hotspur       16      7          0        2
3                Manchester City       15      2          0        3
4                   Liverpool FC       15     15          0        6
..                            …        …      …          …        …
495  Clube de Regatas do Flamengo        0      0          0        0
496               Leicester City        8      1          0        3
497                   SSC Napoli        5      0          0        0
498          Olympique Marseille        8      0          0        0
499                Real Sociedad       15      3          0        1

     Yellow_cards  Second_yellow_cards  Red_cards  Substitute_in  \
0               3                    0          0              0
1               1                    0          0              0
2               2                    0          0              2
3               1                    0          0              2
4               1                    0          0              0
..             …                    …          …              …
495             0                    0          0              0
496             0                    0          1              2
497             0                    0          0              0
498             2                    0          0              0
499             4                    0          0              1

     Substitute_out New_Position
0                 8     Attacker
1                 1     Attacker
2                 2     Attacker
3                 8     Attacker
4                 3     Attacker
..               …           …
495               0   Midfielder
496               5     Attacker
497               0   Goalkeeper
498               2     Defender
```

```
499                1     Defender

[500 rows x 16 columns]
```

[24]:
```
# converting Market_value from float to int

df['Market_value'] = df['Market_value'].astype(int)
df.Market_value.dtype
```

[24]:
```
dtype('int32')
```

[25]:
```
# creating new columns

df['Goals.per.game'] =  (df['Goals']/df['Matches']).round(2)
df['Assists.per.game'] =  (df['Assists']/df['Matches']).round(2)
df
```

[25]:
```
                         Name           Position  Age  Market_value  Country  \
0               Kylian Mbappé     Centre-Forward   22           144   France
1               Erling Haaland     Centre-Forward   21           135   Norway
2                  Harry Kane     Centre-Forward   28           108  England
3               Jack Grealish        Left Winger   26            90  England
4               Mohamed Salah       Right Winger   29            90    Egypt
..                        …                  …    …             …        …
495  Giorgian de Arrascaeta  Attacking Midfield   27            16  Uruguay
496                Ayoze Pérez      Second Striker   28            16    Spain
497                Alex Meret         Goalkeeper   24            16    Italy
498            Duje Caleta-Car        Centre-Back   25            16  Croatia
499           Aritz Elustondo        Centre-Back   27            16    Spain

                           Club  Matches  Goals  Own_goals  Assists  \
0             Paris Saint-Germain       16      7          0       11
1             Borussia Dortmund       10     13          0        4
2             Tottenham Hotspur       16      7          0        2
3               Manchester City       15      2          0        3
4                  Liverpool FC       15     15          0        6
..                          …        …      …          …        …
495  Clube de Regatas do Flamengo        0      0          0        0
496               Leicester City        8      1          0        3
497                  SSC Napoli        5      0          0        0
498           Olympique Marseille        8      0          0        0
499               Real Sociedad       15      3          0        1

     Yellow_cards  Second_yellow_cards  Red_cards  Substitute_in  \
0               3                    0          0              0
1               1                    0          0              0
2               2                    0          0              2
```

```
3              1                     0         0              2
4              1                     0         0              0
..           ...                   ...       ...            ...
495            0                     0         0              0
496            0                     0         1              2
497            0                     0         0              0
498            2                     0         0              0
499            4                     0         0              1

     Substitute_out New_Position  Goals.per.game  Assists.per.game
0                 8     Attacker            0.44              0.69
1                 1     Attacker            1.30              0.40
2                 2     Attacker            0.44              0.12
3                 8     Attacker            0.13              0.20
4                 3     Attacker            1.00              0.40
..              ...          ...             ...               ...
495               0   Midfielder             NaN               NaN
496               5     Attacker            0.12              0.38
497               0   Goalkeeper            0.00              0.00
498               2     Defender            0.00              0.00
499               1     Defender            0.20              0.07

[500 rows x 18 columns]
```

[26]:
```
df.isna().sum()
```

[26]:
```
Name                   0
Position               0
Age                    0
Market_value           0
Country                0
Club                   0
Matches                0
Goals                  0
Own_goals              0
Assists                0
Yellow_cards           0
Second_yellow_cards    0
Red_cards              0
Substitute_in          0
Substitute_out         0
New_Position           0
Goals.per.game         7
Assists.per.game       7
dtype: int64
```

[28]:
```
df.dropna(inplace=True)
df
```

[28]:
```
                 Name        Position  Age  Market_value  Country  \
0       Kylian Mbappé  Centre-Forward   22           144   France
1      Erling Haaland  Centre-Forward   21           135   Norway
2          Harry Kane  Centre-Forward   28           108  England
3       Jack Grealish     Left Winger   26            90  England
4       Mohamed Salah    Right Winger   29            90    Egypt
..                ...             ...  ...           ...      ...
494    Adam Armstrong  Centre-Forward   24            16  England
496       Ayoze Pérez  Second Striker   28            16    Spain
497        Alex Meret      Goalkeeper   24            16    Italy
498  Duje Caleta-Car      Centre-Back   25            16  Croatia
499  Aritz Elustondo      Centre-Back   27            16    Spain

                   Club  Matches  Goals  Own_goals  Assists  Yellow_cards  \
0   Paris Saint-Germain       16      7          0       11             3
1     Borussia Dortmund       10     13          0        4             1
2     Tottenham Hotspur       16      7          0        2             2
3       Manchester City       15      2          0        3             1
4          Liverpool FC       15     15          0        6             1
..                  ...      ...    ...        ...      ...           ...
494      Southampton FC       11      2          0        2             0
496      Leicester City        8      1          0        3             0
497          SSC Napoli        5      0          0        0             0
498  Olympique Marseille       8      0          0        0             2
499       Real Sociedad       15      3          0        1             4

     Second_yellow_cards  Red_cards  Substitute_in  Substitute_out  \
0                      0          0              0               8
1                      0          0              0               1
2                      0          0              2               2
3                      0          0              2               8
4                      0          0              0               3
..                   ...        ...            ...             ...
494                    0          0              1               2
496                    0          1              2               5
497                    0          0              0               0
498                    0          0              0               2
499                    0          0              1               1

    New_Position  Goals.per.game  Assists.per.game
0       Attacker            0.44              0.69
1       Attacker            1.30              0.40
2       Attacker            0.44              0.12
3       Attacker            0.13              0.20
```

```
4       Attacker             1.00              0.40
..          …              …               …
494     Attacker             0.18              0.18
496     Attacker             0.12              0.38
497   Goalkeeper             0.00              0.00
498     Defender             0.00              0.00
499     Defender             0.20              0.07

[493 rows x 18 columns]
```

5 3 Analysing Relationships

3.1 Analzing relationship between marketvalue and other continuous data like goals

[158]:
```
analysis=df.corr()
analysis["Market_value"].sort_values(ascending=False)
```

```
C:\Users\KEERTHAN\AppData\Local\Temp\ipykernel_7808\3132881492.py:1:
FutureWarning: The default value of numeric_only in DataFrame.corr is
deprecated. In a future version, it will default to False. Select only valid
columns or specify the value of numeric_only to silence this warning.
  analysis=df.corr()
```

[158]:
```
Market_value            1.000000
Assists                 0.225424
Goals.per.game          0.224564
Goals                   0.209460
Assists.per.game        0.201473
Matches                 0.103294
Age                     0.049173
Second_yellow_cards     0.040727
Red_cards              -0.000039
Yellow_cards           -0.002807
Substitute_out         -0.004813
Own_goals              -0.034190
Substitute_in          -0.091397
Name: Market_value, dtype: float64
```

so we see how Age,goals,assists and all have positive impact and red cards and all having negative impact

3.2 Checking relation between market value and different countries whether the market value varies in countries using annova

[172]:
```
unique_countries = df['Country'].unique()
max_length = df.groupby('Country')['Market_value'].count().max()
df_country_market_values = pd.DataFrame(0.0, columns=unique_countries,␣
↪index=range(max_length))
for country in unique_countries:
```

```python
    market_values = df[df['Country'] == country]['Market_value']
    df_country_market_values.loc[:len(market_values) - 1, country] = market_values.values

print(df_country_market_values)
```

```
    France  Norway  England  Egypt  Belgium  Brazil  Netherlands  Portugal  \
0    144.0   135.0    108.0   90.0     90.0    90.0         81.0      81.0
1     63.0    36.0     90.0    0.0     90.0    67.0         63.0      67.0
2     54.0    16.0     81.0    0.0     54.0    63.0         49.0      63.0
3     54.0     0.0     81.0    0.0     49.0    54.0         45.0      63.0
4     54.0     0.0     76.0    0.0     36.0    54.0         40.0      49.0
..     …       …      …      …      …      …          …        …
62     0.0     0.0     16.0    0.0      0.0     0.0          0.0       0.0
63     0.0     0.0     16.0    0.0      0.0     0.0          0.0       0.0
64     0.0     0.0     16.0    0.0      0.0     0.0          0.0       0.0
65     0.0     0.0     16.0    0.0      0.0     0.0          0.0       0.0
66     0.0     0.0     16.0    0.0      0.0     0.0          0.0       0.0

    Germany  Senegal  …  Montenegro  Gabon  Albania  Zambia  The Gambia  \
0      81.0     76.0  …        22.0   22.0     19.0    19.0        18.0
1      63.0     43.0  …         0.0    0.0     19.0    18.0         0.0
2      63.0     27.0  …         0.0    0.0      0.0     0.0         0.0
3      63.0     24.0  …         0.0    0.0      0.0     0.0         0.0
4      58.0     22.0  …         0.0    0.0      0.0     0.0         0.0
..      …       … …         …    …     …     …        …
62      0.0      0.0  …         0.0    0.0      0.0     0.0         0.0
63      0.0      0.0  …         0.0    0.0      0.0     0.0         0.0
64      0.0      0.0  …         0.0    0.0      0.0     0.0         0.0
65      0.0      0.0  …         0.0    0.0      0.0     0.0         0.0
66      0.0      0.0  …         0.0    0.0      0.0     0.0         0.0

    Israel  Georgia  Venezuela  Wales  Peru
0     18.0     16.0       16.0   16.0  16.0
1      0.0      0.0        0.0    0.0   0.0
2      0.0      0.0        0.0    0.0   0.0
3      0.0      0.0        0.0    0.0   0.0
4      0.0      0.0        0.0    0.0   0.0
..     …      …        …    …   …
62     0.0      0.0        0.0    0.0   0.0
63     0.0      0.0        0.0    0.0   0.0
64     0.0      0.0        0.0    0.0   0.0
65     0.0      0.0        0.0    0.0   0.0
66     0.0      0.0        0.0    0.0   0.0

[67 rows x 56 columns]
```

[174]:
```python
from scipy.stats import f_oneway

# Perform ANOVA
f_stat, p_value = f_oneway(*[df_country_market_values[country].dropna() for␣
↪country in df_country_market_values.columns])
if p_value < 0.05:
    print("There are significant differences in market value among different␣
↪countries,so analyzing based on countries is beneficial")
else:
    print("There might not be significant differences in market value among␣
↪different countries.")
```

```
There are significant differences in market value among different countries,so
analyzing based on countries is beneficial
```

3.3 Checking relation between market value and different Positions whether the market value varies in Positions using annova

[179]:
```python
unique_positions = df['Position'].unique()
positions_market_values = {position: [0.0] * len(df) for position in␣
↪unique_positions}
df_positions_market_values = pd.DataFrame(positions_market_values,␣
↪columns=unique_positions)

for position in unique_positions:
    market_values = df[df['Position'] == position]['Market_value']
    df_positions_market_values[position][:len(market_values)] = market_values.
↪values

# Perform ANOVA
f_stat_pos, p_value_pos = f_oneway(*[df_positions_market_values[position].
↪dropna() for position in df_positions_market_values.columns])

if p_value_pos < 0.05:
    print("There are significant differences in market value among different␣
↪player positions,So it is beneficial to analyse based on positions")
else:
    print("There might not be significant differences in market value among␣
↪different player positions.")
```

```
There are significant differences in market value among different player
positions,So it is beneficial to analyse based on positions
```

3.4 Checking relation between market value and different Clubs whether the market value varies in Positions using annova

[181]:
```python
unique_clubs = df['Club'].unique()
clubs_market_values = {club: [0.0] * len(df) for club in unique_clubs}
df_clubs_market_values = pd.DataFrame(clubs_market_values, columns=unique_clubs)
```

```
# Assign market values to respective columns for each club
for club in unique_clubs:
    market_values = df[df['Club'] == club]['Market_value']
    df_clubs_market_values[club][:len(market_values)] = market_values.values

# Perform ANOVA
f_stat_clubs, p_value_clubs = f_oneway(*[df_clubs_market_values[club].dropna()␣
↪for club in df_clubs_market_values.columns])

