In [1]:
```python
# import the libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
Matplotlib is building the font cache; this may take a moment.
```

In [4]:
```python
#loading Dataset

epl_df = pd.read_csv('E:\\Data analysis training\\epl analysis\\EPL_20_21.csv')
epl_df.head()
```

Out[4]:

	Name	Club	Nationality	Position	Age	Matches	Starts	Mins	Goals	Assists	Passes_Attempted	Perc_Passes_Completed	Penalty_Goals	Penalty_Attempted	xG	xA	Yellow_Cards	Red_Cards
0	Mason Mount	Chelsea	ENG	MF,FW	21	36	32	2890	6	5	1881	82.3	1	1	0.21	0.24	2	0
1	Edouard Mendy	Chelsea	SEN	GK	28	31	31	2745	0	0	1007	84.6	0	0	0.00	0.00	2	0
2	Timo Werner	Chelsea	GER	FW	24	35	29	2602	6	8	826	77.2	0	0	0.41	0.21	2	0
3	Ben Chilwell	Chelsea	ENG	DF	23	27	27	2286	3	5	1806	78.6	0	0	0.10	0.11	3	0
4	Reece James	Chelsea	ENG	DF	20	32	25	2373	1	2	1987	85.0	0	0	0.06	0.12	3	0

In [5]:
```python
# columns and their names
epl_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 532 entries, 0 to 531
Data columns (total 18 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Name                   532 non-null    object
 1   Club                   532 non-null    object
 2   Nationality            532 non-null    object
 3   Position               532 non-null    object
 4   Age                    532 non-null    int64
 5   Matches                532 non-null    int64
 6   Starts                 532 non-null    int64
 7   Mins                   532 non-null    int64
 8   Goals                  532 non-null    int64
 9   Assists                532 non-null    int64
 10  Passes_Attempted       532 non-null    int64
 11  Perc_Passes_Completed  532 non-null    float64
 12  Penalty_Goals          532 non-null    int64
 13  Penalty_Attempted      532 non-null    int64
 14  xG                     532 non-null    float64
 15  xA                     532 non-null    float64
 16  Yellow_Cards           532 non-null    int64
 17  Red_Cards              532 non-null    int64
dtypes: float64(3), int64(11), object(4)
memory usage: 74.9+ KB
```

In [6]:
```python
# sumarry statistics about dataset, works only on numeric columns
epl_df.describe()
```

Out[6]:

	Age	Matches	Starts	Mins	Goals	Assists	Passes_Attempted	Perc_Passes_Completed	Penalty_Goals	Penalty_Attempted	xG	xA	Yellow_Cards	Red_Cards
count	532.000000	532.000000	532.000000	532.000000	532.000000	532.000000	532.000000	532.000000	532.000000	532.000000	532.000000	532.000000	532.000000	532.000000
mean	25.500000	19.535714	15.714286	1411.443609	1.853383	1.287594	717.750000	77.823872	0.191729	0.234962	0.113289	0.072650	2.114662	0.090226
std	4.319404	11.840459	11.921161	1043.171856	3.338009	2.095191	631.372522	13.011631	0.850881	0.975818	0.148174	0.090072	2.269094	0.293268
min	16.000000	1.000000	0.000000	1.000000	0.000000	0.000000	0.000000	-1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	22.000000	9.000000	4.000000	426.000000	0.000000	0.000000	171.500000	73.500000	0.000000	0.000000	0.010000	0.000000	0.000000	0.000000
50%	26.000000	21.000000	15.000000	1345.000000	1.000000	0.000000	573.500000	79.200000	0.000000	0.000000	0.060000	0.050000	2.000000	0.000000
75%	29.000000	30.000000	27.000000	2303.500000	2.000000	2.000000	1129.500000	84.625000	0.000000	0.000000	0.150000	0.110000	3.000000	0.000000
max	38.000000	38.000000	38.000000	3420.000000	23.000000	14.000000	3214.000000	100.000000	9.000000	10.000000	1.160000	0.900000	12.000000	2.000000

In [7]:
```python
# finding total number of null values
epl_df.isna().sum()
```

Out[7]:
```
Name                     0
Club                     0
Nationality              0
Position                 0
Age                      0
Matches                  0
Starts                   0
Mins                     0
Goals                    0
Assists                  0
Passes_Attempted         0
Perc_Passes_Completed    0
Penalty_Goals            0
Penalty_Attempted        0
xG                       0
xA                       0
Yellow_Cards             0
Red_Cards                0
dtype: int64
```

In [11]:
```python
# Create 2 new columns

epl_df['MinsperMatch'] = epl_df['Mins'] / epl_df['Matches'].astype(int)
epl_df['GoalsPerMatch'] = epl_df['Goals']    / epl_df['Matches'].astype(int)
epl_df.head()
```

Out[11]:

	Name	Club	Nationality	Position	Age	Matches	Starts	Mins	Goals	Assists	Passes_Attempted	Perc_Passes_Completed	Penalty_Goals	Penalty_Attempted	xG	xA	Yellow_Cards	Red_Cards	MinsperMatch	Goals
0	Mason Mount	Chelsea	ENG	MF,FW	21	36	32	2890	6	5	1881	82.3	1	1	0.21	0.24	2	0	80.277778	
1	Edouard Mendy	Chelsea	SEN	GK	28	31	31	2745	0	0	1007	84.6	0	0	0.00	0.00	2	0	88.548387	
2	Timo Werner	Chelsea	GER	FW	24	35	29	2602	6	8	826	77.2	0	0	0.41	0.21	2	0	74.342857	
3	Ben Chilwell	Chelsea	ENG	DF	23	27	27	2286	3	5	1806	78.6	0	0	0.10	0.11	3	0	84.666667	
4	Reece James	Chelsea	ENG	DF	20	32	25	2373	1	2	1987	85.0	0	0	0.06	0.12	3	0	74.156250	

In [12]:
```python
# total Goals

Total_Goals = epl_df['Goals'].sum()
print(Total_Goals)
```

```
986
```

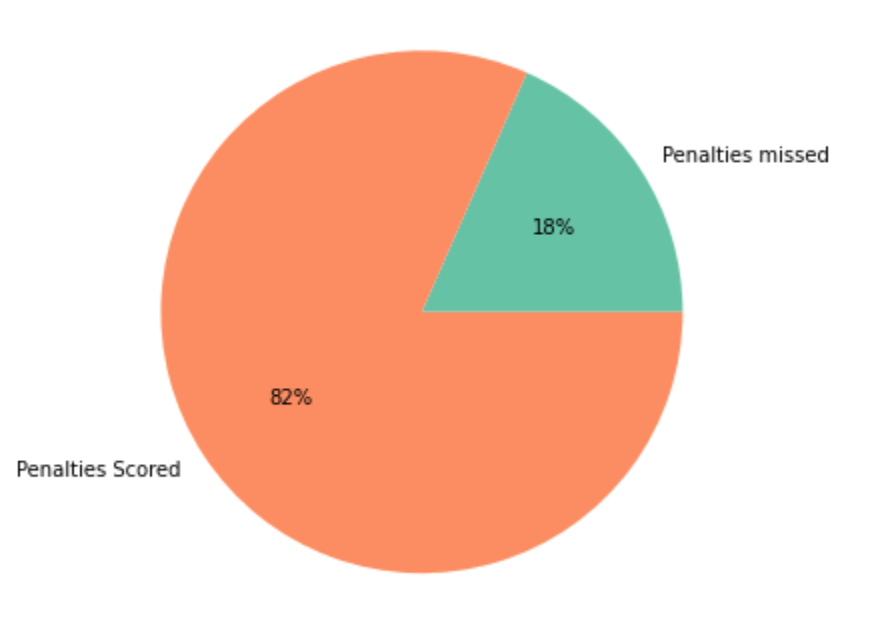
In [13]:
```python
# total penalty goals

Total_Penalty = epl_df['Penalty_Goals'].sum()
print(Total_Penalty)
```

```
102
```

In [15]:
```python
# total penalty attempts

Total_PenaltyAttempts = epl_df['Penalty_Attempted'].sum()
print(Total_PenaltyAttempts)
```

```
125
```

In [21]:
```python
# Pie chart for penalties missed vs scored

plt.figure(figsize=(13,6))
pl_not_scored = epl_df['Penalty_Attempted'].sum() - Total_Penalty
data = [pl_not_scored, Total_Penalty]
labels = ['Penalties missed', 'Penalties Scored']
color = sns.color_palette('Set2')
plt.pie(data,labels = labels, colors =  color, autopct = '%.0f%%')
plt.show()
```



In [22]:
```python
#unique positions

epl_df['Position'].unique()
```

Out[22]:
```
array(['MF,FW', 'GK', 'FW', 'DF', 'MF', 'FW,MF', 'FW,DF', 'DF,MF',
       'MF,DF', 'DF,FW'], dtype=object)
```

In [23]:
```python
# total FW players

epl_df[epl_df['Position'] == 'FW']

#subseting the position
```

Out[23]:

	Name	Club	Nationality	Position	Age	Matches	Starts	Mins	Goals	Assists	Passes_Attempted	Perc_Passes_Completed	Penalty_Goals	Penalty_Attempted	xG	xA	Yellow_Cards	Red_Cards	MinsperMatch
2	Timo Werner	Chelsea	GER	FW	24	35	29	2602	6	8	826	77.2	0	0	0.41	0.21	2	0	74.342857
16	Tammy Abraham	Chelsea	ENG	FW	22	22	12	1040	6	1	218	68.3	0	0	0.56	0.07	0	0	47.272727
19	Olivier Giroud	Chelsea	FRA	FW	33	17	8	748	4	0	217	74.2	0	0	0.58	0.09	1	0	44.000000
23	Ruben Loftus-Cheek	Chelsea	ENG	FW	24	1	1	60	0	0	16	68.8	0	0	0.00	0.00	0	0	60.000000
30	Raheem Sterling	Manchester City	ENG	FW	25	31	28	2536	10	7	1127	85.4	0	1	0.43	0.17	4	0	81.806452
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
516	Oliver Burke	Sheffield United	SCO	FW	23	25	14	1269	1	1	262	70.6	0	0	0.17	0.13	2	0	50.760000
518	Oliver McBurnie	Sheffield United	SCO	FW	24	23	12	1324	1	0	426	62.9	0	0	0.21	0.07	2	0	57.565217
519	Rhian Brewster	Sheffield United	ENG	FW	20	27	12	1128	0	0	225	69.3	0	0	0.14	0.13	1	0	41.777778
523	Billy Sharp	Sheffield United	ENG	FW	34	16	7	735	3	0	123	69.9	2	2	0.33	0.07	1	0	45.937500
526	Daniel Jebbison	Sheffield United	ENG	FW	17	4	3	284	1	0	34	70.6	0	0	0.50	0.01	0	0	71.000000

81 rows × 20 columns

In [24]:
```python
#players from different Nations

np.size((epl_df['Nationality'].unique()))
```