# Check for significance level (usually 0.05)
if p_value_clubs < 0.05:
    print("There are significant differences in market value among different␣
↪clubs,So it is beneficial to analyse based on Different clubs")
else:
    print("There might not be significant differences in market value among␣
↪different clubs.")
```

```
There are significant differences in market value among different clubs,So it is
beneficial to analyse based on Different clubs
```

6 4. Market Value x Categories

6.1 4.1 Position

[29]:
```
df
```

[29]:
```
                 Name        Position  Age  Market_value  Country  \
0       Kylian Mbappé  Centre-Forward   22           144   France
1      Erling Haaland  Centre-Forward   21           135   Norway
2          Harry Kane  Centre-Forward   28           108  England
3       Jack Grealish     Left Winger   26            90  England
4       Mohamed Salah    Right Winger   29            90    Egypt
..                ...             ...  ...           ...      ...
494     Adam Armstrong  Centre-Forward   24            16  England
496        Ayoze Pérez  Second Striker   28            16    Spain
497         Alex Meret      Goalkeeper   24            16    Italy
498   Duje Caleta-Car     Centre-Back   25            16  Croatia
499   Aritz Elustondo     Centre-Back   27            16    Spain

                    Club  Matches  Goals  Own_goals  Assists  Yellow_cards  \
0    Paris Saint-Germain       16      7          0       11             3
1      Borussia Dortmund       10     13          0        4             1
2      Tottenham Hotspur       16      7          0        2             2
3        Manchester City       15      2          0        3             1
4           Liverpool FC       15     15          0        6             1
..                   ...      ...    ...        ...      ...           ...
```

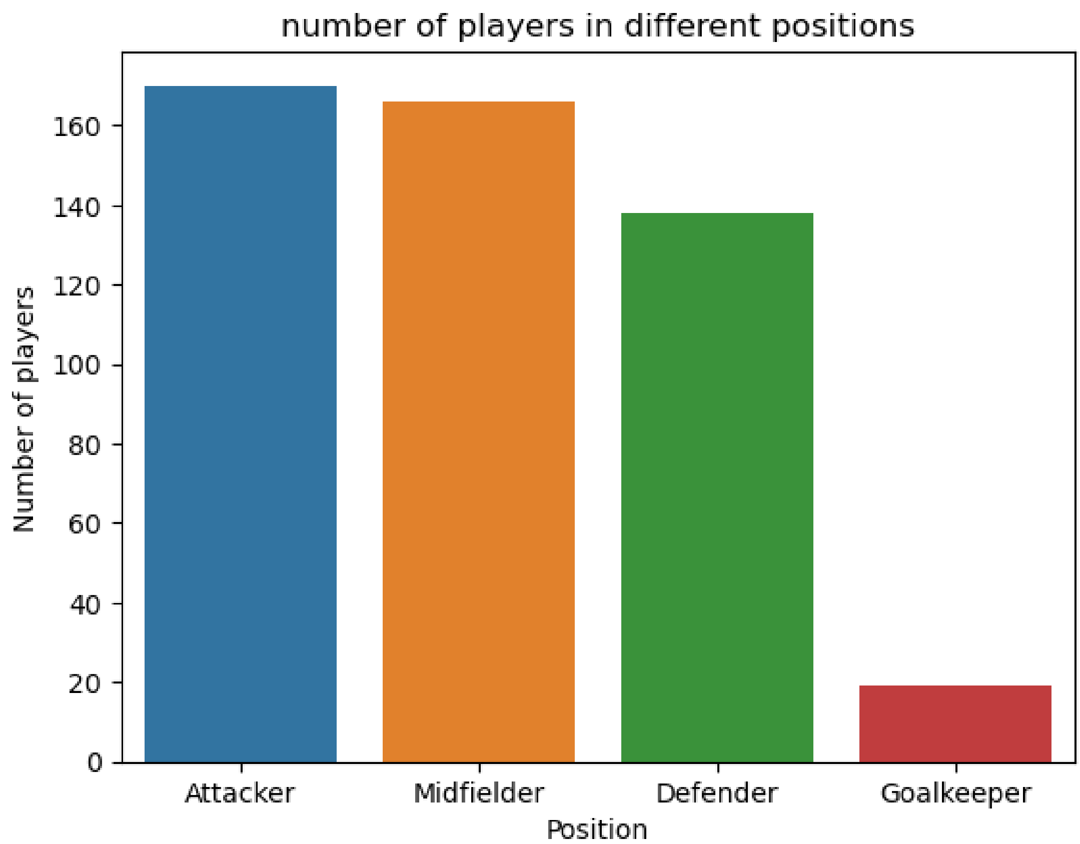
```
494       Southampton FC       11       2          0         2             0
496       Leicester City        8       1          0         3             0
497           SSC Napoli        5       0          0         0             0
498  Olympique Marseille        8       0          0         0             2
499         Real Sociedad      15       3          0         1             4

     Second_yellow_cards  Red_cards  Substitute_in  Substitute_out  \
0                      0          0              0               8
1                      0          0              0               1
2                      0          0              2               2
3                      0          0              2               8
4                      0          0              0               3
..                   …          …              …               …
494                    0          0              1               2
496                    0          1              2               5
497                    0          0              0               0
498                    0          0              0               2
499                    0          0              1               1

    New_Position  Goals.per.game  Assists.per.game
0       Attacker            0.44              0.69
1       Attacker            1.30              0.40
2       Attacker            0.44              0.12
3       Attacker            0.13              0.20
4       Attacker            1.00              0.40
..           …               …                 …
494     Attacker            0.18              0.18
496     Attacker            0.12              0.38
497   Goalkeeper            0.00              0.00
498     Defender            0.00              0.00
499     Defender            0.20              0.07

[493 rows x 18 columns]
```

[30]:
```
df["New_Position"].value_counts()
```

[30]:
```
Attacker      170
Midfielder    166
Defender      138
Goalkeeper     19
Name: New_Position, dtype: int64
```

[33]:
```
#plotting the graph of number of players in different positions
sns.countplot(x="New_Position",data=df)
plt.title("number of players in different positions")
plt.xlabel("Position")
plt.ylabel("Number of players")
```

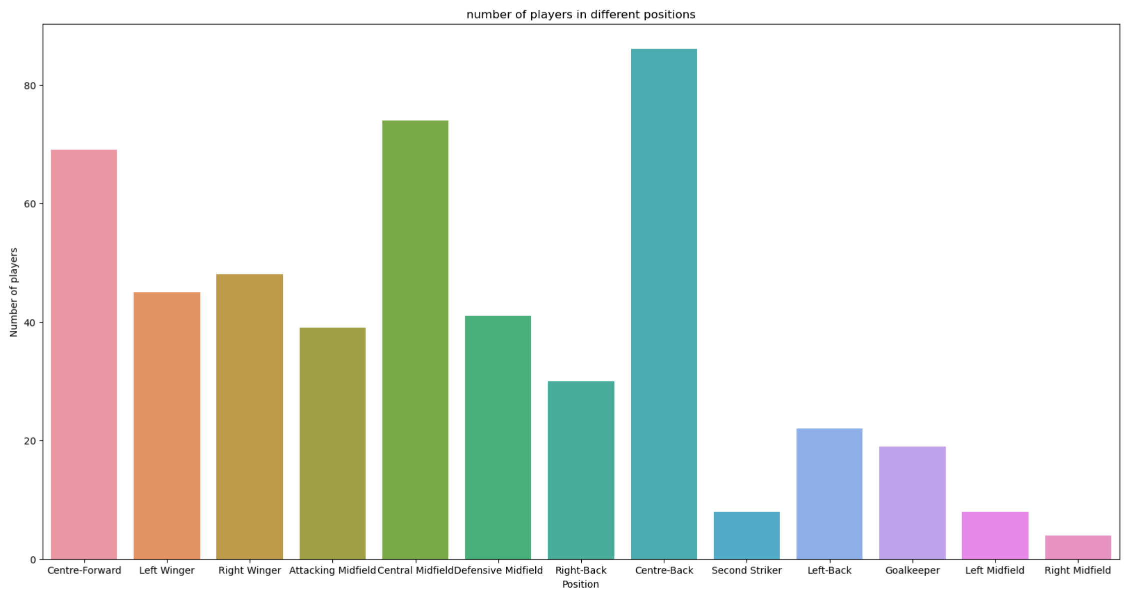
```
plt.show()
```



[34]:
```
df["Position"].value_counts()
```

[34]:
```
Centre-Back           86
Central Midfield      74
Centre-Forward        69
Right Winger          48
Left Winger           45
Defensive Midfield    41
Attacking Midfield    39
Right-Back            30
Left-Back             22
Goalkeeper            19
Second Striker         8
Left Midfield          8
Right Midfield         4
Name: Position, dtype: int64
```

[38]:
```
#plotting the graph of number of players in different positions including all␣
↪positions
plt.figure(figsize=(20,10))
sns.countplot(x="Position",data=df)
plt.title("number of players in different positions")
plt.xlabel("Position")
plt.ylabel("Number of players")
plt.show()
```



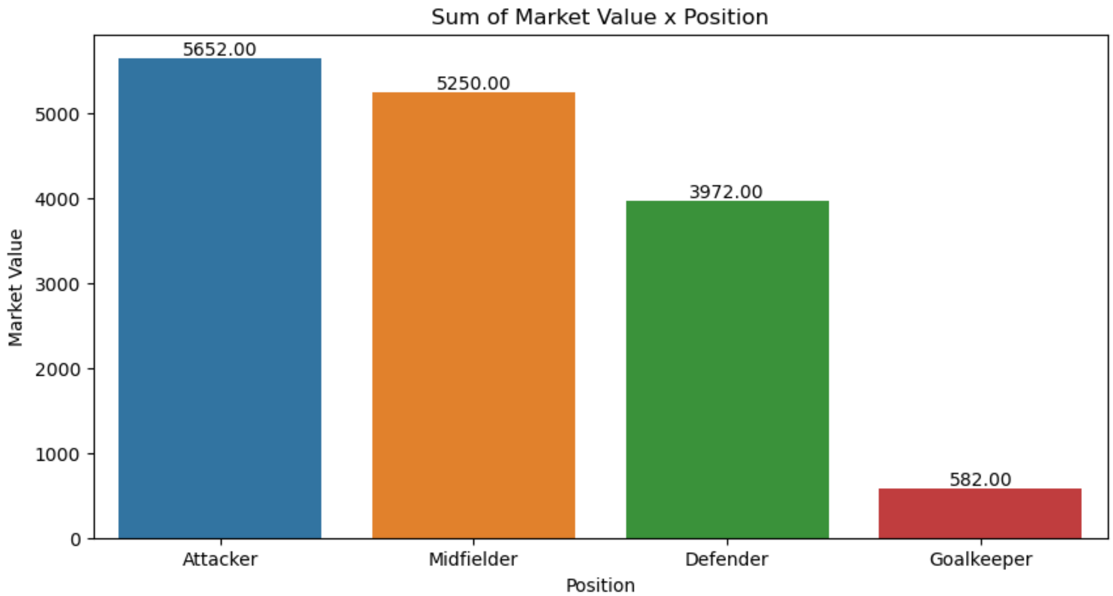
[49]:
```
position1=df.groupby(by="New_Position").Market_value.sum().
↪reset_index(name="Market value based on positions")
position1.sort_values(by="Market value based on␣
↪positions",ascending=False,inplace=True)
position1
```

[49]:

	New_Position	Market value based on positions
0	Attacker	5652
3	Midfielder	5250
1	Defender	3972
2	Goalkeeper	582

[55]:
```
#Visualizing for new categorized positions
plt.figure(figsize=(10, 5))
plt.title("Sum of Market Value x Position")
ax = sns.barplot(x="New_Position", y="Market value based on positions",␣
↪data=position1)
for p in ax.patches:
```