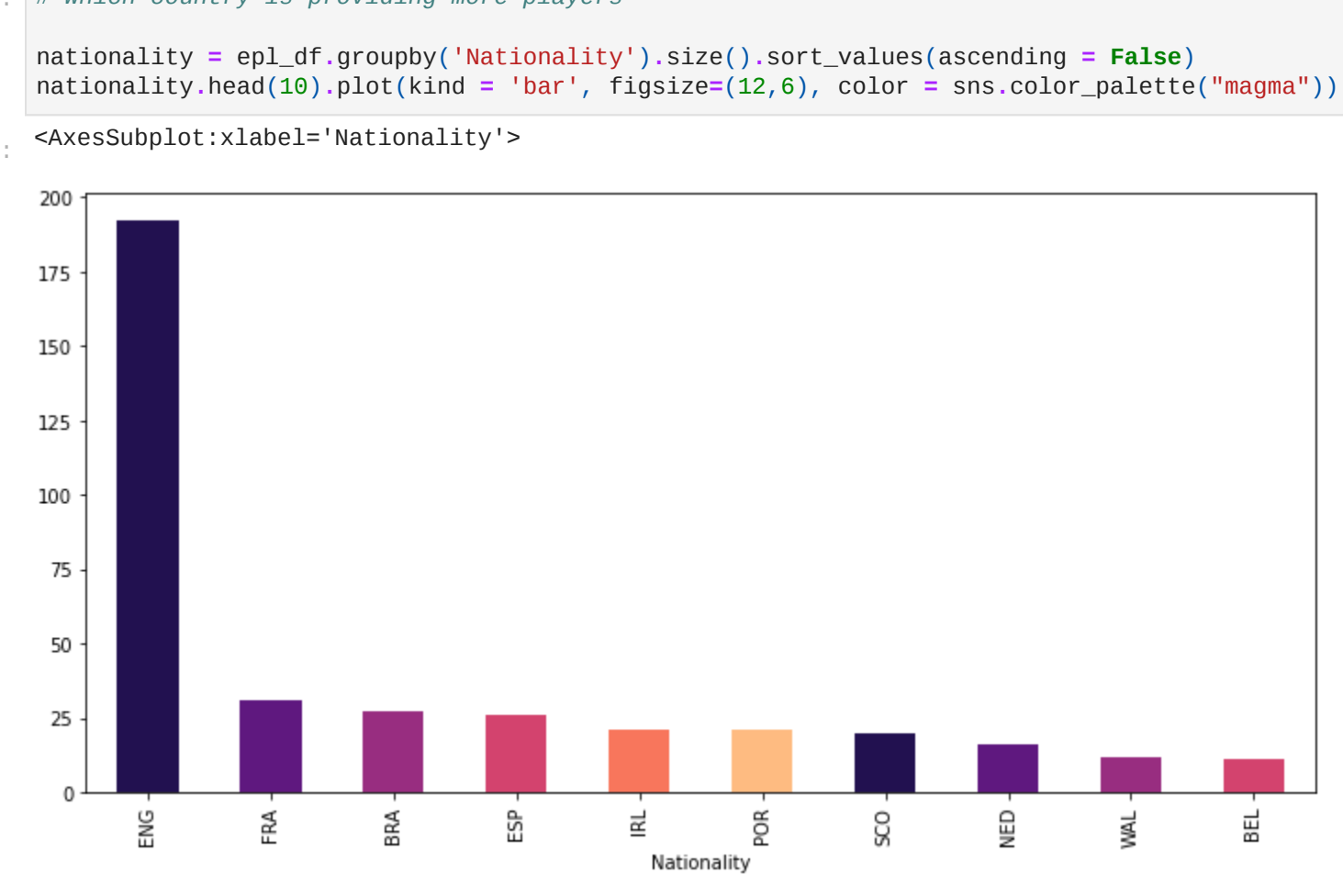
Out[24]:
```
59
```

In [25]:
```python
# what the nations they are representing

epl_df['Nationality'].unique()
```

Out[25]:
```
array(['ENG', 'SEN', 'GER', 'ESP', 'FRA', 'ITA', 'BRA', 'CRO', 'USA',
       'DEN', 'MAR', 'SCO', 'ARG', 'POR', 'BEL', 'ALG', 'UKR', 'NED',
       'SWE', 'URU', 'SRB', 'WAL', 'CIV', 'NGA', 'EGY', 'TUR', 'CMR',
       'GUI', 'SUI', 'JPN', 'IRL', 'GRE', 'NIR', 'GHA', 'AUT', 'JAM',
       'RSA', 'CZE', 'POL', 'PAR', 'COD', 'KOR', 'COL', 'GAB', 'NOR',
       'AUS', 'BIH', 'ISL', 'MKD', 'BFA', 'ZIM', 'SVK', 'MEX', 'CAN',
       'MLI', 'IRN', 'NZL', 'MTN', 'SKN'], dtype=object)
```

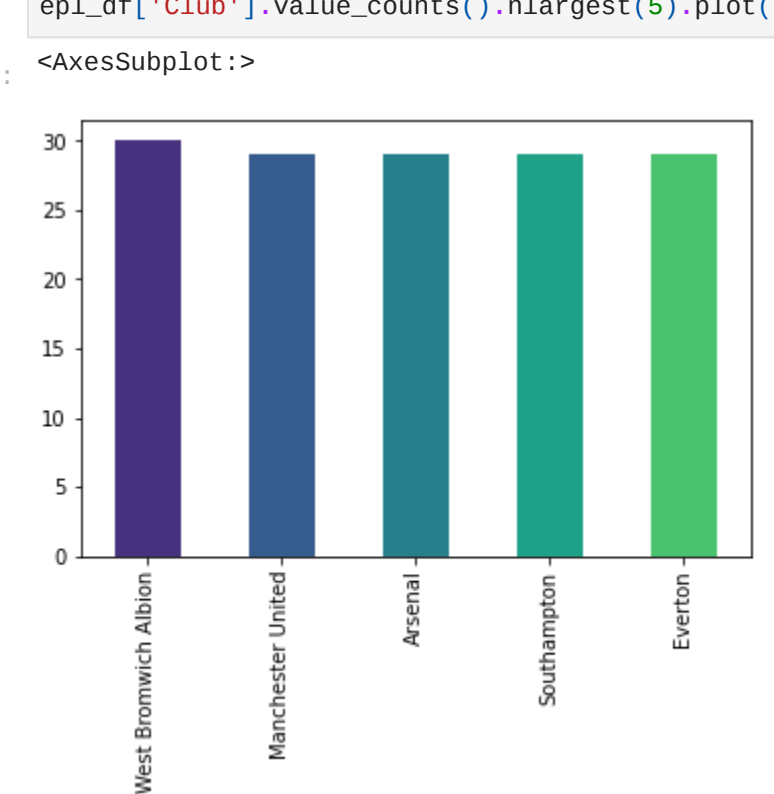
In [27]:
```python
# Which country is providing more players

nationality = epl_df.groupby('Nationality').size().sort_values(ascending = False)
nationality.head(10).plot(kind = 'bar', figsize=(12,6), color = sns.color_palette("magma"))
```

Out[27]:
```
<AxesSubplot:xlabel='Nationality'>
```



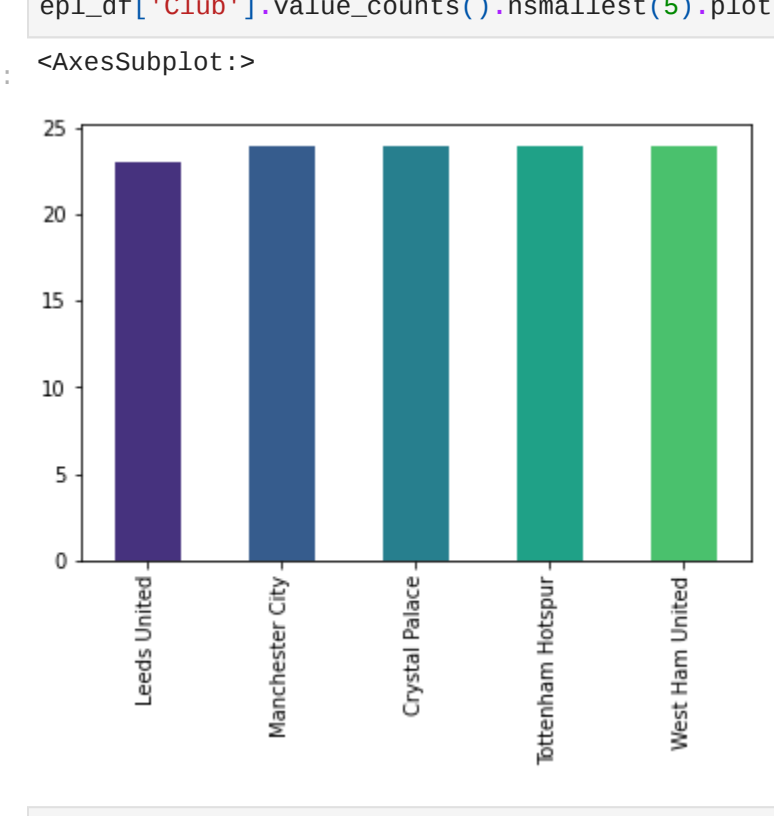
In [30]:
```python
#clubs with maximum players in their squad

epl_df['Club'].value_counts().nlargest(5).plot(kind = 'bar', color = sns.color_palette("viridis"))
```

Out[30]:
```
<AxesSubplot:>
```



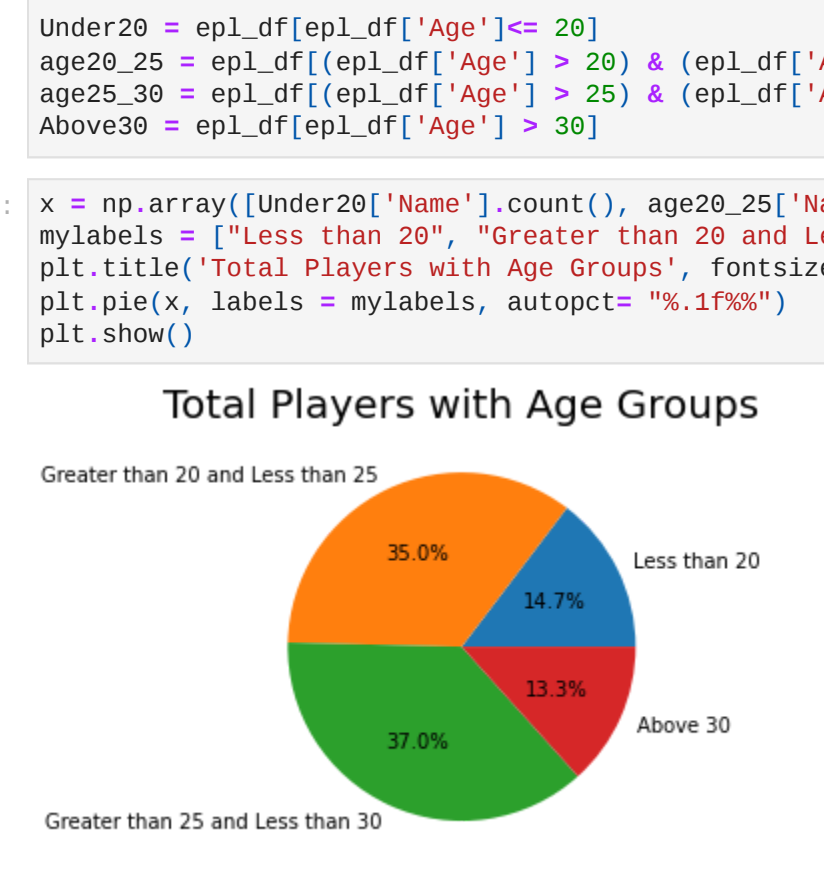
In [31]:
```python
#clubs with least player in the squad

epl_df['Club'].value_counts().nsmallest(5).plot(kind = 'bar', color = sns.color_palette("viridis"))
```

Out[31]:
```
<AxesSubplot:>
```



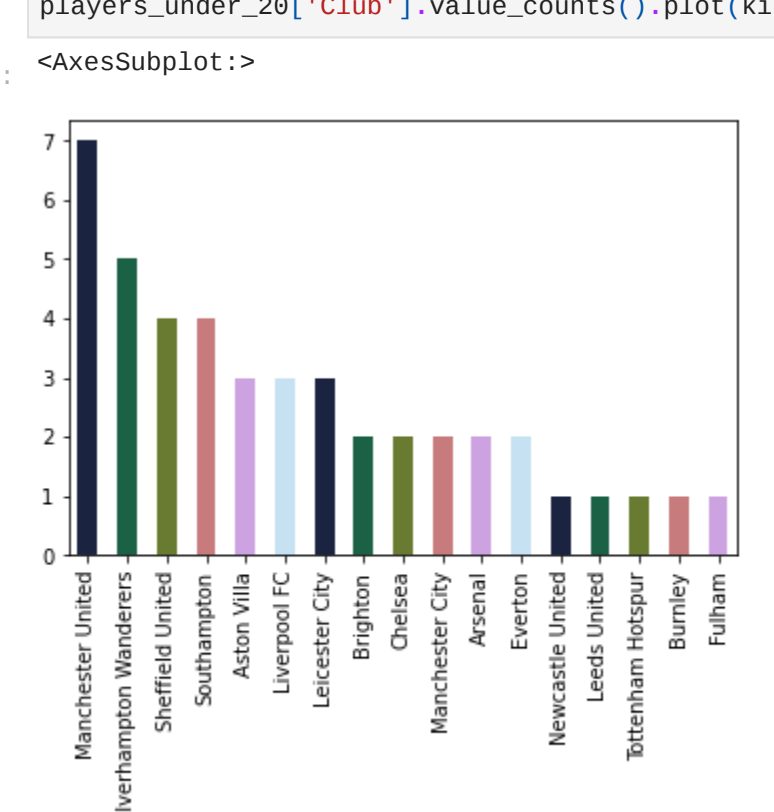
In [39]:
```python
# players based on age group

Under20 = epl_df[epl_df['Age']<= 20]
age20_25 = epl_df[(epl_df['Age'] > 20) & (epl_df['Age'] <= 25)]
age25_30 = epl_df[(epl_df['Age'] > 25) & (epl_df['Age'] <= 30)]
Above30 = epl_df[epl_df['Age'] > 30]
```

In [40]:
```python
x = np.array([Under20['Name'].count(), age20_25['Name'].count(), age25_30['Name'].count(), Above30['Name'].count()])
mylabels = ["Less than 20", "Greater than 20 and Less than 25", "Greater than 25 and Less than 30", "Above 30"]
plt.title('Total Players with Age Groups', fontsize = 20)
plt.pie(x, labels = mylabels, autopct= "%.1f%%")
plt.show()
```



In [44]:
```python
# Toatal under 20 players in each club

players_under_20 = epl_df[epl_df['Age'] < 20]
players_under_20['Club'].value_counts().plot(kind = 'bar', color= sns.color_palette("cubehelix"))
```

Out[44]:
```
<AxesSubplot:>
```



In [46]:
```python
# under 20 players in Manchester united

players_under_20[players_under_20['Club']=='Manchester United']
```

Out[46]:

	Name	Club	Nationality	Position	Age	Matches	Starts	Mins	Goals	Assists	Passes_Attempted	Perc_Passes_Completed	Penalty_Goals	Penalty_Attempted	xG	xA	Yellow_Cards	Red_Cards	MinsperMatch
61	Mason Greenwood	Manchester United	ENG	FW	18	31	21	1822	7	2	732	83.1	0	0	0.37	0.09	2	0	58.774194
72	Brandon Williams	Manchester United	ENG	DF	19	4	2	188	0	0	140	85.7	0	0	0.05	0.01	0	0	47.000000
73	Amad Diallo	Manchester United	CIV	FW	18	3	2	166	0	1	64	84.4	0	0	0.02	0.26	0	0	55.333333
74	Anthony Elanga	Manchester United	SWE	FW	18	2	2	155	1	0	53	81.1	0	0	0.16	0.02	0	0	77.500000
76	Shola Shoretire	Manchester United	ENG	FW	16	2	0	11	0	0	8	75.0	0	0	0.00	0.00	0	0	5.500000
78	Hannibal Mejbri	Manchester United	FRA	MF	17	1	0	9	0	0	3	100.0	0	0	0.00	0.00	0	0	9.000000
79	William Thomas Fish	Manchester United	ENG	DF	17	1	0	1	0	0	1	0.0	0	0	0.00	0.00	0	0	1.000000