```
    ax.annotate(format(p.get_height(), '.2f'),
                (p.get_x() + p.get_width() / 2., p.get_height()),
                ha = 'center', va = 'center',
                xytext = (0, 5),
                textcoords = 'offset points')

plt.xlabel("Position")
plt.ylabel("Market Value")
plt.show()
```



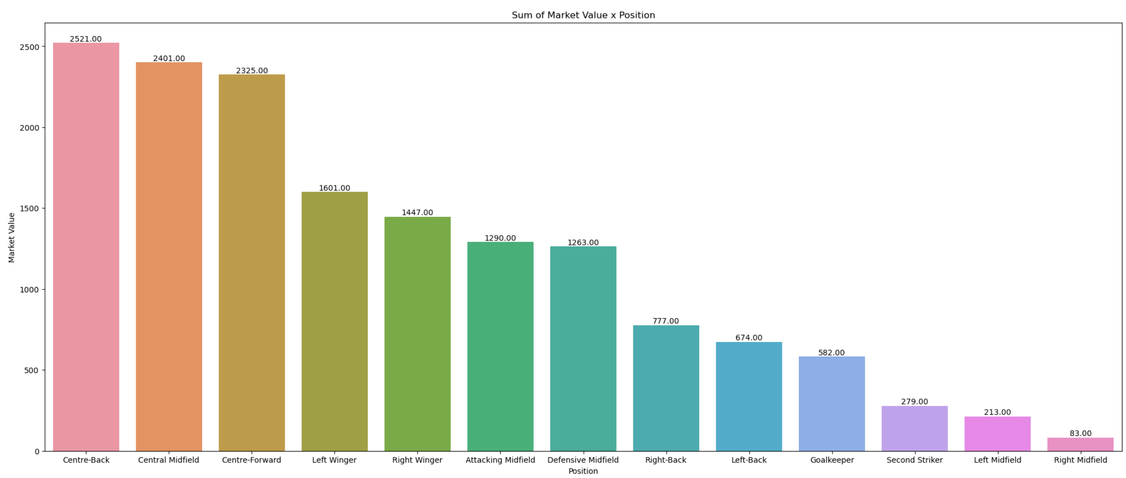
[56]:
```
position2=df.groupby(by="Position").Market_value.sum().reset_index(name="Market␣
 ↪value based on positions")
position2.sort_values(by="Market value based on␣
 ↪positions",ascending=False,inplace=True)
position2
```

[56]:

	Position	Market value based on positions
2	Centre-Back	2521
1	Central Midfield	2401
3	Centre-Forward	2325
7	Left Winger	1601
10	Right Winger	1447
0	Attacking Midfield	1290
4	Defensive Midfield	1263
11	Right-Back	777
8	Left-Back	674
5	Goalkeeper	582

```
12      Second Striker                                   279
6        Left Midfield                                   213
9       Right Midfield                                    83
```

[58]:
```
#Visualing for the positions
plt.figure(figsize=(25, 10))
plt.title("Sum of Market Value x Position")
ax = sns.barplot(x="Position", y="Market value based on positions",␣
 ↪data=position2)
for p in ax.patches:
    ax.annotate(format(p.get_height(), '.2f'),
                (p.get_x() + p.get_width() / 2., p.get_height()),
                ha = 'center', va = 'center',
                xytext = (0, 5),
                textcoords = 'offset points')

plt.xlabel("Position")
plt.ylabel("Market Value")
plt.show()
```

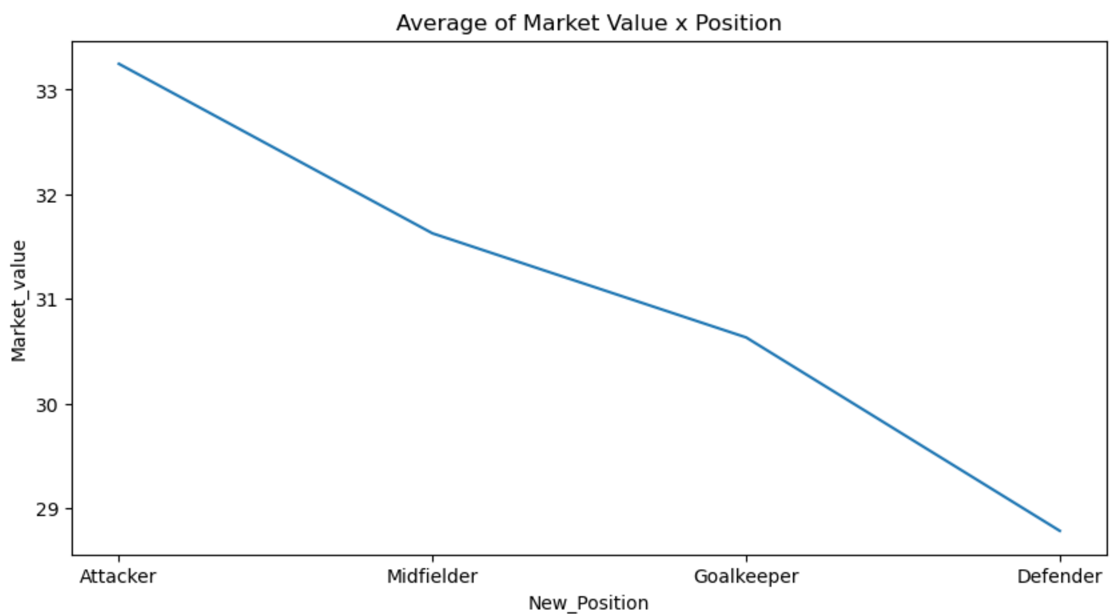


[59]:
```
y = df.groupby('New_Position').Market_value.mean().sort_values(ascending=False)
y
```

[59]:
```
New_Position
Attacker      33.247059
Midfielder    31.626506
Goalkeeper    30.631579
Defender      28.782609
Name: Market_value, dtype: float64
```

[60]:
```
# visualizing

plt.figure(figsize=(10,5))

plt.title("Average of Market Value x Position")

sns.lineplot(data=y)
```

[60]:
```
<Axes: title={'center': 'Average of Market Value x Position'},
xlabel='New_Position', ylabel='Market_value'>
```



[61]:
```
position2
```

[61]:
```
              Position  Market value based on positions
2          Centre-Back                             2521
1     Central Midfield                             2401
3       Centre-Forward                             2325
7          Left Winger                             1601
10        Right Winger                             1447
0   Attacking Midfield                             1290
4   Defensive Midfield                             1263
11          Right-Back                              777
8            Left-Back                              674
5           Goalkeeper                              582
12      Second Striker                              279
6        Left Midfield                              213
9       Right Midfield                               83
```

[63]:
```
df.groupby('Position').Market_value.agg(['max', 'min']).sort_values(by=['max',␣
↪'min'], ascending = False)
```

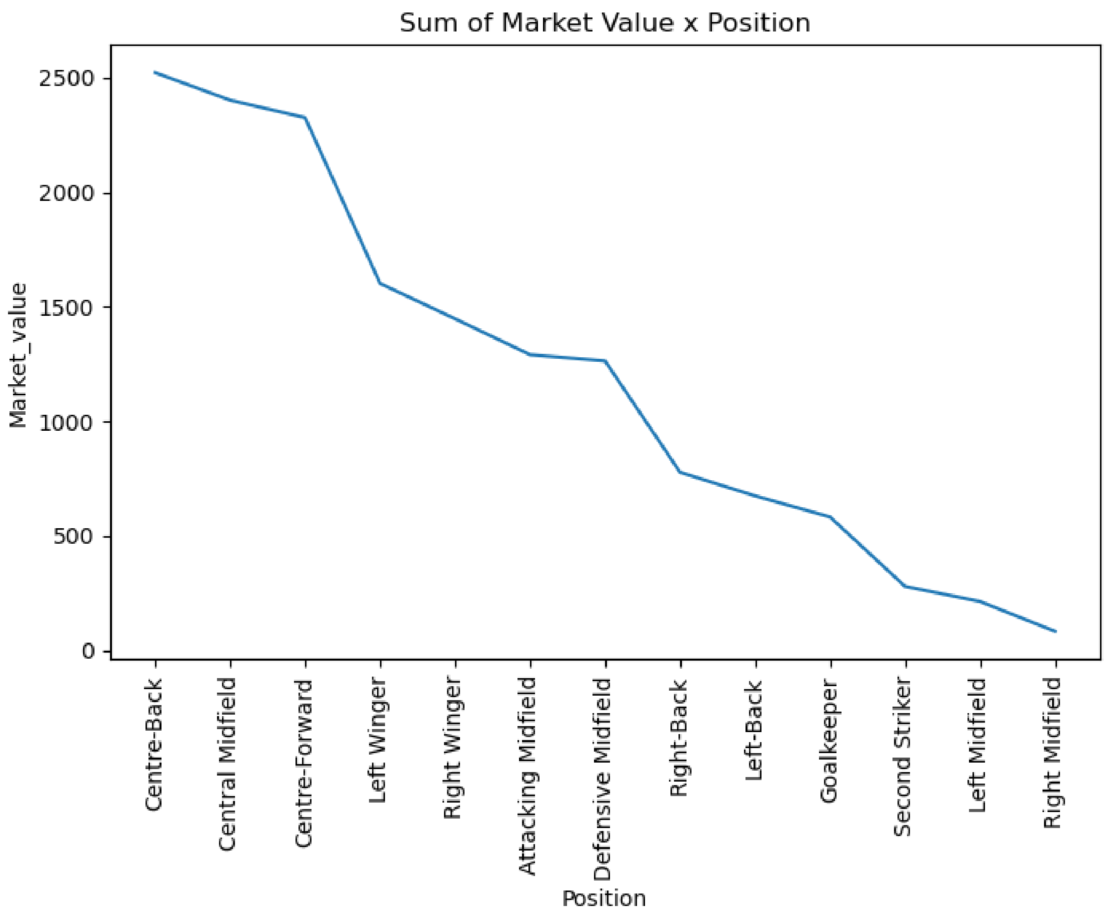
[63]:
```
                    max  min
Position
Centre-Forward      144   16
Attacking Midfield   90   18
Left Winger          90   16
Right Winger         90   16
Central Midfield     81   16
Defensive Midfield   81   16
Centre-Back          67   16
Right-Back           67   16
Goalkeeper           63   16
Left-Back            63   16
Second Striker       63   16
Left Midfield        58   16
Right Midfield       27   16
```

[66]:
```
c=df.groupby('Position').Market_value.sum().sort_values(ascending = False)
c
```

[66]:
```
Position
Centre-Back           2521
Central Midfield      2401
Centre-Forward        2325
Left Winger           1601
Right Winger          1447
Attacking Midfield    1290
Defensive Midfield    1263
Right-Back             777
Left-Back              674
Goalkeeper             582
Second Striker         279
Left Midfield          213
Right Midfield          83
Name: Market_value, dtype: int32
```

[67]:
```
# visualizing

plt.figure(figsize=(8,5))

plt.title("Sum of Market Value x Position")

plt.xticks(rotation=90)

sns.lineplot(data=c)
```

[67]: <Axes: title={'center': 'Sum of Market Value x Position'}, xlabel='Position', ylabel='Market_value'>

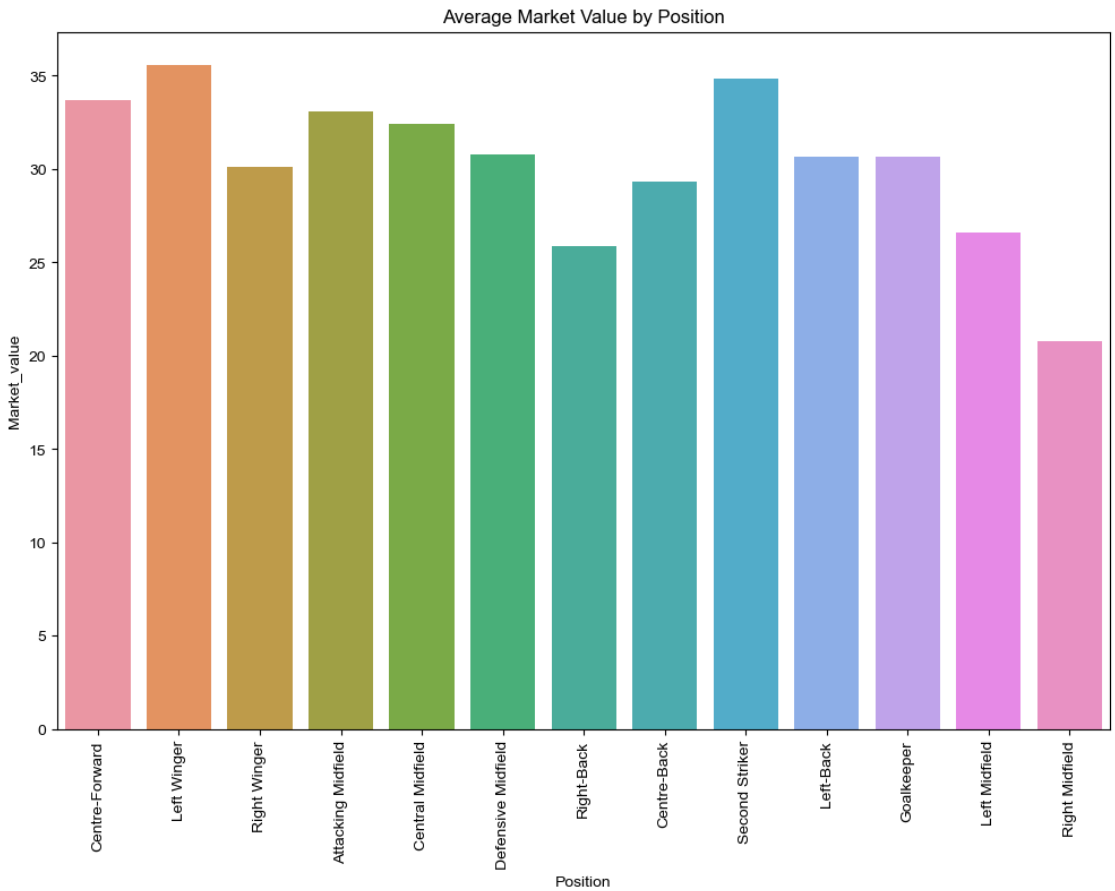


[69]:
```
df.groupby("Position").Market_value.mean().sort_values(ascending=False)
```

[69]:
```
Position
Left Winger           35.577778
Second Striker        34.875000
Centre-Forward        33.695652
Attacking Midfield    33.076923
Central Midfield      32.445946
Defensive Midfield    30.804878
Left-Back             30.636364
Goalkeeper            30.631579
Right Winger          30.145833
Centre-Back           29.313953
Left Midfield         26.625000
Right-Back            25.900000
```