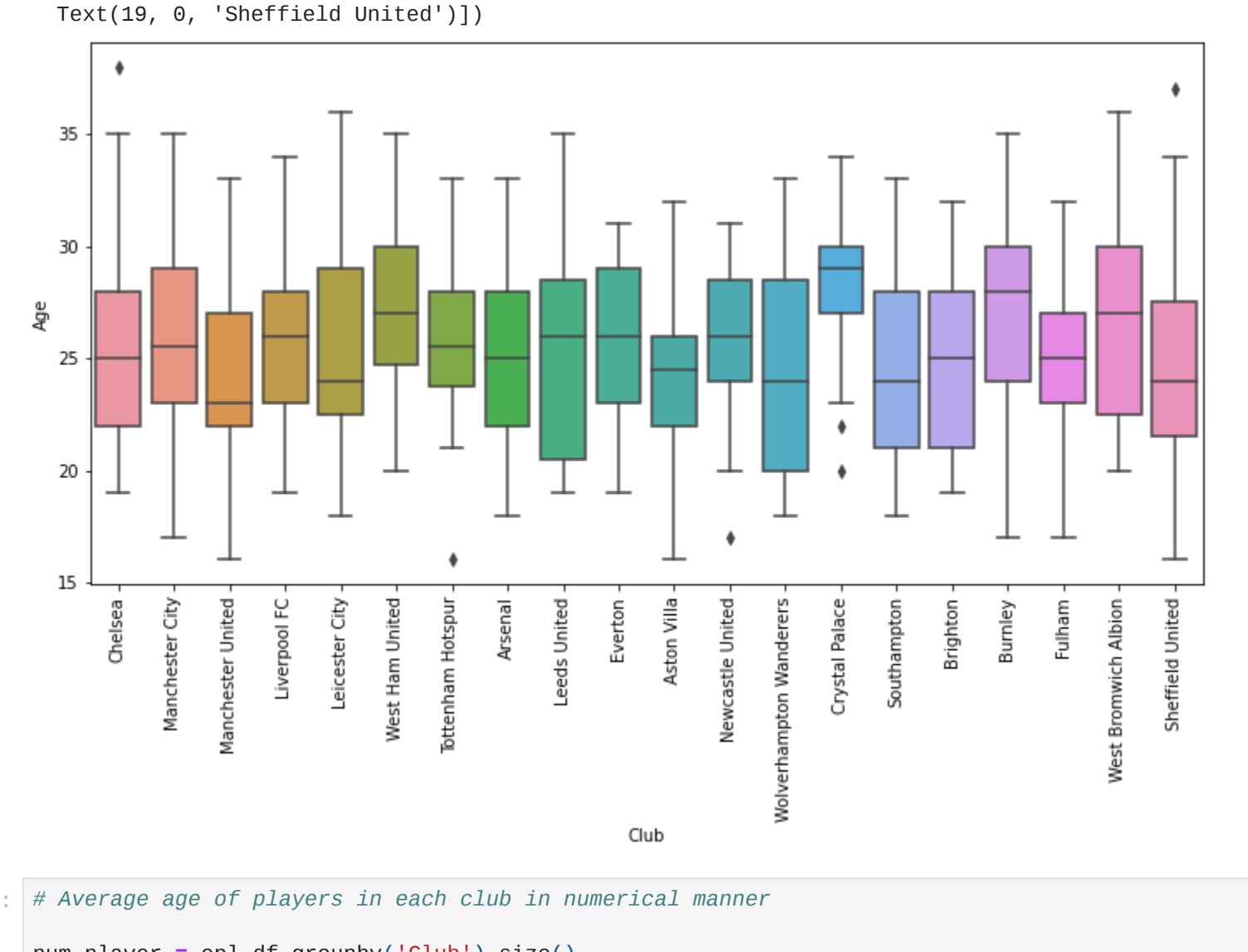
In [48]:
```python
#under 20 players in Liverpool

players_under_20[players_under_20['Club']=='Liverpool FC']
```

Out[48]:

	Name	Club	Nationality	Position	Age	Matches	Starts	Mins	Goals	Assists	Passes_Attempted	Perc_Passes_Completed	Penalty_Goals	Penalty_Attempted	xG	xA	Yellow_Cards	Red_Cards	MinsperMatch	Go
91	Curtis Jones	Liverpool FC	ENG	MF	19	24	13	1179	1	2	976	91.2	0	0	0.11	0.12	2	0	49.125000	
96	Rhys Williams	Liverpool FC	ENG	DF	19	9	7	661	0	0	451	92.0	0	0	0.07	0.05	0	0	73.444444	
102	Neco Williams	Liverpool FC	WAL	DF	19	6	3	249	0	0	213	73.2	0	0	0.03	0.10	1	0	41.500000	

In [50]:
```python
# Average age of players in each club

plt.figure(figsize=(12,6))
sns.boxplot(x = 'Club' , y='Age', data = epl_df)
plt.xticks(rotation = 90)
```

Out[50]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19]),
 [Text(0, 0, 'Chelsea'),
  Text(1, 0, 'Manchester City'),
  Text(2, 0, 'Manchester United'),
  Text(3, 0, 'Liverpool FC'),
  Text(4, 0, 'Leicester City'),
  Text(5, 0, 'West Ham United'),
  Text(6, 0, 'Tottenham Hotspur'),
  Text(7, 0, 'Arsenal'),
  Text(8, 0, 'Leeds United'),
  Text(9, 0, 'Everton'),
  Text(10, 0, 'Aston Villa'),
  Text(11, 0, 'Newcastle United'),
  Text(12, 0, 'Wolverhampton Wanderers'),
  Text(13, 0, 'Crystal Palace'),
  Text(14, 0, 'Southampton'),
  Text(15, 0, 'Brighton'),
  Text(16, 0, 'Burnley'),
  Text(17, 0, 'Fulham'),
  Text(18, 0, 'West Bromwich Albion'),
  Text(19, 0, 'Sheffield United')])
```

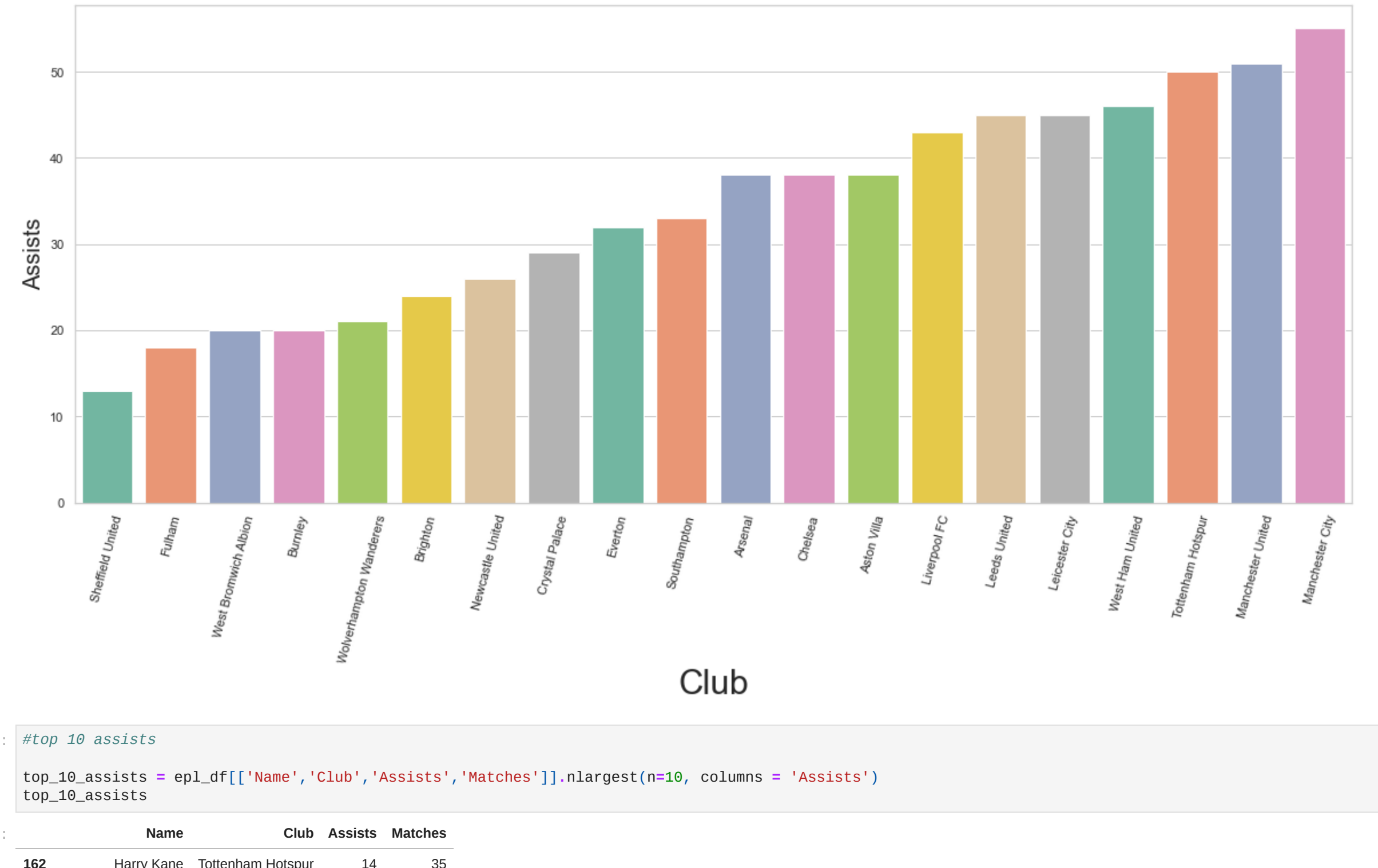


In [52]:
```python
# Average age of players in each club in numerical manner

num_player = epl_df.groupby('Club').size()
data = (epl_df.groupby('Club')['Age'].sum())/num_player
data.sort_values(ascending = False)
```

Out[52]:
```
Club
Crystal Palace             28.333333
West Ham United            27.500000
Burnley                    27.040000
West Bromwich Albion       26.766667
Newcastle United           26.074074
Manchester City            25.708333
Tottenham Hotspur          25.625000
Chelsea                    25.592593
Leicester City             25.592593
Liverpool FC               25.571429
Everton                    25.413793
Leeds United               25.347826
Fulham                     25.035714
Arsenal                    24.965517
Sheffield United           24.814815
Brighton                   24.555556
Wolverhampton Wanderers    24.444444
Aston Villa                24.291667
Southampton                24.137931
Manchester United          23.862069
dtype: float64
```

In [55]:
```python
# total assist from each clubs