```
Right Midfield        20.750000
Name: Market_value, dtype: float64
```

[71]:
```
plt.figure(figsize=(12, 8))
sns.barplot(data=df, x='Position', y='Market_value',errorbar=None)
plt.title('Average Market Value by Position')
sns.set_style("white")
plt.xticks(rotation=90)
plt.show()
```



6.1.1 Insights

1.Analyzing by general positions, we notice that the dataset has a higher number of forwards, followed by midfielders and defenders. The goalkeepers appear in a smaller quantity.

2.We can also observe that forwards are more valued, followed by midfielders and defenders. Goalkeepers have a significantly lower total market value. However, when we look at the average, defenders are below goalkeepers.

3.Analyzing specific positions within forwards, midfielders, and defenders, we notice that we have a large number of center-backs, central midfielders, and center-forwards in the dataset. Due to the higher quantity, these positions also have the highest sum of market value.

However, when we consider the average, left-wingers, second strikers, and center-forwards have the highest value index.

6.2 4.2 Country

[72]:
```
df
```

[72]:
```
                 Name        Position  Age  Market_value  Country  \
0       Kylian Mbappé  Centre-Forward   22           144   France
1      Erling Haaland  Centre-Forward   21           135   Norway
2          Harry Kane  Centre-Forward   28           108  England
3       Jack Grealish     Left Winger   26            90  England
4       Mohamed Salah    Right Winger   29            90    Egypt
..                ...             ...  ...           ...      ...
494    Adam Armstrong  Centre-Forward   24            16  England
496       Ayoze Pérez  Second Striker   28            16    Spain
497       Alex Meret      Goalkeeper   24            16    Italy
498  Duje Caleta-Car     Centre-Back   25            16  Croatia
499  Aritz Elustondo     Centre-Back   27            16    Spain

                    Club  Matches  Goals  Own_goals  Assists  Yellow_cards  \
0    Paris Saint-Germain       16      7          0       11             3
1      Borussia Dortmund       10     13          0        4             1
2      Tottenham Hotspur       16      7          0        2             2
3        Manchester City       15      2          0        3             1
4           Liverpool FC       15     15          0        6             1
..                   ...      ...    ...        ...      ...           ...
494       Southampton FC       11      2          0        2             0
496       Leicester City        8      1          0        3             0
497          SSC Napoli         5      0          0        0             0
498  Olympique Marseille        8      0          0        0             2
499        Real Sociedad       15      3          0        1             4

     Second_yellow_cards  Red_cards  Substitute_in  Substitute_out  \
0                      0          0              0               8
1                      0          0              0               1
2                      0          0              2               2
3                      0          0              2               8
4                      0          0              0               3
..                   ...        ...            ...             ...
494                    0          0              1               2
496                    0          1              2               5
497                    0          0              0               0
498                    0          0              0               2
499                    0          0              1               1
```

```
    New_Position  Goals.per.game  Assists.per.game
0       Attacker            0.44              0.69
1       Attacker            1.30              0.40
2       Attacker            0.44              0.12
3       Attacker            0.13              0.20
4       Attacker            1.00              0.40
..           …               …                 …
494     Attacker            0.18              0.18
496     Attacker            0.12              0.38
497   Goalkeeper            0.00              0.00
498     Defender            0.00              0.00
499     Defender            0.20              0.07

[493 rows x 18 columns]
```

[81]:
```python
#number of players playing for different countries
df["Country"].value_counts()
```

[81]:
```
England          67
France           56
Spain            52
Brazil           40
Germany          29
Portugal         25
Italy            25
Argentina        22
Netherlands      17
Belgium          14
Uruguay          10
Cote d'Ivoire     8
Croatia           8
Denmark           7
Nigeria           7
Colombia          7
Switzerland       6
Turkey            6
Austria           6
Senegal           6
Scotland          5
Poland            5
United States     5
Serbia            5
Algeria           4
Sweden            4
Morocco           4
Mexico            3
```

```
Norway             3
Ukraine            3
Japan              2
Mali               2
Albania            2
Zambia             2
Guinea             2
Canada             2
Burkina Faso       2
Czech Republic     2
Jamaica            1
Gabon              1
Wales              1
Venezuela          1
Georgia            1
Israel             1
The Gambia         1
Korea, South       1
Slovenia           1
Montenegro         1
Ghana              1
Iran               1
Cameroon           1
Slovakia           1
Egypt              1
Hungary            1
Russia             1
Peru               1
Name: Country, dtype: int64
```

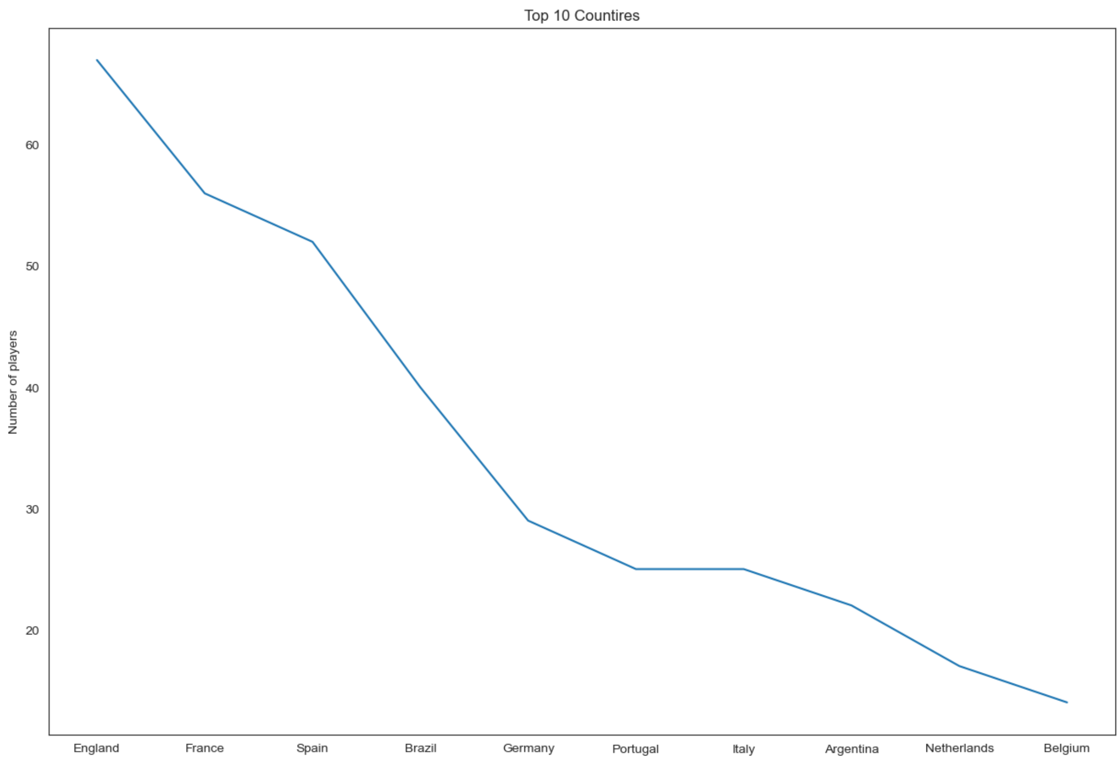
[94]:
```
#Top 10 countries
country1=pd.DataFrame(df["Country"].value_counts().head(10))
country1.rename(columns={"Country":"Number of players"},inplace=True)
country1
```

[94]:

	Number of players
England	67
France	56
Spain	52
Brazil	40
Germany	29
Portugal	25
Italy	25
Argentina	22
Netherlands	17
Belgium	14

[99]:
```python
#Visualizing top 10 countires
plt.figure(figsize=(15,10))
plt.title("Top 10 Countires")
sns.lineplot(data=country1["Number of players"])
```

[99]: <Axes: title={'center': 'Top 10 Countires'}, ylabel='Number of players'>



[100]:
```python
df.groupby('Country').Market_value.agg(['max', 'min']).sort_values(by=['max',␣
↪'min'], ascending = False).head(10)
```

[100]:

Country	max	min
France	144	16
Norway	135	16
England	108	16
Egypt	90	90
Belgium	90	16
Brazil	90	16
Netherlands	81	19
Germany	81	16
Portugal	81	16
Korea, South	76	76

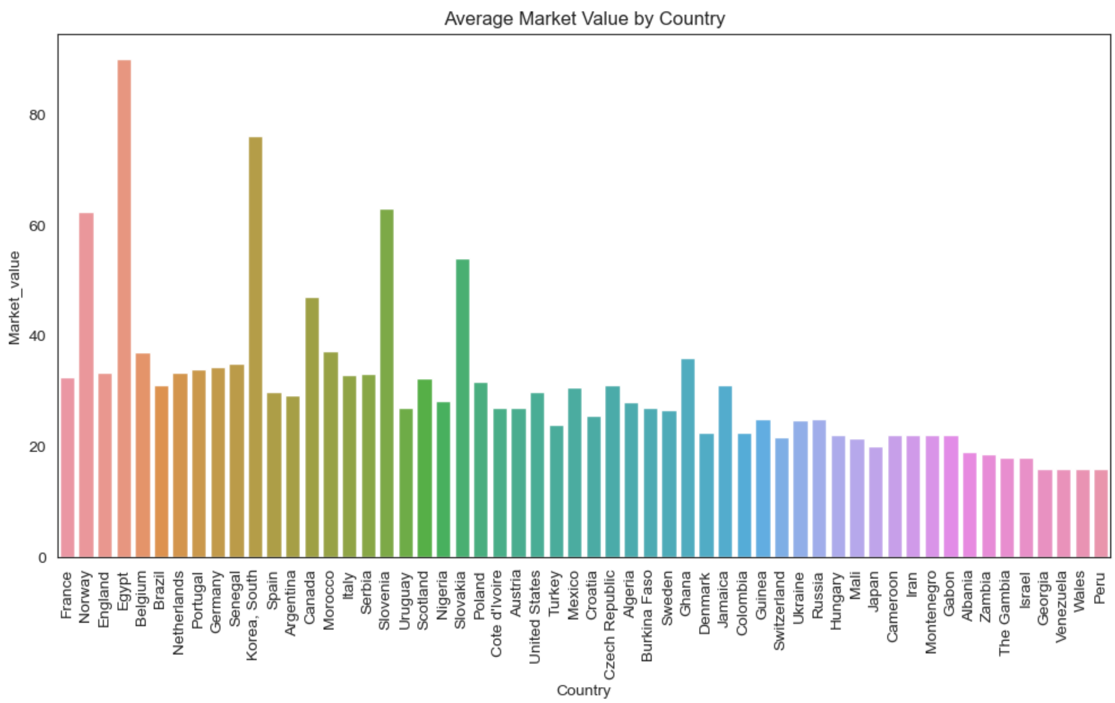
[101]:
```
df.groupby('Country').Market_value.mean().sort_values(ascending = False).
↪head(10)
```

[101]:
```
Country
Egypt           90.000000
Korea, South    76.000000
Slovenia        63.000000
Norway          62.333333
Slovakia        54.000000
Canada          47.000000
Morocco         37.250000
Belgium         37.071429
Ghana           36.000000
Senegal         35.000000
Name: Market_value, dtype: float64
```

[103]:
```
selected_countries = ['Slovenia', 'Norway', 'Slovakia', 'Canada', 'Marocco',␣
↪'Belgium', 'Ghana', 'Senegal']
counts = df[df['Country'].isin(selected_countries)]['Country'].value_counts()
counts
```

[103]:
```
Belgium     14
Senegal      6
Norway       3
Canada       2
Slovenia     1
Slovakia     1
Ghana        1
Name: Country, dtype: int64
```

[105]:
```
# visualizing
plt.figure(figsize=(12, 6))
sns.barplot(data=df, x='Country', y='Market_value',errorbar=None)
plt.title('Average Market Value by Country')
plt.xticks(rotation=90)
plt.show()
```



6.2.1 Insights

1.England has the highest number of players in the dataset, followed by France, Spain, Brazil, and Germany. These same five countries have the highest total market values due to having more players on the list.

2.France and Norway have the highest maximum value, as they have the two most valued players. Egypt has the same maximum and minimum value, along with the highest average value, because the country has only one player in the list (Salah).

3.Other countries have higher averages because they have few players in the database, such as South Korea, Norway, Slovakia, and Slovenia.

6.3 4.3 Age

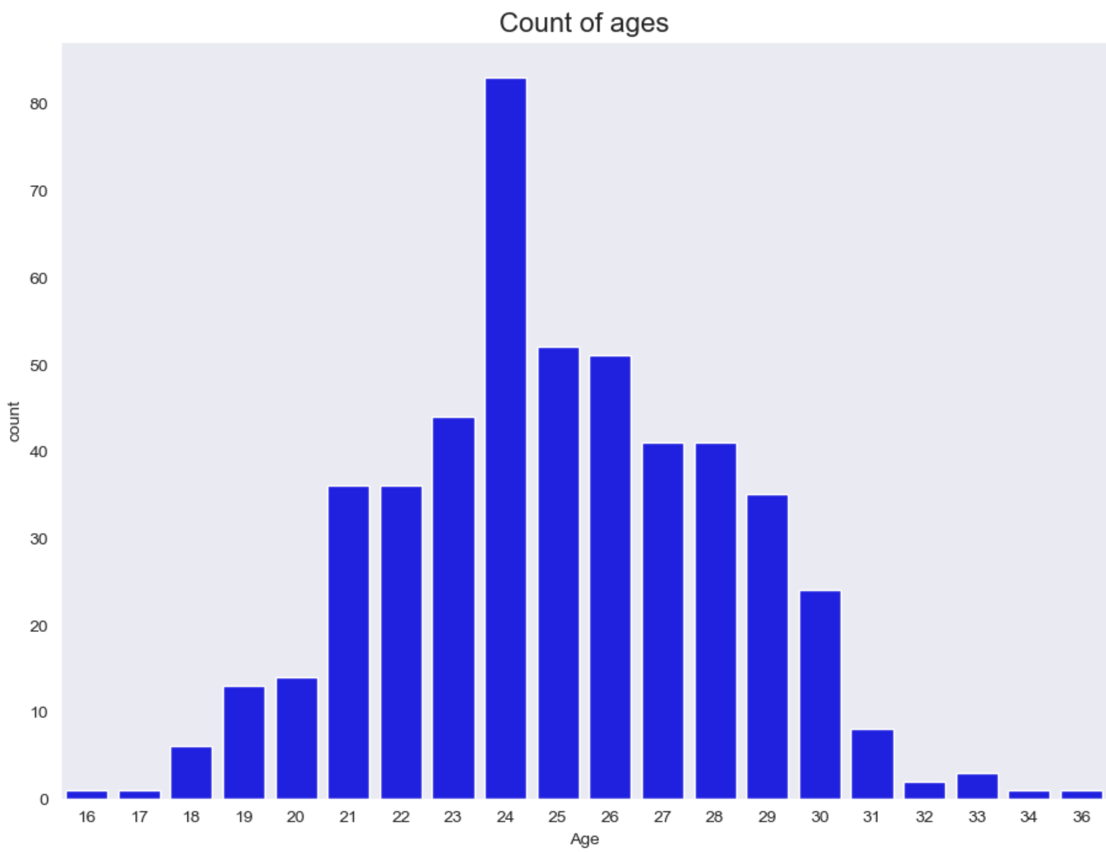
[107]:
```
df.Age.value_counts()
```

[107]:
```
24    83
25    52
26    51
23    44
27    41
28    41
21    36
22    36
```

```
29    35
30    24
20    14
19    13
31     8
18     6
33     3
32     2
34     1
36     1
17     1
16     1
Name: Age, dtype: int64
```

[109]:
```
plt.figure(figsize=(11,8))

sns.set_style("dark")

plt.title("Count of ages", fontsize=16)

plt.xlabel=("Age")

sns.countplot(data=df, x="Age", color='blue')
```

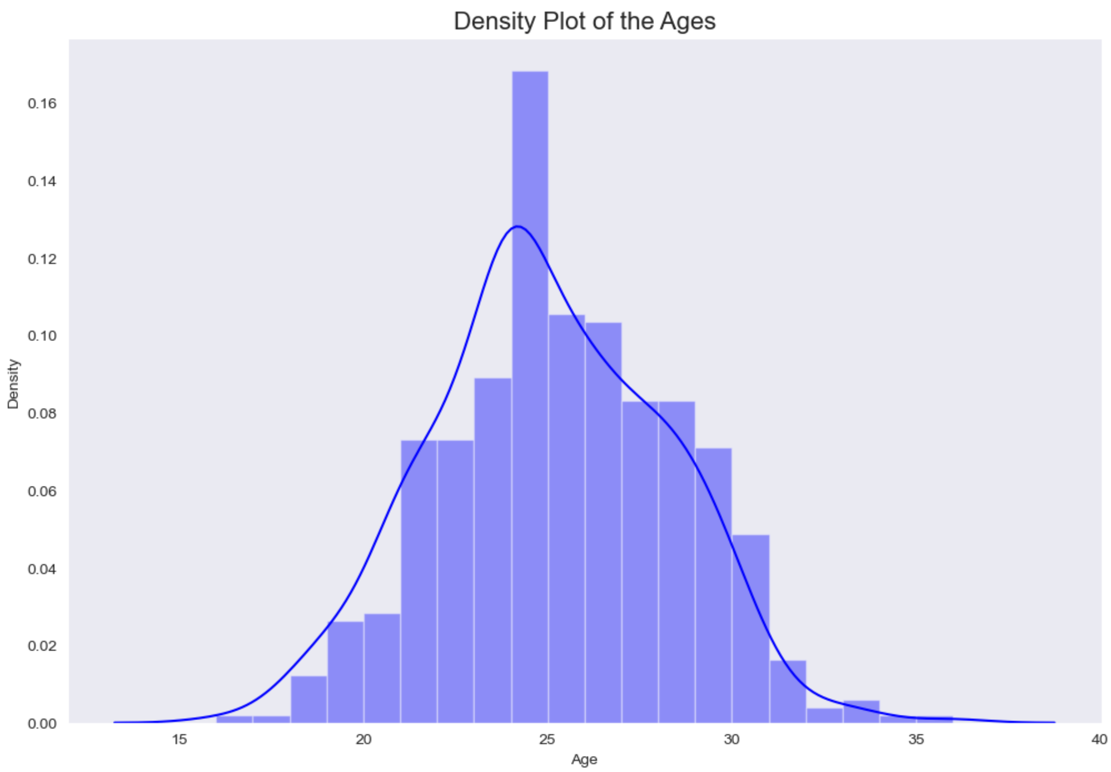
[109]:
```
<Axes: title={'center': 'Count of ages'}, xlabel='Age', ylabel='count'>
```



[111]:
```
plt.figure(figsize=(12, 8))
sns.distplot(df['Age'], hist=True, color='Blue')
plt.title("Density Plot of the Ages", fontsize=16)
```

```
C:\Users\KEERTHAN\AppData\Local\Temp\ipykernel_7808\1600338225.py:2:
UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['Age'], hist=True, color='Blue')
```

[111]:
```
Text(0.5, 1.0, 'Density Plot of the Ages')
```



[112]:
```
# age x market value sum

df.groupby('Age').Market_value.sum().sort_values(ascending=False).head(10)
```

[112]:
```
Age
24    2526
26    1694
25    1386
28    1357
29    1350
22    1275
27    1245
23    1231
21    1154
30     729
Name: Market_value, dtype: int32
```

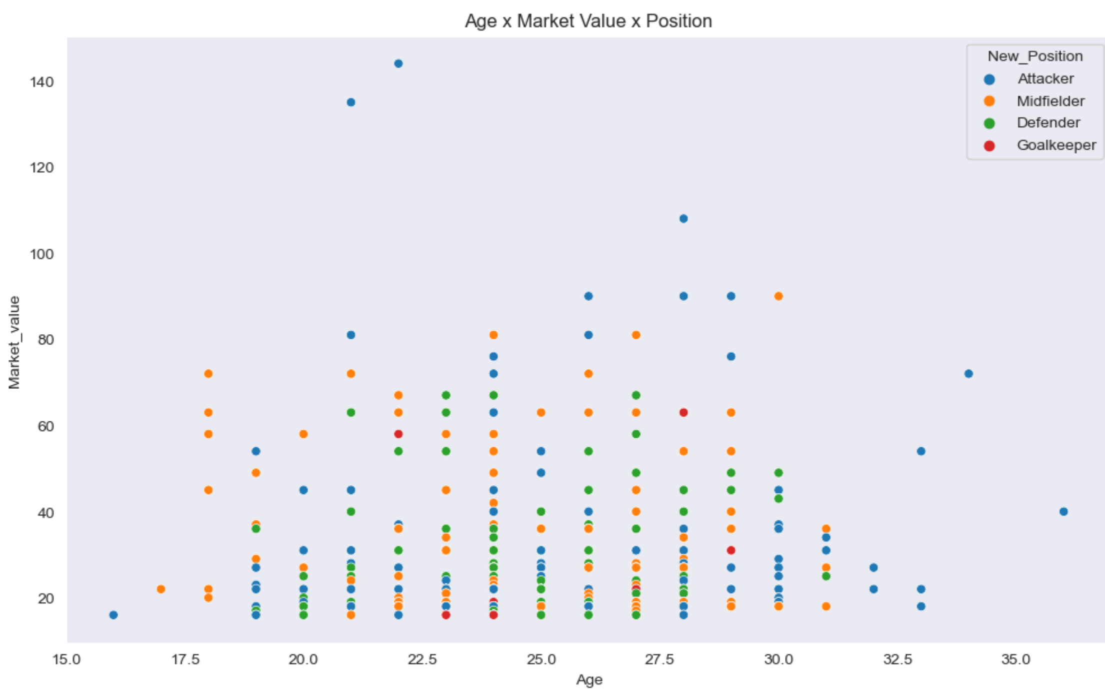
[114]:
```
# age x market value average

df.groupby('Age').Market_value.mean().sort_values(ascending=False).head(10)
```

[114]: Age

```
34    72.000000
18    46.666667
36    40.000000
29    38.571429
22    35.416667
26    33.215686
28    33.097561
21    32.055556
33    31.333333
24    30.433735
Name: Market_value, dtype: float64
```

[116]:

```
plt.figure(figsize=(12,7))

plt.title("Age x Market Value x Position")

sns.scatterplot(data=df, x='Age', y='Market_value', hue='New_Position')

sns.set_style("darkgrid")
```



6.3.1 Insights

1.Most of the players in the dataset are in the age range of 20 to 30 years.

2.We have a higher number of players aged 24 (83 players), followed by 25 years (53 players), and 26 years (51 players). Due to the larger number of players in these age groups, they have accumulated a higher total market value.

6.4 4.4 Clubs

[118]:
```
df.Club.unique()
```

[118]:
```
array(['Paris Saint-Germain', 'Borussia Dortmund', 'Tottenham Hotspur',
       'Manchester City', 'Liverpool FC', 'Chelsea FC',
       'Manchester United', 'FC Barcelona', 'Bayern Munich',
       'Inter Milan', 'Atlético de Madrid', 'West Ham United',
       'Real Sociedad', 'Juventus FC', 'SS Lazio', 'Real Madrid',
       'Bayer 04 Leverkusen', 'Arsenal FC', 'Sevilla FC', 'SSC Napoli',
       'Leicester City', 'Everton FC', 'AC Milan', 'Villarreal CF',
       'ACF Fiorentina', 'RB Leipzig', 'AS Roma', 'Crystal Palace',
       'Wolverhampton Wanderers', 'Valencia CF', 'Leeds United',
       'LOSC Lille', 'Aston Villa', 'Olympique Lyon', 'OGC Nice',
       'AS Monaco', 'Atalanta BC', 'US Sassuolo', 'Torino FC',
       'Ajax Amsterdam', 'Brentford FC', 'Southampton FC',
       'Newcastle United', 'VfL Wolfsburg', 'FC Porto',
       'Olympique Marseille', 'Eintracht Frankfurt',
       'Borussia Mönchengladbach', 'Watford FC', 'Stade Rennais FC',
       'Club Brugge KV', 'Sporting CP', 'Brighton & Hove Albion',
       'Dynamo Kyiv', 'Athletic Bilbao', 'Real Betis Balompié',
       'Zenit St. Petersburg', 'Burnley FC', 'SL Benfica',
       'TSG 1899 Hoffenheim', 'Norwich City', 'PSV Eindhoven',
       'VfB Stuttgart', 'KRC Genk', 'Club Atlético Vélez Sarsfield',
       'Club Atlético River Plate', 'FC Metz', 'UC Sampdoria',
       'Red Bull Salzburg', 'Bologna FC 1909', 'Shakhtar Donetsk',
       'Cagliari Calcio', 'Getafe CF', 'Al-Rayyan SC', 'Rubin Kazan',
       'Feyenoord Rotterdam', 'RCD Espanyol Barcelona', 'UD Almería',
       'Sheffield United', 'Celta de Vigo'], dtype=object)
```

[119]:
```
df.Club.nunique()
```

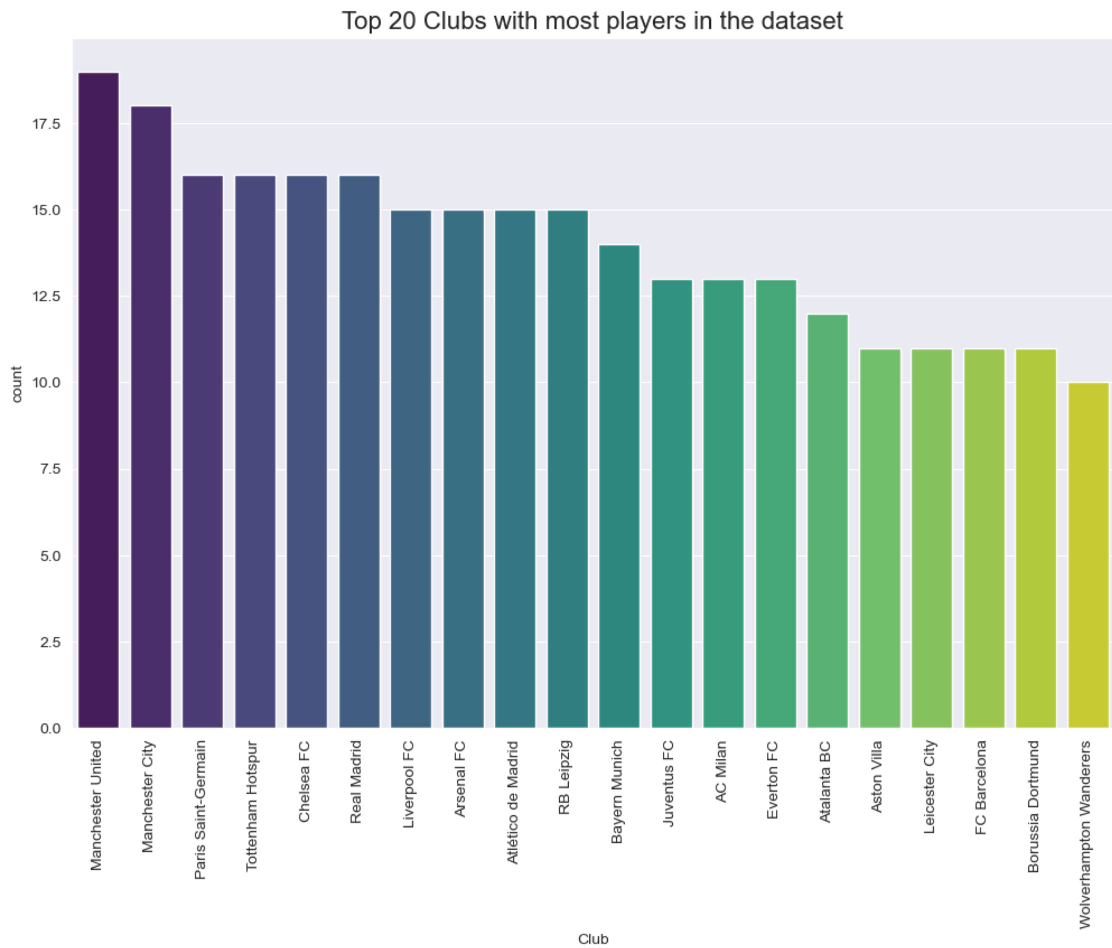
[119]:
```
80
```

[120]:
```
top_20_clubs = df.Club.value_counts().head(20)
top_20_clubs
```

[120]:
```
Manchester United          19
Manchester City            18
Paris Saint-Germain        16
Tottenham Hotspur          16
Chelsea FC                 16
Real Madrid                16
Liverpool FC               15
Arsenal FC                 15
```

```
Atlético de Madrid         15
RB Leipzig                 15
Bayern Munich              14
Juventus FC                13
AC Milan                   13
Everton FC                 13
Atalanta BC                12
Aston Villa                11
Leicester City             11
FC Barcelona               11
Borussia Dortmund          11
Wolverhampton Wanderers    10
Name: Club, dtype: int64
```

[121]:
```python
# visualizing

plt.figure(figsize=(12, 8))

plt.title("Top 20 Clubs with most players in the dataset", fontsize=16)

sns.countplot(data=df, x='Club', order=top_20_clubs.index, palette='viridis')

plt.xticks(rotation=90)

plt.show()
```



[122]:
```python
df.groupby('Club').Market_value.sum().sort_values(ascending = False).
↪nlargest(20)
```

[122]:
```
Club
Manchester City        937
Paris Saint-Germain    772
Manchester United      754
Chelsea FC             704
Bayern Munich          683
Liverpool FC           677
Atlético de Madrid     615
Real Madrid            590
Tottenham Hotspur      530
FC Barcelona           448
Juventus FC            447
Borussia Dortmund      422
Arsenal FC             403
```

```
RB Leipzig              375
Inter Milan             373
AC Milan                354
SSC Napoli              351
Leicester City          349
Everton FC              316
Aston Villa             284
Name: Market_value, dtype: int32
```

[123]:
```
top_20_clubs_value = df.groupby('Club').Market_value.mean().
 ↪sort_values(ascending = False).nlargest(20)
top_20_clubs_value
```

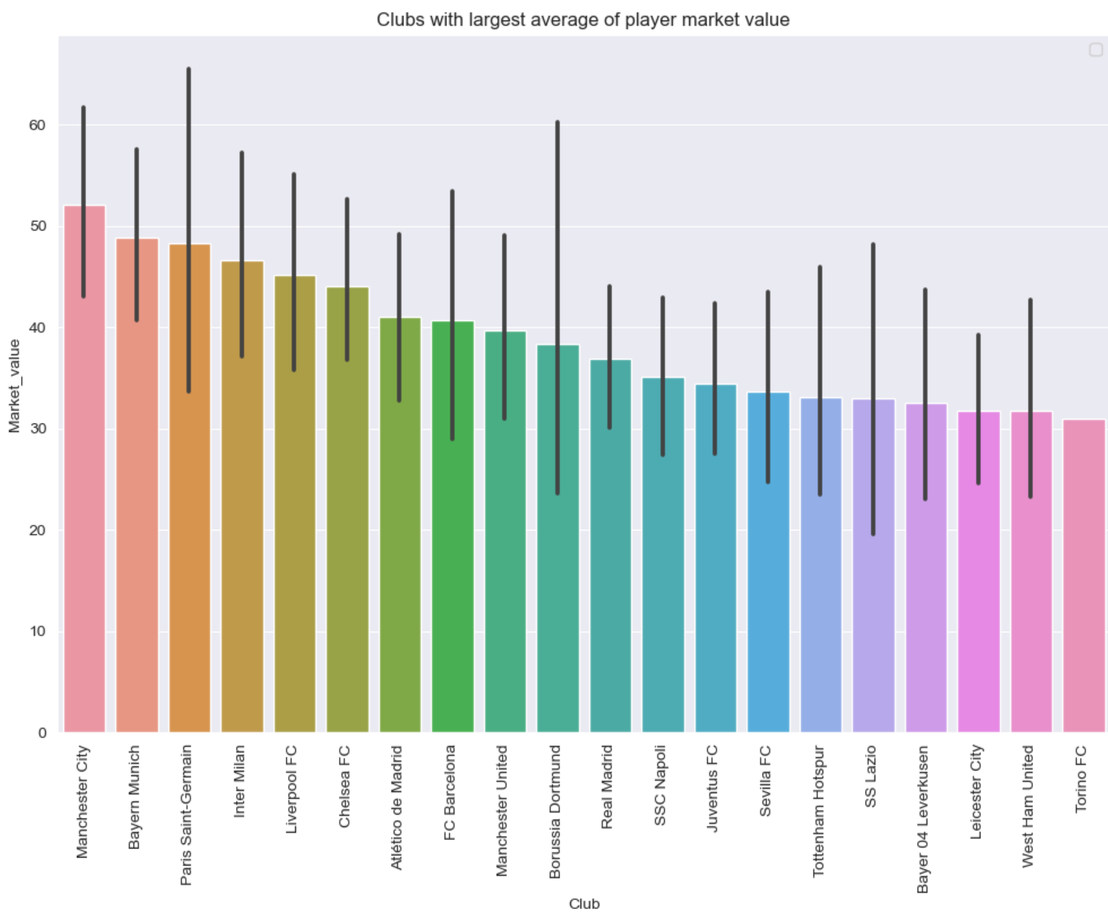
[123]:
```
Club
Manchester City         52.055556
Bayern Munich           48.785714
Paris Saint-Germain     48.250000
Inter Milan             46.625000
Liverpool FC            45.133333
Chelsea FC              44.000000
Atlético de Madrid      41.000000
FC Barcelona            40.727273
Manchester United       39.684211
Borussia Dortmund       38.363636
Real Madrid             36.875000
SSC Napoli              35.100000
Juventus FC             34.384615
Sevilla FC              33.666667
Tottenham Hotspur       33.125000
SS Lazio                33.000000
Bayer 04 Leverkusen     32.500000
Leicester City          31.727273
West Ham United         31.714286
Torino FC               31.000000
Name: Market_value, dtype: float64
```

[124]:
```
# visualizing

plt.figure(figsize=(12,8))

plt.title("Clubs with largest average of player market value")

sns.barplot(data=df, x='Club', y='Market_value', order=top_20_clubs_value.index)

sns.set_style("dark")

plt.xticks(rotation=90)
```

```
plt.legend()
```

```
No artists with labels found to put in legend.  Note that artists whose label
start with an underscore are ignored when legend() is called with no argument.
```

[124]: <matplotlib.legend.Legend at 0x2b068836230>



6.4.1 Insights

1.As we have seen, there are multiple clubs in the dataset. The ones with the highest number of players are as follows: Manchester United (19 players), Manchester City (18 players), Paris Saint-Germain (16 players), Tottenham Hotspur (16 players), Chelsea FC (16 players), and Real Madrid (16 players).

2.Manchester City leads in the total market value sum, followed by Paris Saint-Germain, Manchester United, Chelsea FC, and Bayern Munich.

3.When we observe the average market value per club, we notice that Manchester City, Bayern Munich, Paris Saint-Germain, Inter Milan, and Liverpool have the highest values, indicating how valuable the players of these teams are.

6.5 4.5 Players

[125]:
```
#Number of players
df.Name.nunique()
```

[125]:
```
493
```

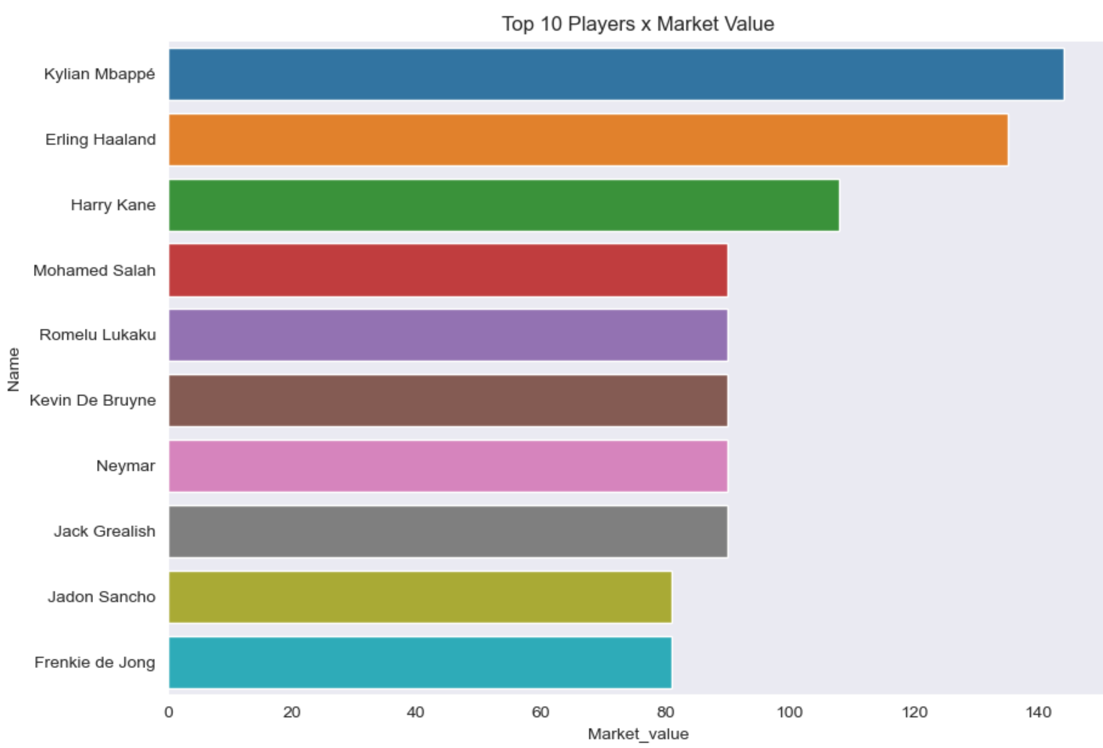
[126]:
```
#Top 10 players based on Market Value
top_10 = df.sort_values(ascending = False, by='Market_value').head(10)
top_10
```

[126]:
```
                Name            Position  Age  Market_value      Country  \
0      Kylian Mbappé      Centre-Forward   22           144       France
1     Erling Haaland      Centre-Forward   21           135       Norway
2         Harry Kane      Centre-Forward   28           108      England
4      Mohamed Salah        Right Winger   29            90        Egypt
5      Romelu Lukaku      Centre-Forward   28            90      Belgium
6    Kevin De Bruyne  Attacking Midfield   30            90      Belgium
7             Neymar         Left Winger   29            90       Brazil
3       Jack Grealish        Left Winger   26            90      England
8       Jadon Sancho         Left Winger   21            81      England
9   Frenkie de Jong     Central Midfield   24            81  Netherlands

                  Club  Matches  Goals  Own_goals  Assists  Yellow_cards  \
0  Paris Saint-Germain       16      7          0       11             3
1    Borussia Dortmund       10     13          0        4             1
2    Tottenham Hotspur       16      7          0        2             2
4         Liverpool FC       15     15          0        6             1
5           Chelsea FC       11      4          0        1             0
6      Manchester City       14      3          0        1             1
7  Paris Saint-Germain       11      3          0        3             3
3      Manchester City       15      2          0        3             1
8    Manchester United       13      0          0        0             0
9         FC Barcelona       13      0          0        2             2

   Second_yellow_cards  Red_cards  Substitute_in  Substitute_out New_Position  \
0                    0          0              0               8     Attacker
1                    0          0              0               1     Attacker
2                    0          0              2               2     Attacker
4                    0          0              0               3     Attacker
5                    0          0              1               2     Attacker
6                    0          0              4               6   Midfielder
7                    0          0              0               3     Attacker
3                    0          0              2               8     Attacker
8                    0          0              7               5     Attacker
```

```
9                       1           0               0                 2   Midfielder

   Goals.per.game  Assists.per.game
0            0.44              0.69
1            1.30              0.40
2            0.44              0.12
4            1.00              0.40
5            0.36              0.09
6            0.21              0.07
7            0.27              0.27
3            0.13              0.20
8            0.00              0.00
9            0.00              0.15
```

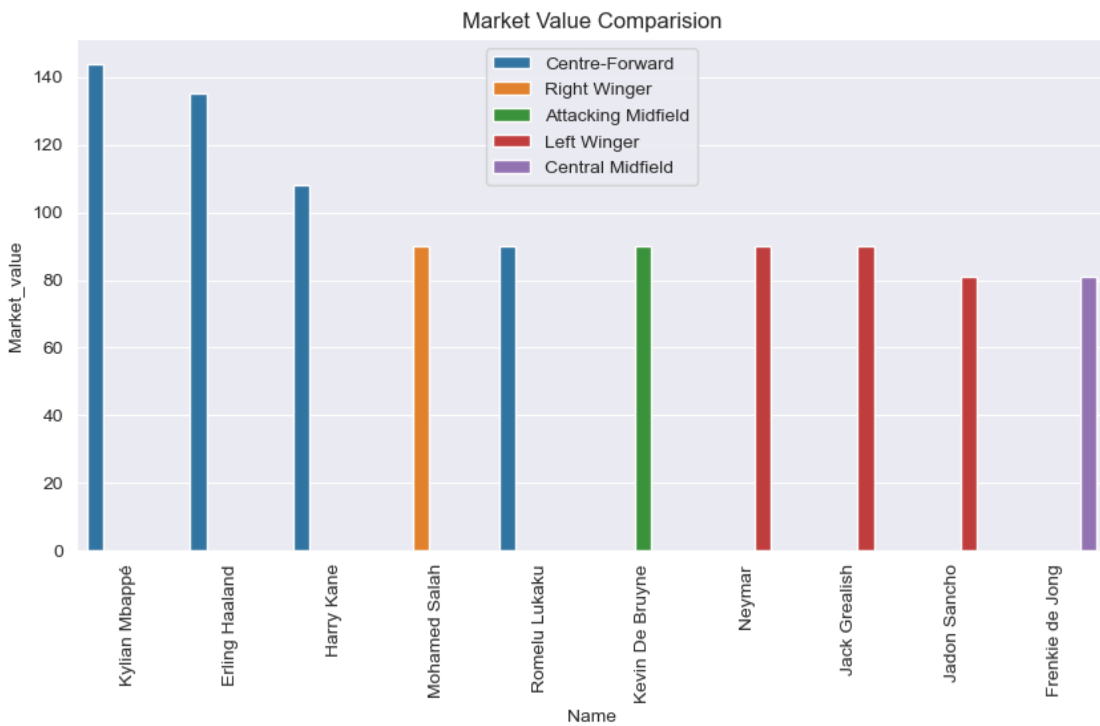
[127]:

```
#visualizing

plt.figure(figsize=(10,7))

plt.title("Top 10 Players x Market Value")

sns.barplot(data=top_10, y="Name", x="Market_value")

sns.set_style("darkgrid")
```



[128]:
```python
# visualizing

plt.figure(figsize=(10,5))

plt.title("Market Value Comparision")

sns.barplot(x='Name', y='Market_value', hue='Position', data=top_10)

plt.xticks(rotation=90)

plt.legend()
```

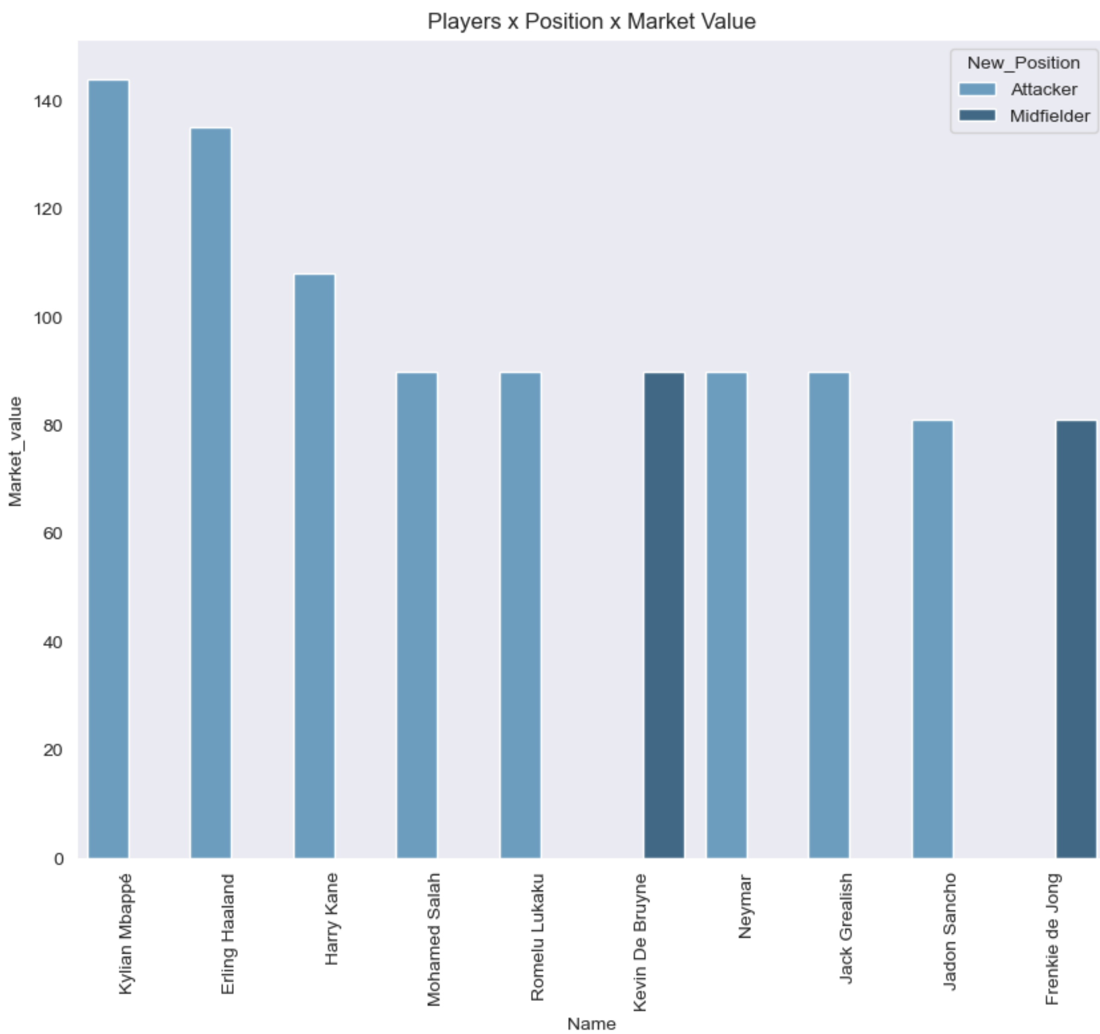
[128]: <matplotlib.legend.Legend at 0x2b06ff6a890>



[134]:
```python
plt.figure(figsize=(10,8))

plt.title("Players x Position x Market Value")

plt.xticks(rotation=90)

sns.set_style("dark")
```

```
sns.barplot(x='Name', y='Market_value', hue='New_Position', data=top_10,␣
↪palette='Blues_d')
```

[134]: <Axes: title={'center': 'Players x Position x Market Value'}, xlabel='Name', ylabel='Market_value'>



[131]:
```
df.groupby(['Name', 'Club']).Market_value.sum().sort_values(ascending = False).
↪nlargest(10)
```

[131]:
```
Name             Club
Kylian Mbappé    Paris Saint-Germain    144
Erling Haaland   Borussia Dortmund      135
Harry Kane       Tottenham Hotspur      108
Mohamed Salah    Liverpool FC            90
Kevin De Bruyne  Manchester City         90
```

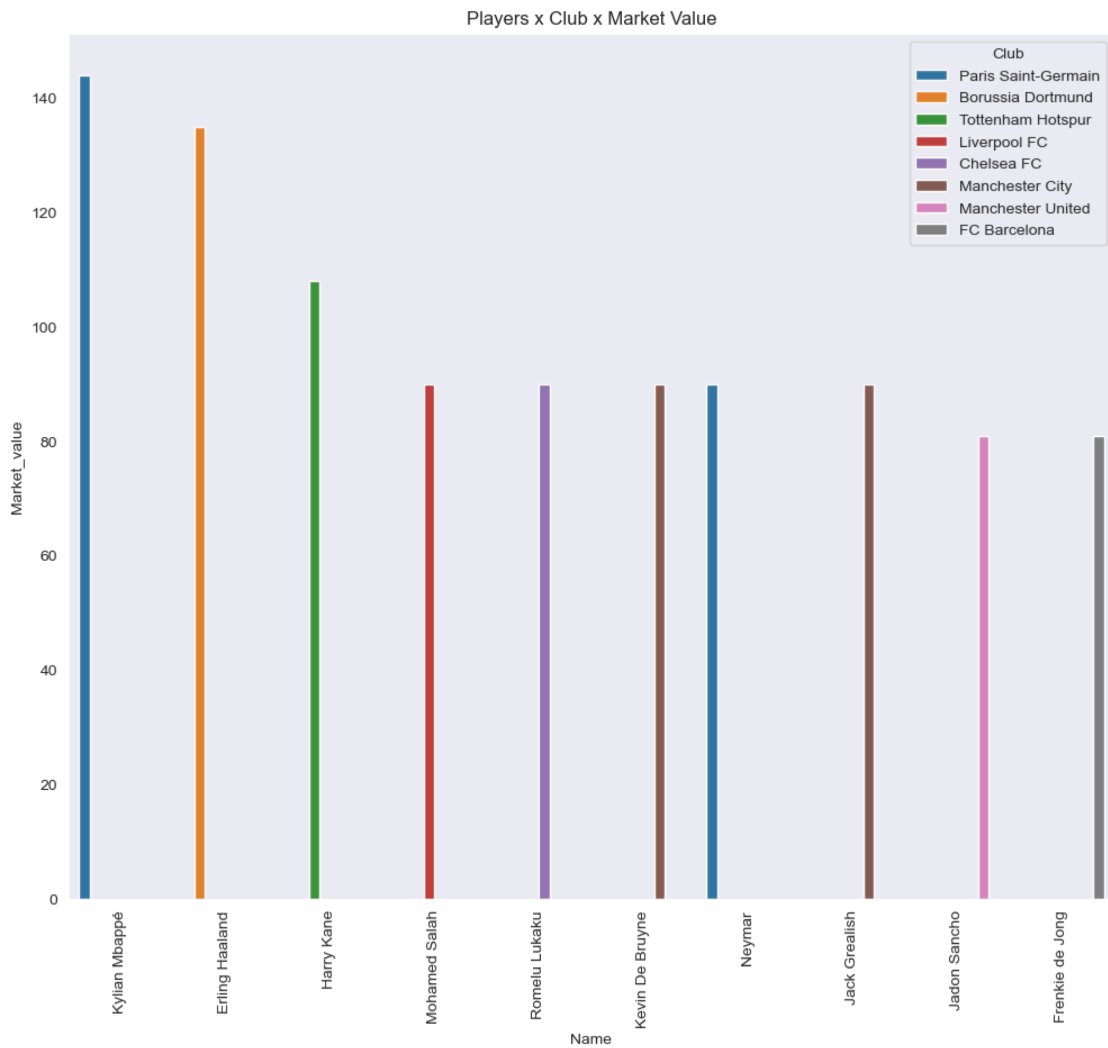
```
Jack Grealish    Manchester City        90
Neymar           Paris Saint-Germain    90
Romelu Lukaku    Chelsea FC             90
Raheem Sterling  Manchester City        81
Joshua Kimmich   Bayern Munich          81
Name: Market_value, dtype: int32
```

[132]:

```
# visualizing

plt.figure(figsize=(12,10))

plt.title("Players x Club x Market Value")

plt.xticks(rotation=90)

sns.set_style("dark")

sns.barplot(x='Name', y='Market_value', hue='Club', data=top_10)
```

[132]:

```
<Axes: title={'center': 'Players x Club x Market Value'}, xlabel='Name',
ylabel='Market_value'>
```



6.5.1 Insights

1.I analyzed the top 10 players with the highest market value in the dataset. As we can see, Kylian Mbappé, Erling Haaland, Harry Kane, Mohamed Salah, and Lukaku are leading the list.

2.Among these 10 players, 8 are forwards, while 2 are midfielders. Additionally, we have 4 Centre-Forwards, 3 Left-Wingers, 1 Right-Winger, 1 Attacking Midfield, and 1 Center-Midfielder.

3.Analyzing by club, we have two players from Manchester City and two from Paris Saint-Germain among the top 10 most valued players. The other clubs have one player each. The clubs are as follows: Borussia Dortmund, Tottenham Hotspur, Liverpool FC, Chelsea FC, Manchester United, and FC Barcelona.

7 5. comparing the two most valuable players based on various parameters.

The two most valuable players are Kylian Mbappé and Erling Haaland

[135]:
```
df.groupby(['Name', 'Club']).Market_value.sum().sort_values(ascending = False).
 ↪nlargest(2)
```

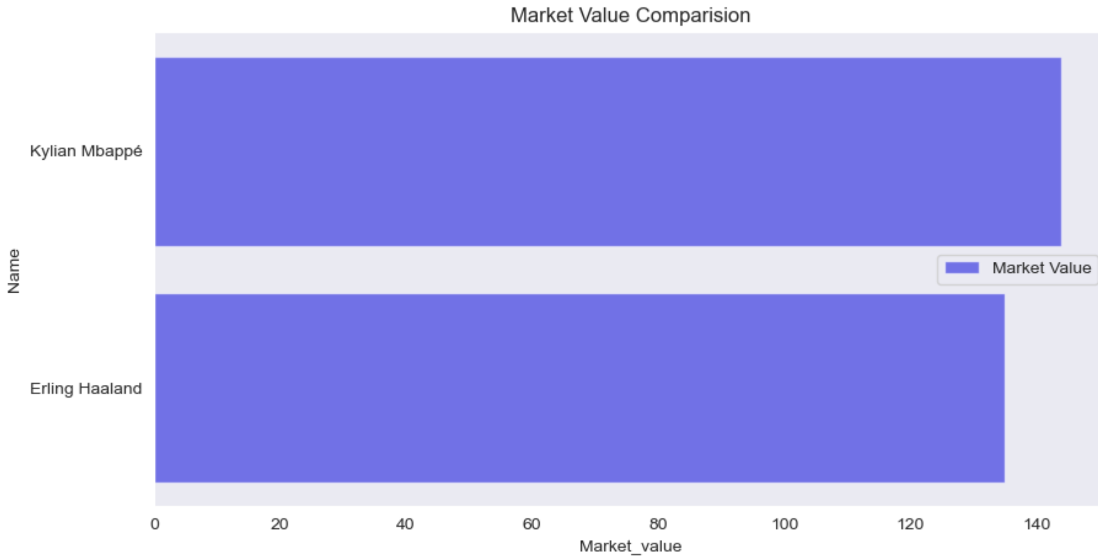
[135]:
```
Name             Club
Kylian Mbappé    Paris Saint-Germain    144
Erling Haaland   Borussia Dortmund      135
Name: Market_value, dtype: int32
```

[137]:
```
top_2 = df[df['Name'].isin(['Erling Haaland', 'Kylian Mbappé'])]
top_2
```

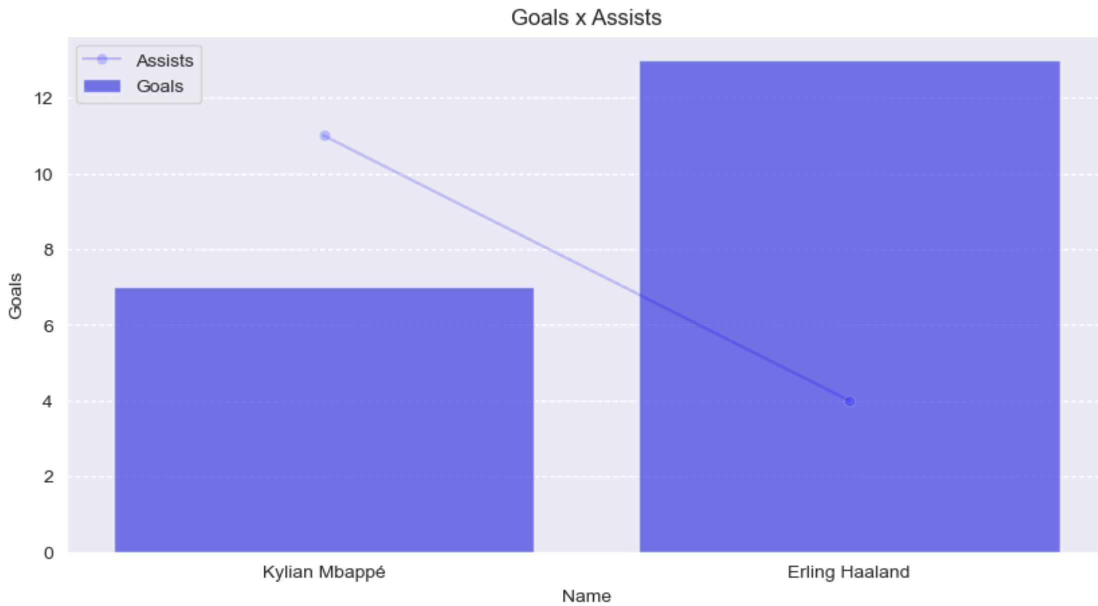
[137]:
```
             Name        Position  Age  Market_value Country  \
0   Kylian Mbappé  Centre-Forward   22           144  France
1  Erling Haaland  Centre-Forward   21           135  Norway

                  Club  Matches  Goals  Own_goals  Assists  Yellow_cards  \
0  Paris Saint-Germain       16      7          0       11             3
1    Borussia Dortmund       10     13          0        4             1

   Second_yellow_cards  Red_cards  Substitute_in  Substitute_out New_Position  \
0                    0          0              0               8     Attacker
1                    0          0              0               1     Attacker

   Goals.per.game  Assists.per.game
0            0.44              0.69
1            1.30              0.40
```

[138]:
```
# visualizing the Market value of two players

plt.figure(figsize=(10,5))

plt.title("Market Value Comparision")

sns.set_style("darkgrid")

sns.barplot(y='Name', x='Market_value', data=top_2, color='blue', alpha=0.6,␣
 ↪label='Market Value')

plt.legend()
```

[138]:
```
<matplotlib.legend.Legend at 0x2b07555f400>
```

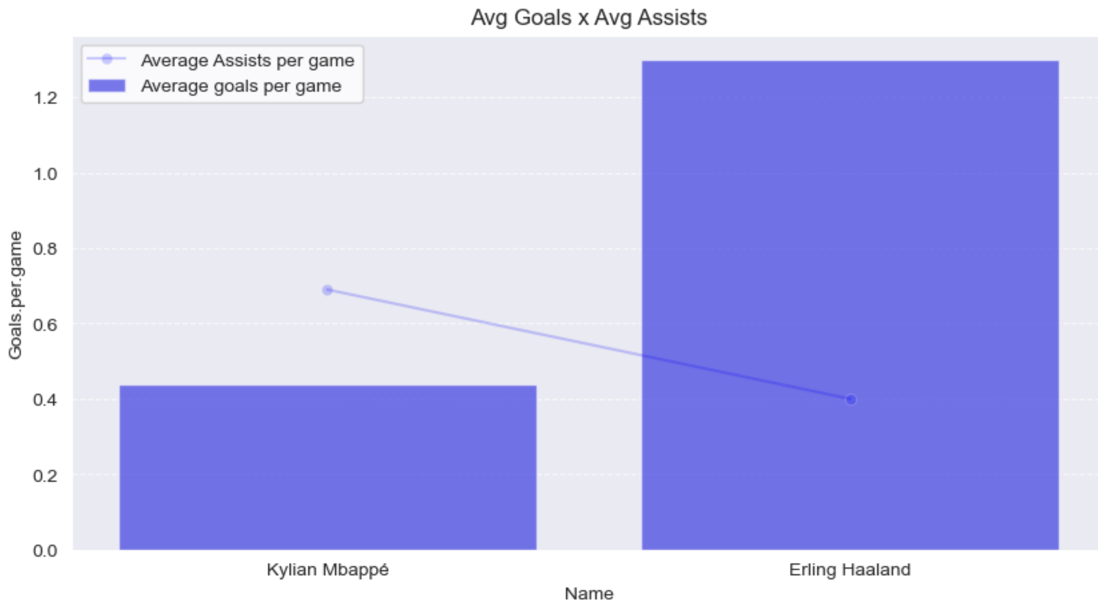


[139]:
```
# visualizing the goals and assists of two players

plt.figure(figsize=(10,5))

plt.title("Goals x Assists")

sns.barplot(x='Name', y='Goals', data=top_2, color='blue', alpha=0.6,␣
↪label='Goals')

sns.lineplot(data=top_2['Assists'], color='blue', alpha=0.2, marker='o',␣
↪label='Assists')

plt.grid(True, axis='y', linestyle='--', alpha=1.0)

sns.set_style("dark")

plt.legend()

plt.show()
```



[140]:

```
# visualizing the Goals per game and assists per game of two players

plt.figure(figsize=(10,5))

plt.title("Avg Goals x Avg Assists")

sns.barplot(x='Name', y='Goals.per.game', data=top_2, color='blue', alpha=0.6,␣
 ↪label='Average goals per game')

sns.lineplot(data=top_2['Assists.per.game'], color='blue', alpha=0.2,␣
 ↪marker='o', label='Average Assists per game')

plt.grid(True, axis='y', linestyle='--', alpha=0.7)

sns.set_style("white")

plt.legend()

plt.show()
```



7.0.1 Insights

1.We noticed that Kylian Mbappé is ahead of Erling Haaland in terms of market value.

2.When comparing goals and assists, we observed two opposite dynamics: Mbappé had more assists, while Haaland scored more goals. This same trend repeated for the average goals per match and average assists per match.

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```