Assists_by_clubs = pd.DataFrame(epl_df.groupby('Club', as_index=False)['Assists'].sum())
sns.set_theme(style="whitegrid", color_codes=True)
ax = sns.barplot(x='Club', y='Assists', data = Assists_by_clubs.sort_values(by="Assists"), palette='Set2')
ax.set_xlabel("Club", fontsize=30)
ax.set_ylabel("Assists", fontsize = 20)
plt.xticks(rotation=75)
plt.rcParams['figure.figsize']= (20,8)
plt.title('Plot of Clubs vs Total Assists', fontsize = 20)
```

Out[55]:
```
Text(0.5, 1.0, 'Plot of Clubs vs Total Assists')
```



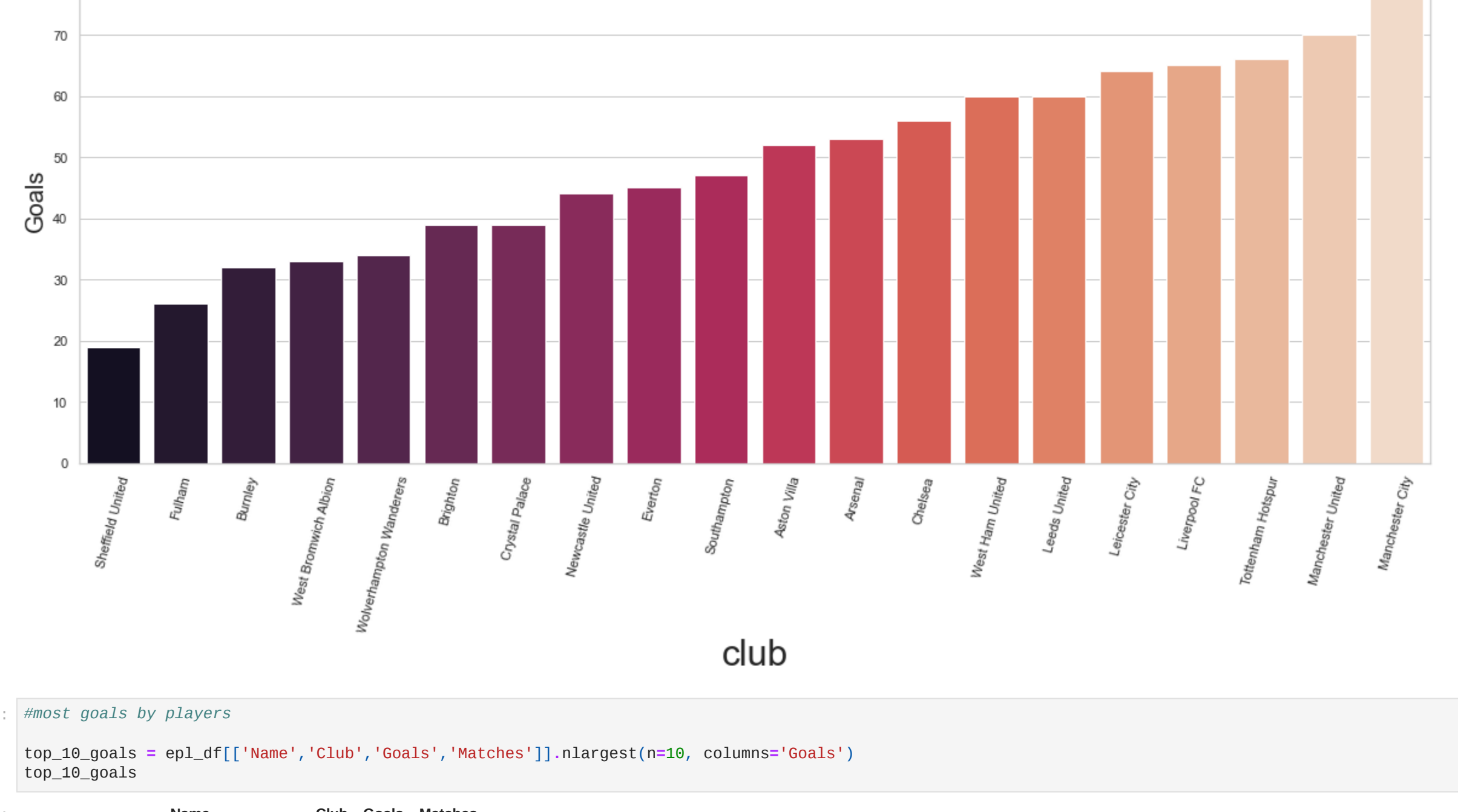
In [65]:
```python
#top 10 assists

top_10_assists = epl_df[['Name','Club','Assists','Matches']].nlargest(n=10, columns = 'Assists')
top_10_assists
```

Out[65]:

	Name	Club	Assists	Matches
162	Harry Kane	Tottenham Hotspur	14	35
34	Kevin De Bruyne	Manchester City	12	25
51	Bruno Fernandes	Manchester United	12	37
161	Son Heung-min	Tottenham Hotspur	10	37
273	Jack Grealish	Aston Villa	10	26
54	Marcus Rashford	Manchester United	9	37
110	Jamie Vardy	Leicester City	9	34
220	Raphael Dias Belloli	Leeds United	9	30
2	Timo Werner	Chelsea	8	35
136	Aaron Cresswell	West Ham United	8	36

In [59]:
```python
#total goals scored by club

Goals_by_clubs = pd.DataFrame(epl_df.groupby('Club', as_index = False )['Goals'].sum() )
sns.set_theme(style = "whitegrid", color_codes = True )
ax = sns.barplot (x= 'Club', y = 'Goals' , data = Goals_by_clubs.sort_values(by= "Goals"), palette = 'rocket')
ax.set_xlabel ("club" ,  fontsize = 30 )
ax.set_ylabel ("Goals" ,  fontsize = 20 )
plt.xticks ( rotation = 75 )
plt.rcParams ["figure.figsize"] = ( 20,8 )
plt.title ('Plot of Clubs vs Total Goals , fontsize = 20 ')
```

Out[59]:
```
Text(0.5, 1.0, 'Plot of Clubs vs Total Goals , fontsize = 20 ')
```



In [66]:
```python
#most goals by players

top_10_goals = epl_df[['Name','Club','Goals','Matches']].nlargest(n=10, columns='Goals')
top_10_goals
```

Out[66]:

	Name	Club	Goals	Matches
162	Harry Kane	Tottenham Hotspur	23	35
81	Mohamed Salah	Liverpool FC	22	37
51	Bruno Fernandes	Manchester United	18	37
161	Son Heung-min	Tottenham Hotspur	17	37
214	Patrick Bamford	Leeds United	17	38
237	Dominic Calvert-Lewin	Everton	16	33
110	Jamie Vardy	Leicester City	15	34
267	Ollie Watkins	Aston Villa	14	37
33	İlkay Gündoğan	Manchester City	13	28
191	Alexandre Lacazette	Arsenal	13	31

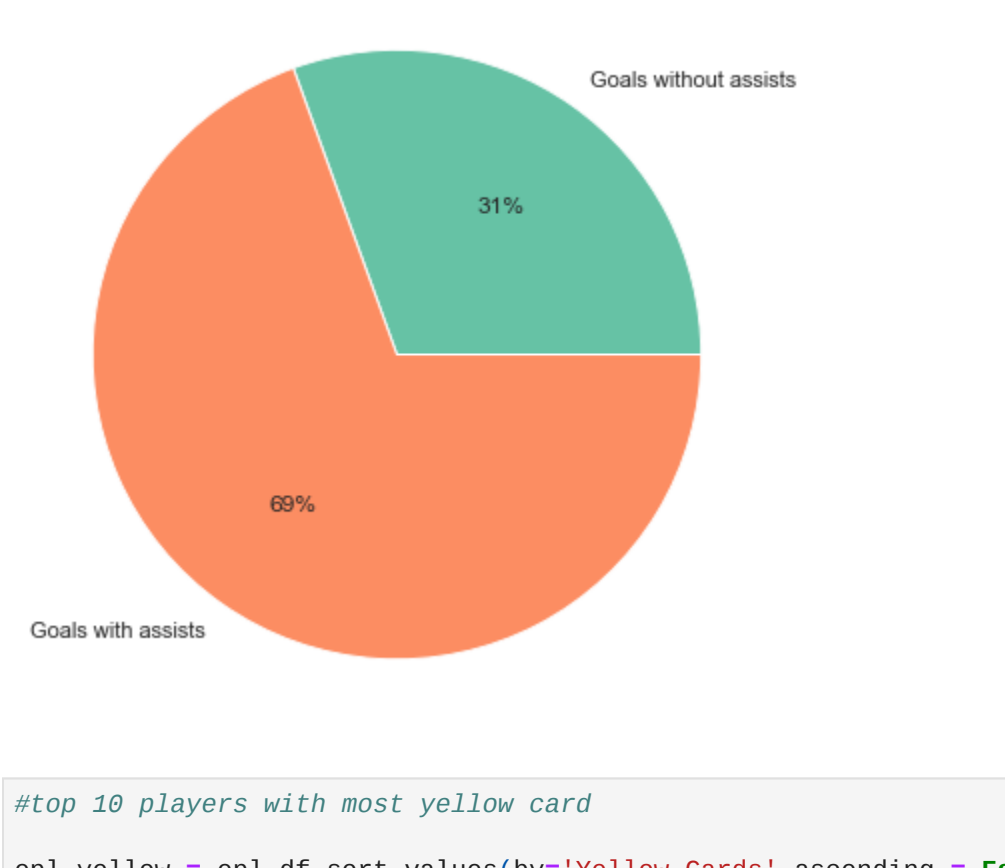
In [71]:
```python
#goals per match

top_10_goals_permatch = epl_df[['Name', 'GoalsPerMatch', 'Matches', 'Goals']].nlargest(n=10, columns='GoalsPerMatch')
top_10_goals_permatch
```

Out[71]:

	Name	GoalsPerMatch	Matches	Goals
162	Harry Kane	0.657143	35	23
81	Mohamed Salah	0.594595	37	22
307	Joe Willock	0.571429	14	8
145	Jesse Lingard	0.562500	16	9
175	Gareth Bale	0.550000	20	11
74	Anthony Elanga	0.500000	2	1
51	Bruno Fernandes	0.486486	37	18
237	Dominic Calvert-Lewin	0.484848	33	16
120	Kelechi Iheanacho	0.480000	25	12
92	Diogo Jota	0.473684	19	9

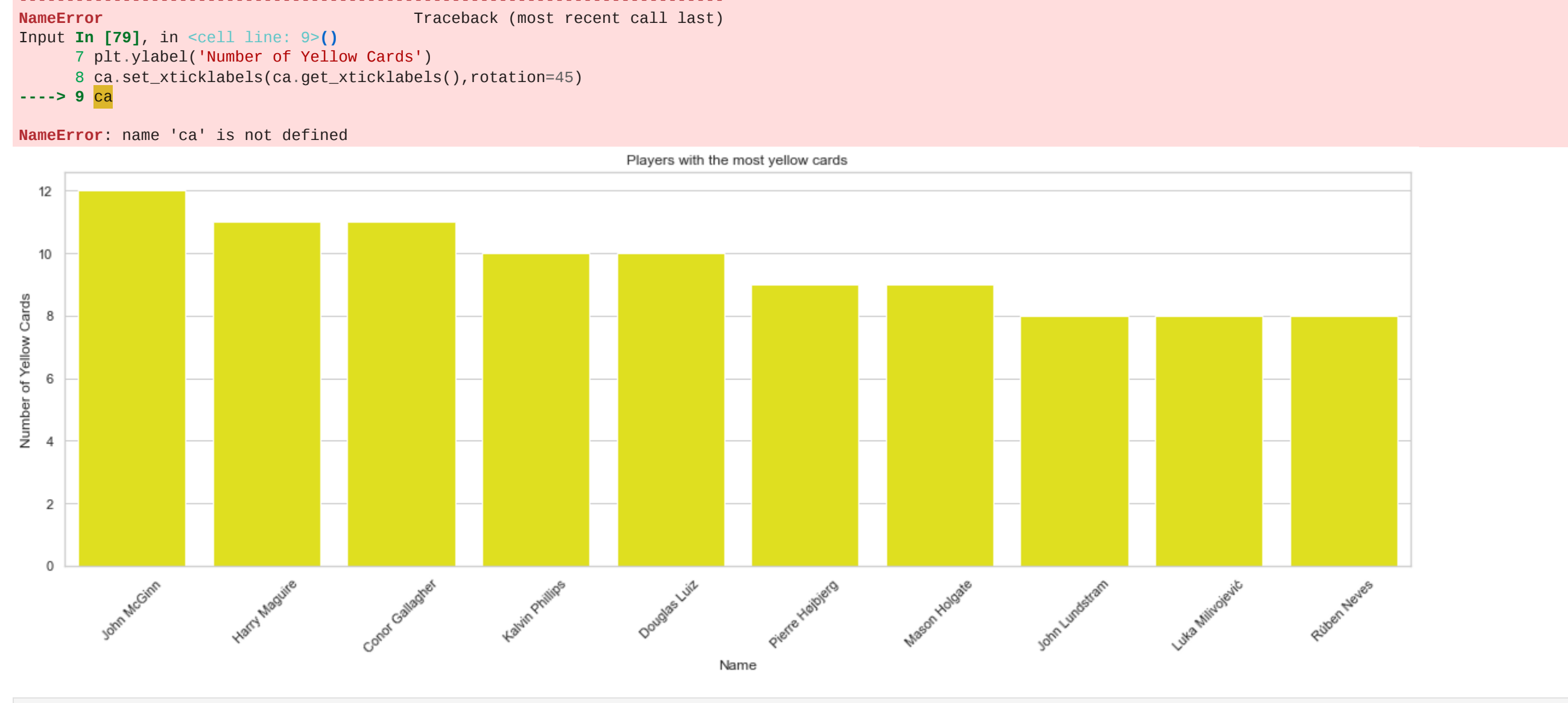
In [75]:
```python
#pie chart - Goals with assist and without assist

plt.figure(figsize=(14,7))
assists = epl_df['Assists'].sum()
data = [Total_Goals - assists, assists]
labels = ['Goals without assists', 'Goals with assists']
color = sns.color_palette('Set2')
plt.pie(data, labels=labels, colors = color, autopct= '%.0f%%')
```

Out[75]:
```
([<matplotlib.patches.Wedge at 0x182793c4100>,
  <matplotlib.patches.Wedge at 0x182793c4820>],
 [Text(0.6317313350719539, 0.9005084787430971, 'Goals without assists'),
  Text(-0.6317313350719536, -0.9005084787430973, 'Goals with assists')],
 [Text(0.3445807282210657, 0.4911864429507802, '31%'),
  Text(-0.3445807282210655, -0.4911864429507803, '69%')])
```



In [79]:
```python
#top 10 players with most yellow card

epl_yellow = epl_df.sort_values(by='Yellow_Cards',ascending = False)[:10]
plt.figure(figsize =(20,6))
plt.title ("Players with the most yellow cards")
ca=sns.barplot(x=epl_yellow['Name'],y=epl_yellow['Yellow_Cards'],label='Players',color="yellow")
plt.ylabel('Number of Yellow Cards')
ca.set_xticklabels(ca.get_xticklabels(),rotation=45)
ca
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [79], in <cell line: 9>()
      7 plt.ylabel('Number of Yellow Cards')
      8 ca.set_xticklabels(ca.get_xticklabels(),rotation=45)
----> 9 ca

NameError: name 'ca' is not defined
```



In [ ]: