

In [1]:

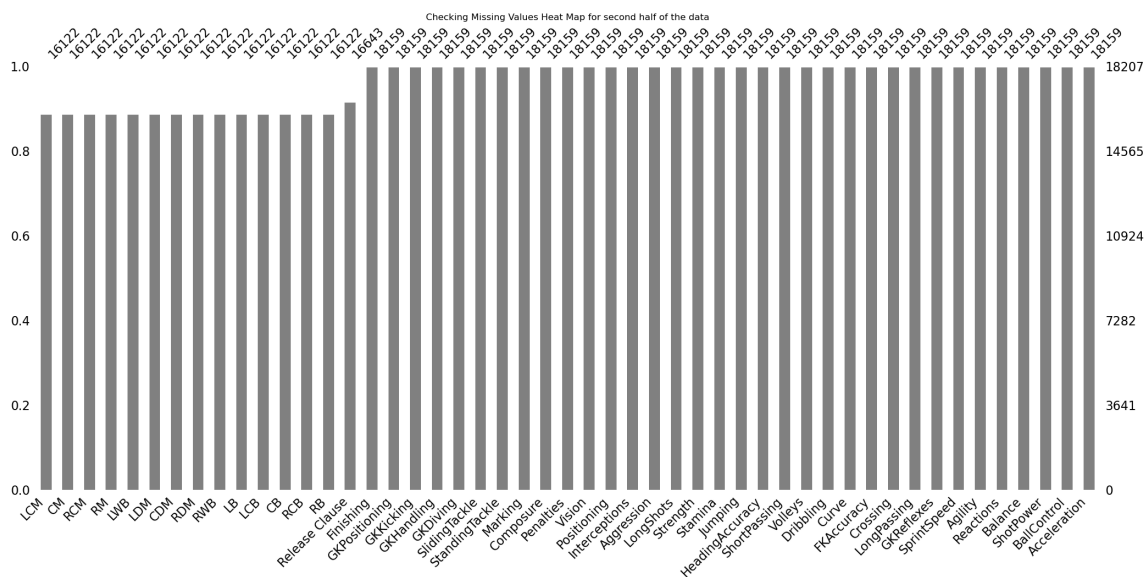
```
1 #Importing Libraries
2
3 import numpy as np
4 import pandas as pd
5 import matplotlib.pyplot as plt
6 import seaborn as sns
7 import missingno as mno
8 import datetime
9 import ipywidgets as widgets
10 from ipywidgets import interact
11 from ipywidgets import interact_manual
12
```

In [2]:

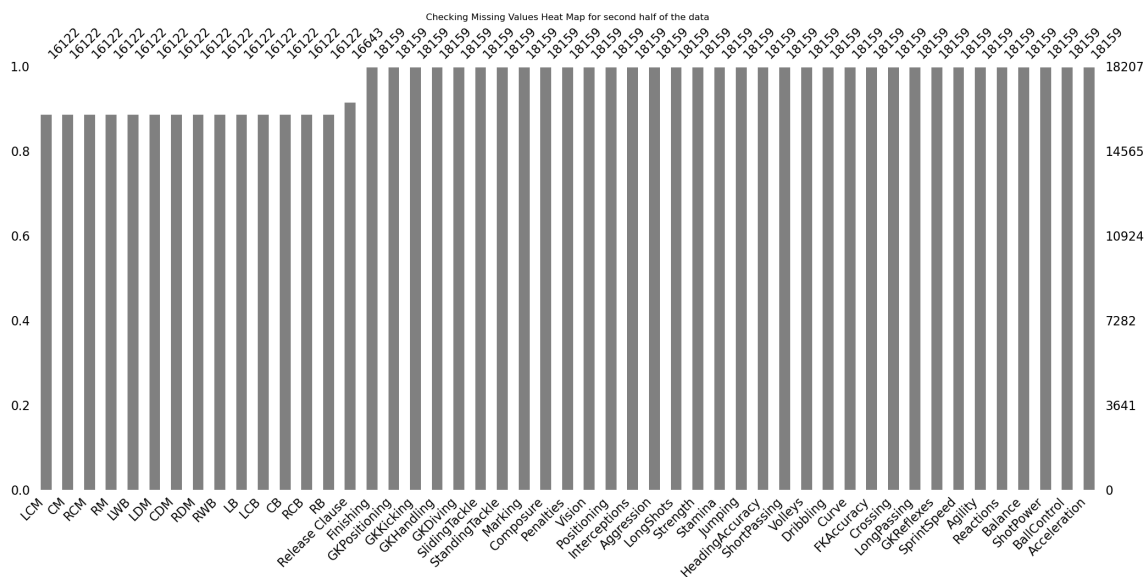
```
1 # Loading the data
2 data = pd.read_csv('D:/5 Data Science Projects/Player Performance Reviewer - Full M
3 #checking rows and columns
4 print(data.shape)
5 print(data.info())
6 print(data.isnull().sum())
```

```
84 GKHandling          18159 non-null float64
85 GKKicking           18159 non-null float64
86 GKPositioning       18159 non-null float64
87 GKReflexes          18159 non-null float64
88 Release Clause      16643 non-null object
dtypes: float64(38), int64(6), object(45)
memory usage: 12.4+ MB
None
Unnamed: 0           0
ID                   0
Name                 0
Age                  0
Photo                0
...
GKHandling           48
GKKicking             48
GKPositioning        48
GKReflexes            48
Release Clause       1564
Length: 89, dtype: int64
```

```
1 # checking NULL values for first 40 columns using missingno and visualizing it
2 mno.bar(data.iloc[:, :40],
3         color = 'grey',
4         sort = 'ascending')
5 plt.title('Checking Missing Values Heat Map for first half of the data')
6 plt.show()
```



```
1 # checking NULL values for remaining columns using missingno and visualizing it
2 mno.bar(data.iloc[:, 40:],
3         color = 'grey',
4         sort = 'ascending')
5 plt.title('Checking Missing Values Heat Map for second half of the data')
6 plt.show()
```



In [5]:

```
1 # filling the missing value with maen,mode and hardcoded values based on column anal
2
3 data['ShortPassing'].fillna(data['ShortPassing'].mean(), inplace = True)
4 data['Volleys'].fillna(data['Volleys'].mean(), inplace = True)
5 data['Dribbling'].fillna(data['Dribbling'].mean(), inplace = True)
6 data['Curve'].fillna(data['Curve'].mean(), inplace = True)
7 data['FKAccuracy'].fillna(data['FKAccuracy'], inplace = True)
8 data['LongPassing'].fillna(data['LongPassing'].mean(), inplace = True)
9 data['BallControl'].fillna(data['BallControl'].mean(), inplace = True)
10 data['HeadingAccuracy'].fillna(data['HeadingAccuracy'].mean(), inplace = True)
11 data['Finishing'].fillna(data['Finishing'].mean(), inplace = True)
12 data['Crossing'].fillna(data['Crossing'].mean(), inplace = True)
13
14 data['Weight'].fillna(data['Weight'].mode(),inplace=True)
15
16 data['Contract Valid Until'].fillna(2019, inplace = True)
17 data['Height'].fillna("5'11", inplace = True)
18 data['Loaned From'].fillna('None', inplace = True)
19 data['Joined'].fillna('Jul 1, 2018', inplace = True)
20 data['Jersey Number'].fillna(8, inplace = True)
21 data['Body Type'].fillna('Normal', inplace = True)
22 data['Position'].fillna('ST', inplace = True)
23 data['Club'].fillna('No Club', inplace = True)
24 data['Work Rate'].fillna('Medium/ Medium', inplace = True)
25 data['Skill Moves'].fillna(data['Skill Moves'].median(), inplace = True)
26 data['Weak Foot'].fillna(3, inplace = True)
27 data['Preferred Foot'].fillna('Right', inplace = True)
28 data['International Reputation'].fillna(1, inplace = True)
29 data['Wage'].fillna('€200K', inplace = True)
30
31 #filling remaining NULL values with value 0
32 data.fillna(0, inplace = True)
33
34 #checking missing values
35 data.isnull().sum().sum()
```

Out[5]:

0

In [6]:

```
1  # creating new features by aggregating the features
2
3  def defending(data):
4      return int(round((data[['Marking', 'StandingTackle',
5                              'SlidingTackle']].mean()).mean()))
6
7  def general(data):
8      return int(round((data[['HeadingAccuracy', 'Dribbling', 'Curve',
9                              'BallControl']].mean()).mean()))
10
11 def mental(data):
12     return int(round((data[['Aggression', 'Interceptions', 'Positioning',
13                             'Vision', 'Composure']].mean()).mean()))
14
15 def passing(data):
16     return int(round((data[['Crossing', 'ShortPassing',
17                             'LongPassing']].mean()).mean()))
18
19 def mobility(data):
20     return int(round((data[['Acceleration', 'SprintSpeed',
21                             'Agility', 'Reactions']].mean()).mean()))
22
23 def power(data):
24     return int(round((data[['Balance', 'Jumping', 'Stamina',
25                             'Strength']].mean()).mean()))
26
27 def rating(data):
28     return int(round((data[['Potential', 'Overall']].mean()).mean()))
29
30 def shooting(data):
31     return int(round((data[['Finishing', 'Volleys', 'FKAccuracy',
32                             'ShotPower', 'LongShots', 'Penalties']].mean()).mean()))
```

In [7]:

```

1  # adding these categories to the data
2
3  data['Defending'] = data.apply(defending, axis = 1)
4  data['General'] = data.apply(general, axis = 1)
5  data['Mental'] = data.apply(mental, axis = 1)
6  data['Passing'] = data.apply(passing, axis = 1)
7  data['Mobility'] = data.apply(mobility, axis = 1)
8  data['Power'] = data.apply(power, axis = 1)
9  data['Rating'] = data.apply(rating, axis = 1)
10 data['Shooting'] = data.apply(shooting, axis = 1)
11
12 # Lets check the column names in the data after adding new features
13 data.columns

```

Out[7]:

```

Index(['Unnamed: 0', 'ID', 'Name', 'Age', 'Photo', 'Nationality', 'Flag',
      'Overall', 'Potential', 'Club', 'Club Logo', 'Value', 'Wage', 'Spec
      ial',
      'Preferred Foot', 'International Reputation', 'Weak Foot',
      'Skill Moves', 'Work Rate', 'Body Type', 'Real Face', 'Position',
      'Jersey Number', 'Joined', 'Loaned From', 'Contract Valid Until',
      'Height', 'Weight', 'LS', 'ST', 'RS', 'LW', 'LF', 'CF', 'RF', 'RW',
      'LAM', 'CAM', 'RAM', 'LM', 'LCM', 'CM', 'RCM', 'RM', 'LWB', 'LDM',
      'CDM', 'RDM', 'RWB', 'LB', 'LCB', 'CB', 'RCB', 'RB', 'Crossing',
      'Finishing', 'HeadingAccuracy', 'ShortPassing', 'Volleys', 'Dribbli
      ng',
      'Curve', 'FKAccuracy', 'LongPassing', 'BallControl', 'Acceleratio
      n',
      'SprintSpeed', 'Agility', 'Reactions', 'Balance', 'ShotPower',
      'Jumping', 'Stamina', 'Strength', 'LongShots', 'Aggression',
      'Interceptions', 'Positioning', 'Vision', 'Penalties', 'Composure',
      'Marking', 'StandingTackle', 'SlidingTackle', 'GKDividing', 'GKHandli
      ng',
      'GKKicking', 'GKPositioning', 'GKReflexes', 'Release Clause',
      'Defending', 'General', 'Mental', 'Passing', 'Mobility', 'Power',
      'Rating', 'Shooting'],
      dtype='object')

```

In [8]:

```

1  #Distribution of Scores of Different Skills based on feature engineering
2
3  plt.rcParams['figure.figsize'] = (18, 12)
4  plt.subplot(2, 4, 1)
5  sns.distplot(data['Defending'], color = 'red')
6  plt.grid()
7
8  plt.subplot(2, 4, 2)
9  sns.distplot(data['General'], color = 'black')
10 plt.grid()
11
12 plt.subplot(2, 4, 3)
13 sns.distplot(data['Mental'], color = 'red')
14 plt.grid()
15
16 plt.subplot(2, 4, 4)
17 sns.distplot(data['Passing'], color = 'black')
18 plt.grid()
19
20 plt.subplot(2, 4, 5)
21 sns.distplot(data['Mobility'], color = 'red')
22 plt.grid()
23
24 plt.subplot(2, 4, 6)
25 sns.distplot(data['Power'], color = 'black')
26 plt.grid()
27
28 plt.subplot(2, 4, 7)
29 sns.distplot(data['Shooting'], color = 'red')
30 plt.grid()
31
32 plt.subplot(2, 4, 8)
33 sns.distplot(data['Rating'], color = 'black')
34 plt.grid()
35
36 plt.suptitle('Score Distributions for Different Abilities')
37 plt.show()

```

C:\Users\Pradeep\AppData\Local\Temp\ipykernel_15780\333795502.py:5: Use
rWarning:

`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).

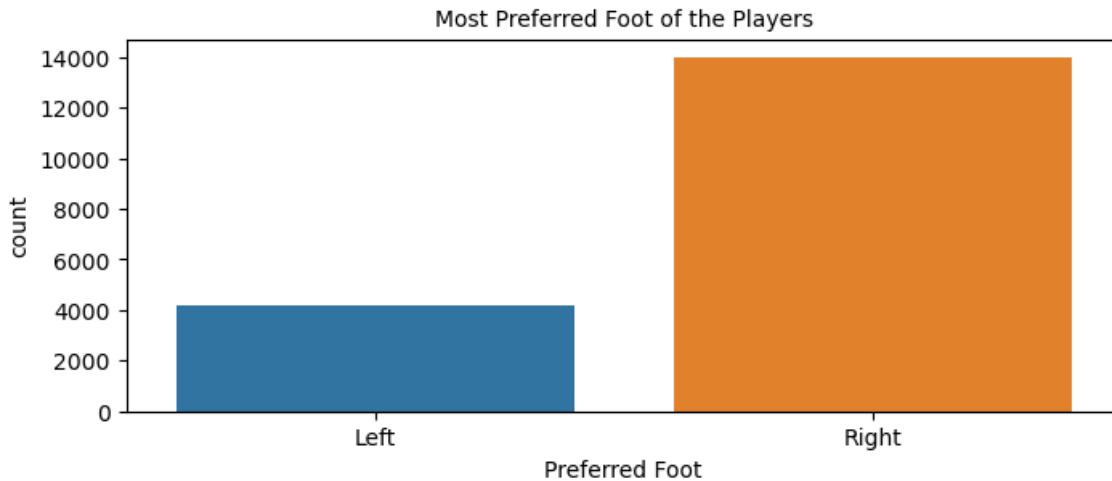
For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751> (<http://s://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>)

sns.distplot(data['Defending'], color = 'red')
C:\Users\Pradeep\AppData\Local\Temp\ipykernel_15780\333795502.py:9: Use
rWarning:

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

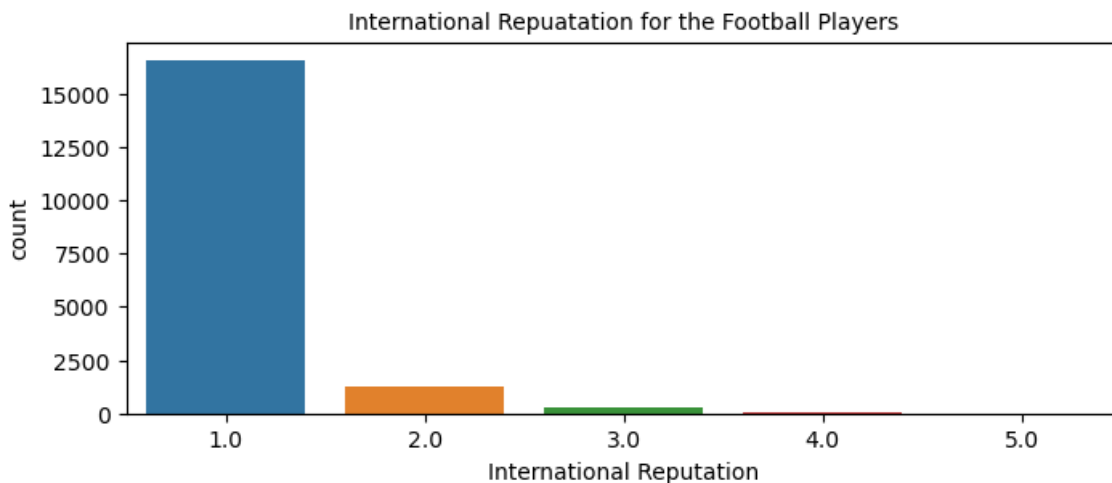
In [9]:

```
1 # comparison of preferred foot over the different players
2
3 plt.rcParams['figure.figsize'] = (8, 3)
4 sns.countplot(x=data['Preferred Foot'])
5 plt.title('Most Preferred Foot of the Players', fontsize = 10)
6 plt.show()
```



In [10]:

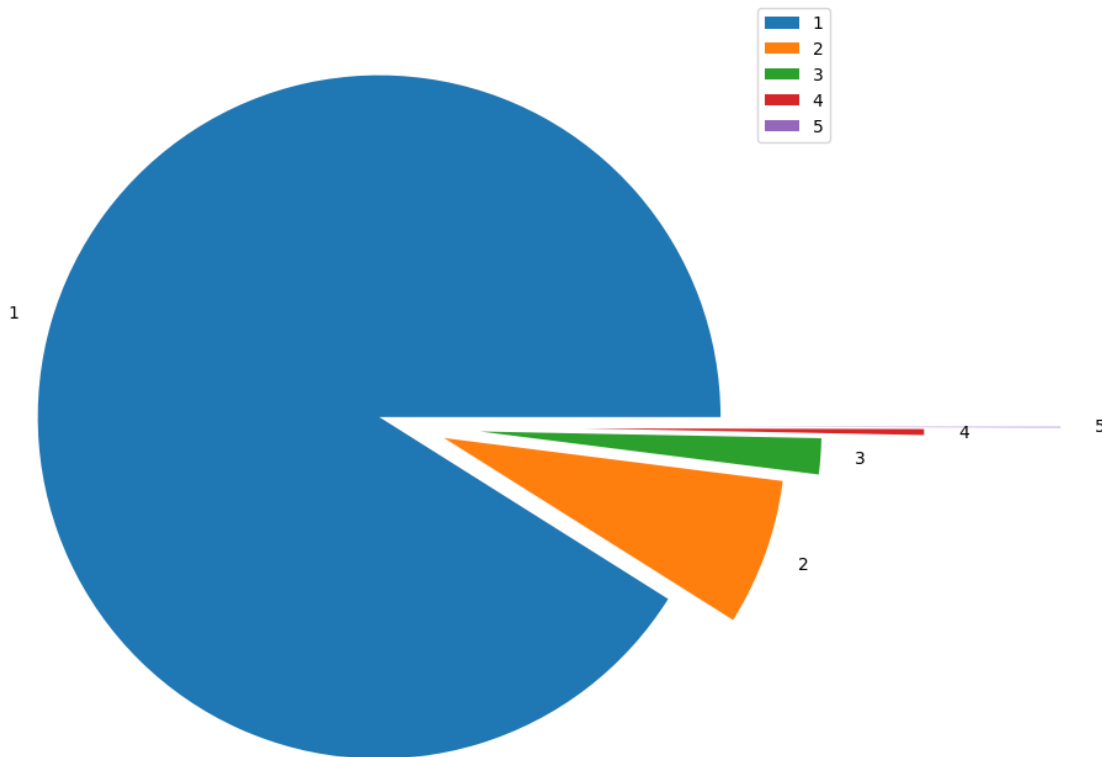
```
1 # plotting countplot to represent share of international reputation
2
3 plt.rcParams['figure.figsize'] = (8, 3)
4 sns.countplot(x=data['International Reputation'])
5 plt.title('International Reputation for the Football Players', fontsize = 10)
6 plt.show()
```



In [11]:

```
1 # plotting a pie chart to represent share of international reputation
2
3 labels = ['1', '2', '3', '4', '5'] #data['International Reputation'].index
4 sizes = data['International Reputation'].value_counts()
5 explode = [0.1, 0.1, 0.2, 0.5, 0.9]
6
7 plt.rcParams['figure.figsize'] = (9, 9)
8 plt.pie(sizes, labels = labels, explode = explode)
9 plt.title('International Reputation for the Football Players', fontsize = 20)
10 plt.legend()
11 plt.show()
```

International Reputation for the Football Players



In [12]:

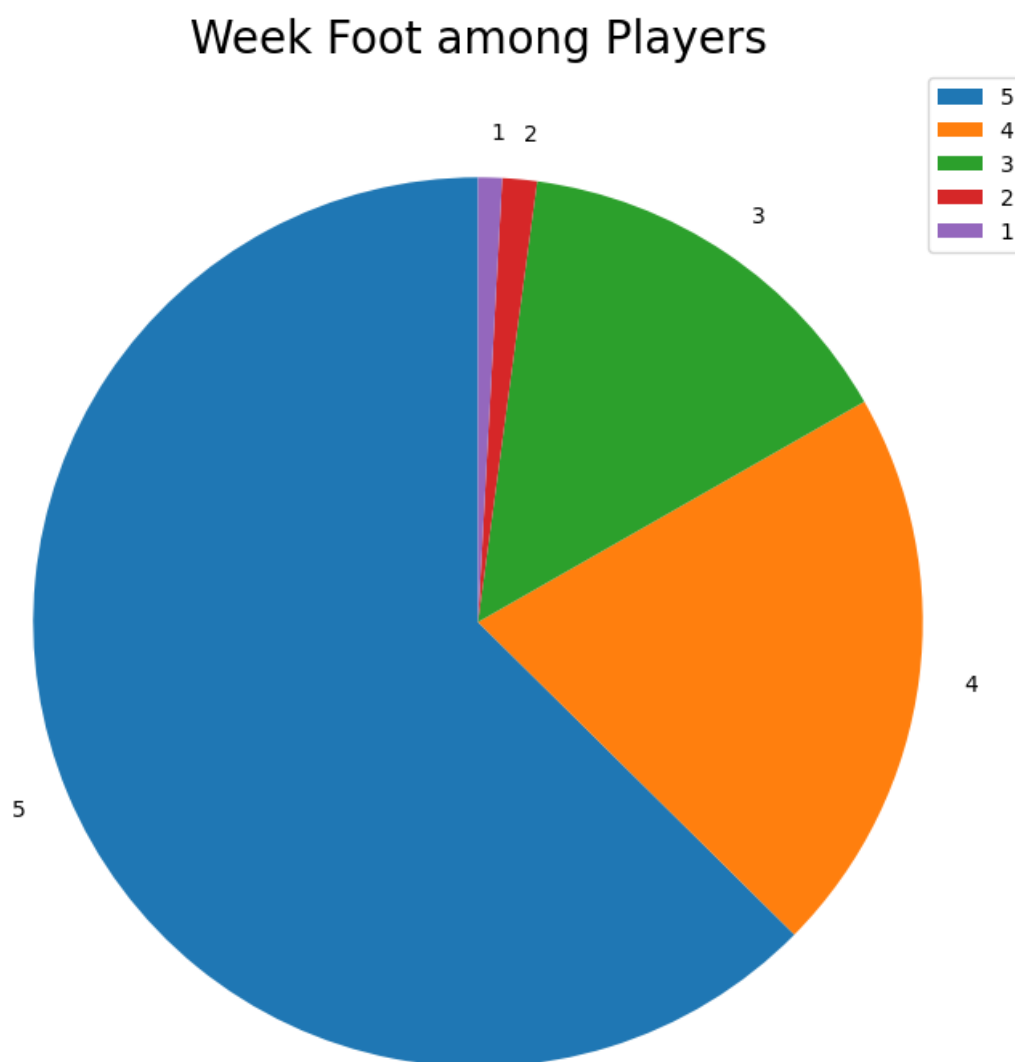
```
1 #viewing selected columns of 5star rated reputed p
2 data[data['International Reputation'] == 5][['Name', 'Nationality',
3                                             'Overall']]
```

Out[12]:

	Name	Nationality	Overall
0	L. Messi	Argentina	94
1	Cristiano Ronaldo	Portugal	94
2	Neymar Jr	Brazil	92
7	L. Suárez	Uruguay	91
22	M. Neuer	Germany	89
109	Z. Ibrahimović	Sweden	85

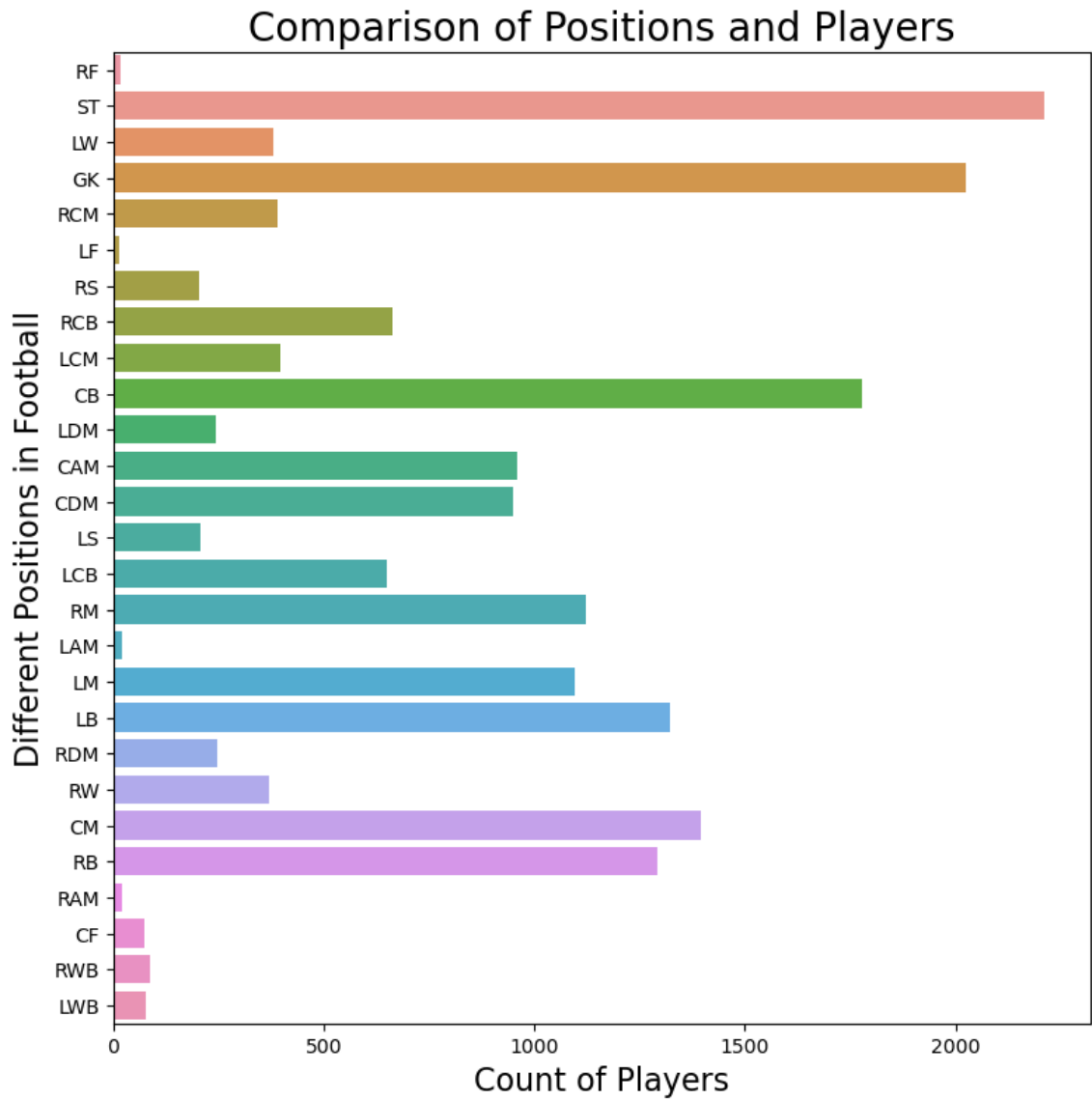
In [13]:

```
1 # plotting a pie chart to represent Week Foot among Players
2
3 labels = ['5','4','3','2','1']
4 sizes = data['Weak Foot'].value_counts()
5
6 plt.rcParams['figure.figsize'] = (9, 9)
7 plt.pie(sizes, labels = labels, startangle = 90)
8 plt.title('Week Foot among Players', fontsize = 20)
9 plt.legend()
10 plt.show()
```



In [14]:

```
1 # different positions acquired by the players
2 plt.figure(figsize = (9,9))
3 ax = sns.countplot(y = 'Position', data = data)
4 ax.set_xlabel(xlabel = 'Count of Players', fontsize = 16)
5 ax.set_ylabel(ylabel = 'Different Positions in Football', fontsize = 16)
6 ax.set_title(label = 'Comparison of Positions and Players', fontsize = 20)
7 plt.show()
8
```



In [15]:

```
1 # defining a function for cleaning the Weight data
2 def extract(value):
3     a = str(value).replace('lbs', '')
4     return float(a)
5
6 # applying the function to weight column
7 data['Weight'] = data['Weight'].apply(lambda x : extract(x))
8
9 # plotting the distribution of weight of the players
10 plt.rcParams['figure.figsize'] = (16,16)
11 sns.distplot(data['Weight'])
12 plt.title("Distribution of Players Weight", fontsize = 15)
13 plt.show()
```

C:\Users\Pradeep\AppData\Local\Temp\ipykernel_15780\1451629379.py:11: UserWarning:

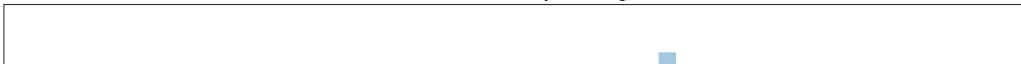
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

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> (<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>)

```
sns.distplot(data['Weight'])
```

Distribution of Players Weight



In [16]:

```
1  # defining a function for cleaning the wage column
2
3  def extract(column):
4      column = str(column)
5      out = column.replace('€', '')
6      if 'M' in out:
7          out = float(out.replace('M', ''))/1000
8      elif 'K' in column:
9          out = float(out.replace('K', ''))
10     return float(out)
11
12
13  # applying the function to the wage and value column
14  data['Value'] = data['Value'].apply(lambda x: extract(x))
15  data['Wage'] = data['Wage'].apply(lambda x: extract(x))
16
17  # visualizing the data
18  plt.rcParams['figure.figsize'] = (16, 5)
19  plt.subplot(1, 2, 1)
20  sns.distplot(data['Value'])
21  plt.title('Distribution of Value of the Players', fontsize = 15)
22
23  plt.subplot(1, 2, 2)
24  sns.distplot(data['Wage'])
25  plt.title('Distribution of Wages of the Players', fontsize = 15)
26  plt.show()
```

C:\Users\Pradeep\AppData\Local\Temp\ipykernel_15780\1663190044.py:20: User Warning:

`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> (<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>)

```
sns.distplot(data['Value'])
```

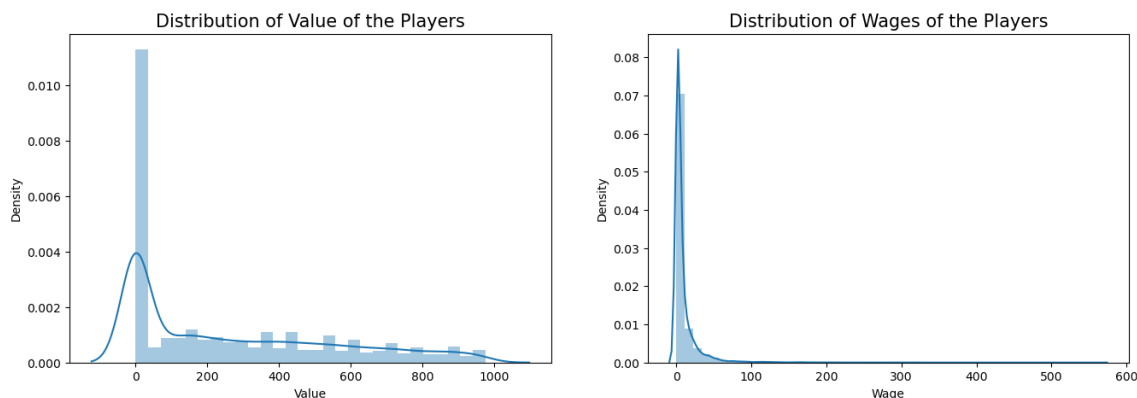
C:\Users\Pradeep\AppData\Local\Temp\ipykernel_15780\1663190044.py:24: User Warning:

`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> (<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>)

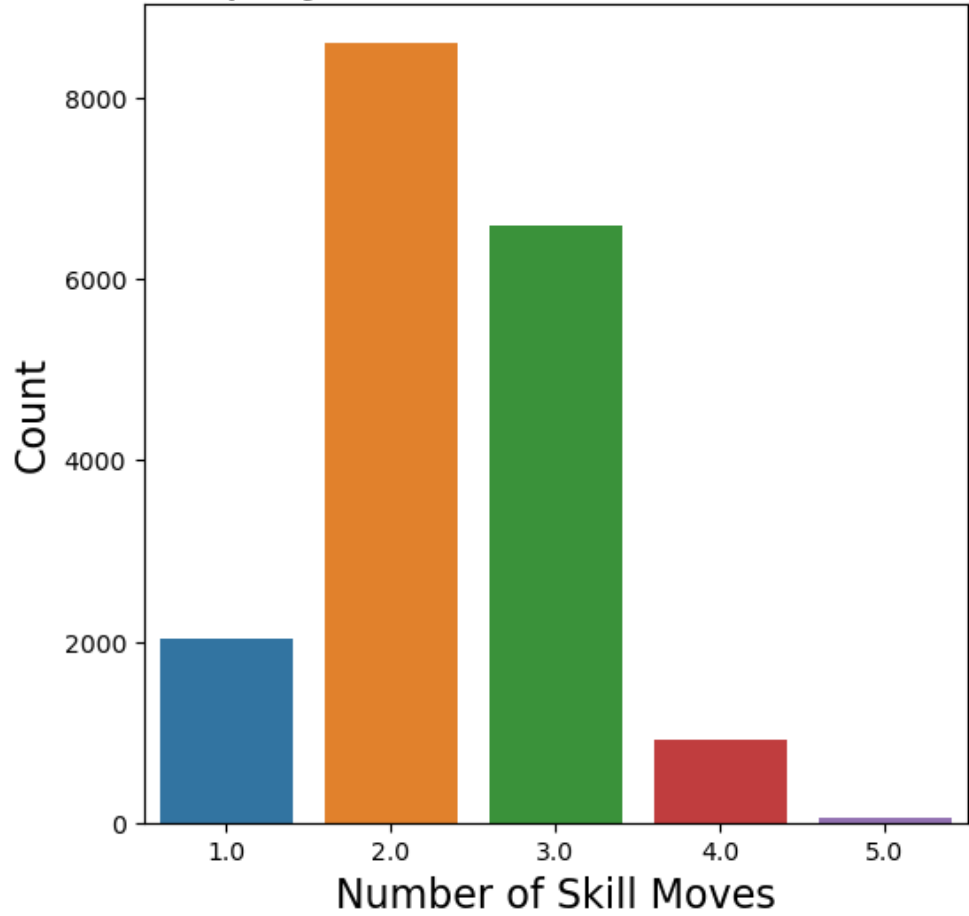
```
sns.distplot(data['Wage'])
```



In [17]:

```
1 # Skill Moves of Players
2
3 plt.figure(figsize = (6, 6))
4 ax = sns.countplot(x = 'Skill Moves', data = data)
5 ax.set_title(label = 'Count of players on Basis of their skill moves', fontsize = 20)
6 ax.set_xlabel(xlabel = 'Number of Skill Moves', fontsize = 16)
7 ax.set_ylabel(ylabel = 'Count', fontsize = 16)
8 plt.show()
9
10 #viewing skills with value 5-
11 data[(data['Skill Moves'] == 5.0) & (data['Age'] < 20)][['Name', 'Age']]
```

Count of players on Basis of their skill moves

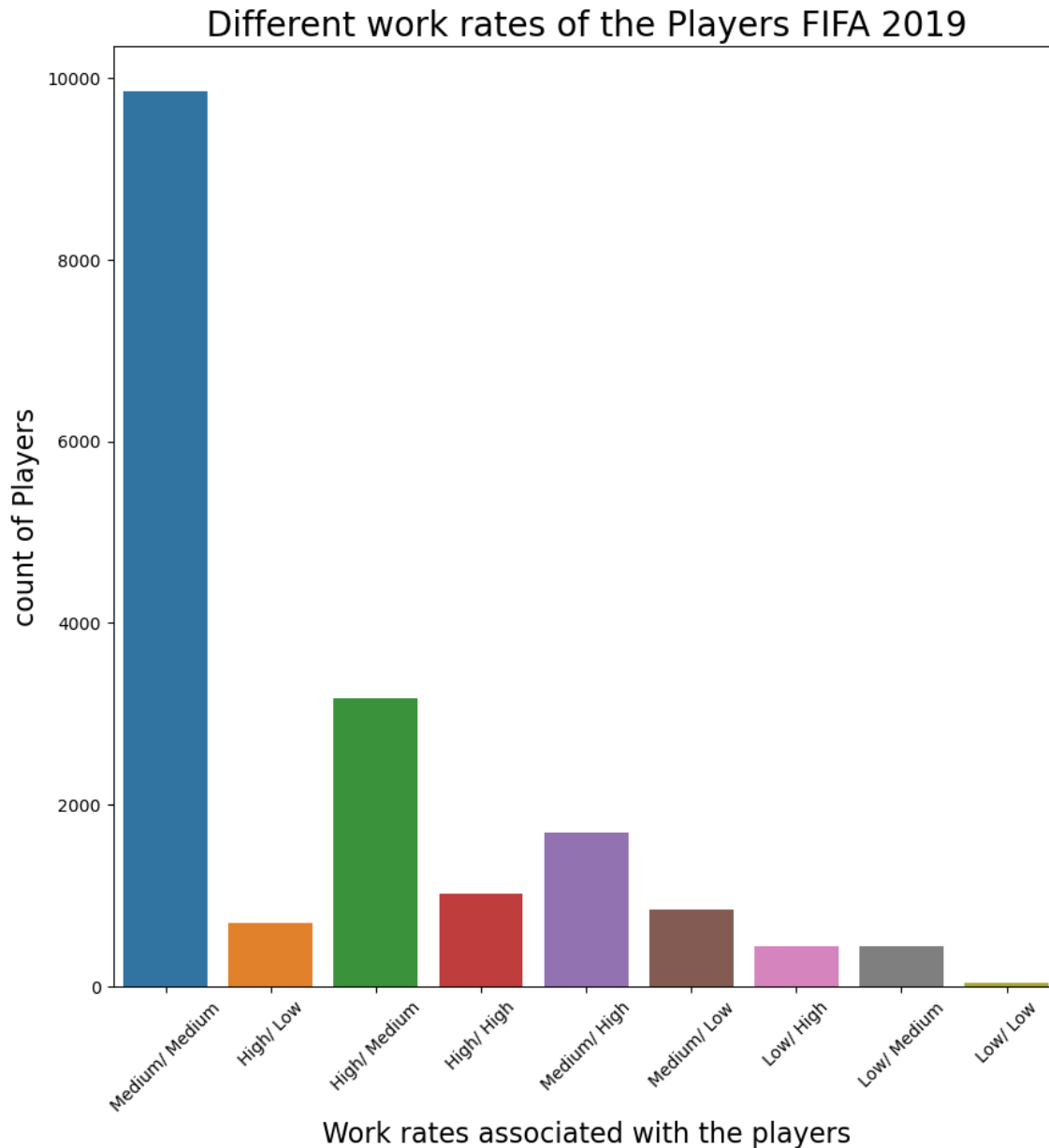


Out[17]:

	Name	Age
25	K. Mbappé	19
1004	J. Sancho	18
1143	Vinícius Júnior	17
2495	M. Ødegaard	19

In [18]:

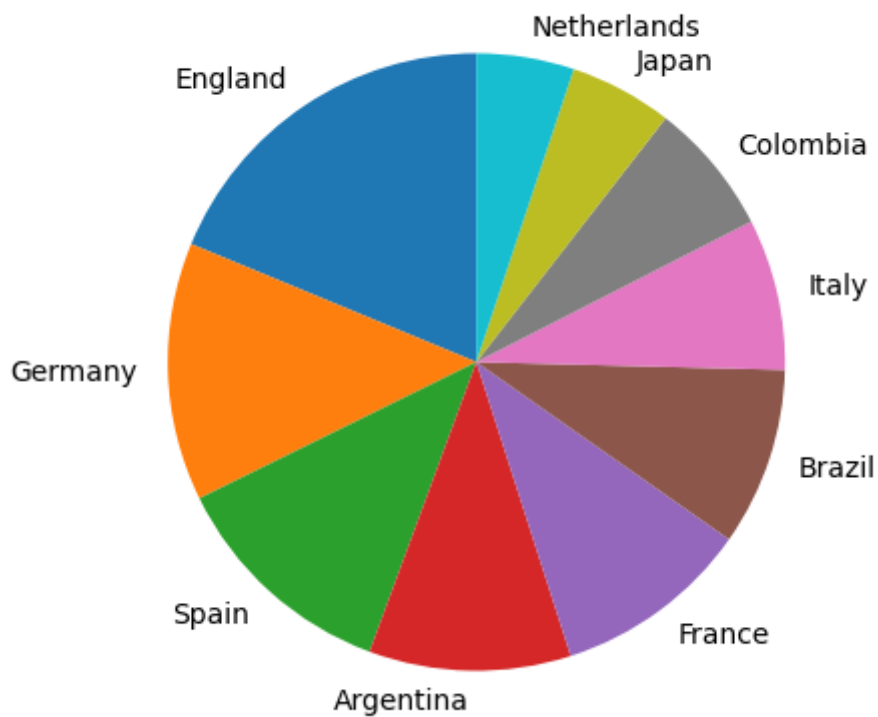
```
1 #Different Work rate of the players participating in the FIFA
2 plt.figure(figsize = (10,10))
3
4 sns.countplot(x = 'Work Rate', data = data)
5 plt.title('Different work rates of the Players FIFA 2019', fontsize = 20)
6 plt.xlabel('Work rates associated with the players', fontsize = 16)
7 plt.ylabel('count of Players', fontsize = 16)
8 plt.xticks(rotation = 45)
9 plt.show()
```



In [19]:

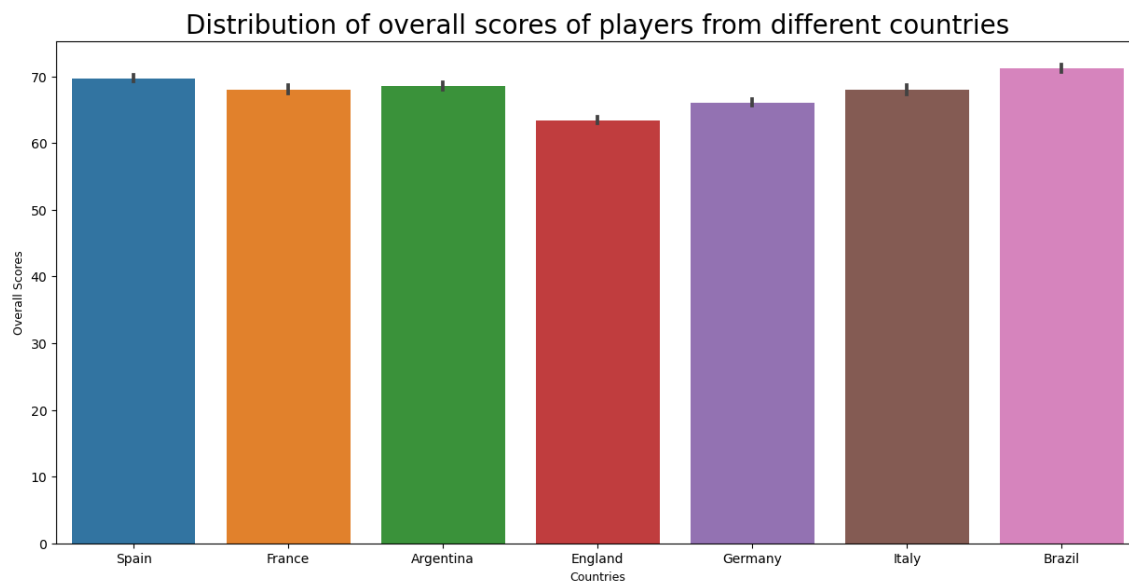
```
1 # Top 10 countries with highest number of players
2
3 data['Nationality'].value_counts().head(10).plot(kind = 'pie',startangle =90)
4 plt.title('Countries having Highest Number of players', fontsize = 15)
5 plt.axis('off')
6 plt.show()
```

Countries having Highest Number of players



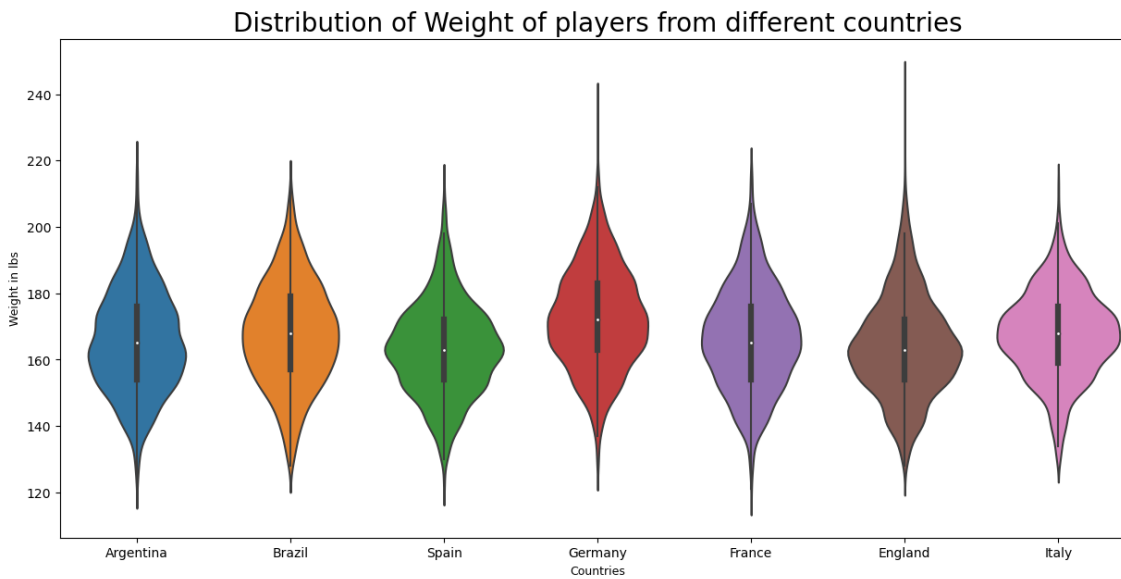
In [20]:

```
1 #Some Nations' Player and their overall scores
2
3 some_countries = ('England', 'Germany', 'Spain', 'Argentina', 'France', 'Brazil', 'I
4 data_countries = data.loc[data['Nationality'].isin(some_countries) & data['Overall']
5
6 plt.rcParams['figure.figsize'] = (15, 7)
7 ax = sns.barplot(x = data_countries['Nationality'], y = data_countries['Overall'])
8 ax.set_xlabel(xlabel = 'Countries', fontsize = 9)
9 ax.set_ylabel(ylabel = 'Overall Scores', fontsize = 9)
10 ax.set_title(label = 'Distribution of overall scores of players from different count
11 plt.show()
```



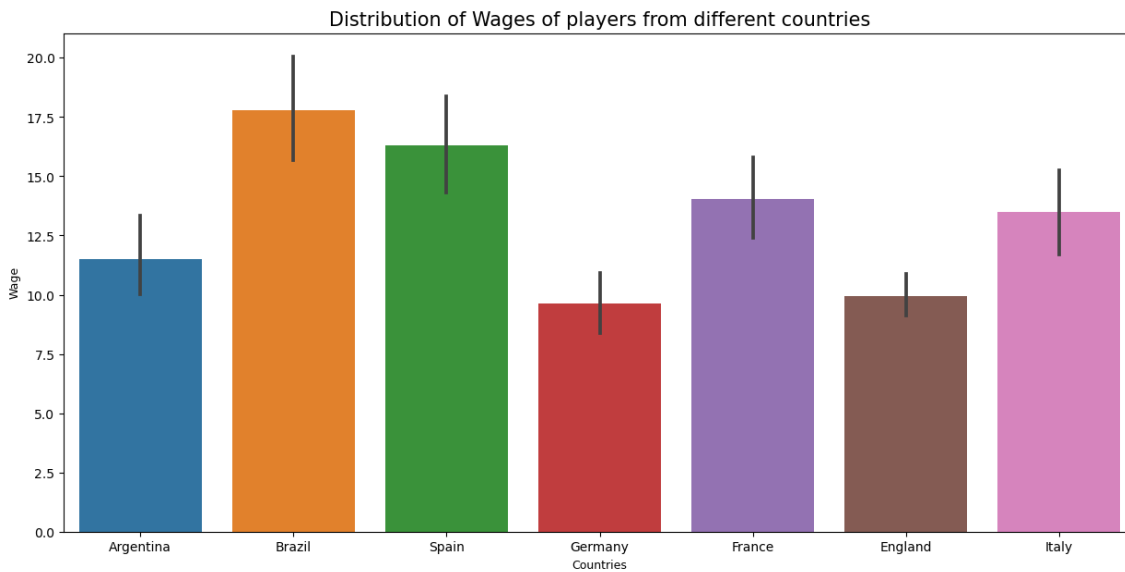
In [21]:

```
1 #Some Nations' Player and their Weights
2
3 some_countries = ('England', 'Germany', 'Spain', 'Argentina', 'France', 'Brazil', 'I
4 data_countries = data.loc[data['Nationality'].isin(some_countries) & data['Weight']]
5
6 plt.rcParams['figure.figsize'] = (15, 7)
7 ax = sns.violinplot(x = data_countries['Nationality'], y = data_countries['Weight'])
8 ax.set_xlabel(xlabel = 'Countries', fontsize = 9)
9 ax.set_ylabel(ylabel = 'Weight in lbs', fontsize = 9)
10 ax.set_title(label = 'Distribution of Weight of players from different countries', f
11 plt.show()
```



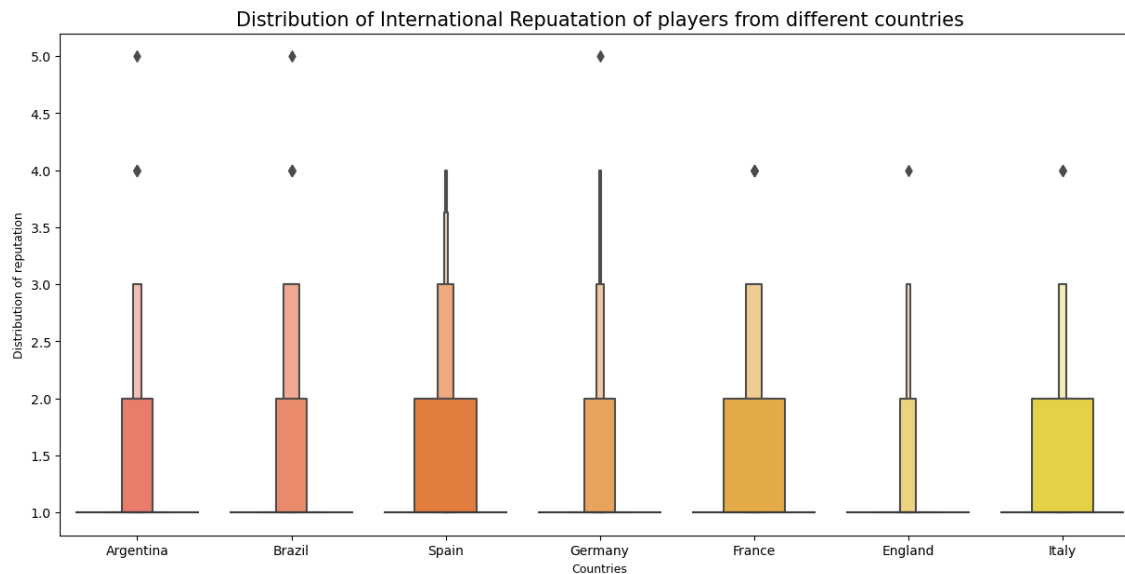
In [22]:

```
1 #Some Nations' Player and their wages
2
3 some_countries = ('England', 'Germany', 'Spain', 'Argentina', 'France', 'Brazil', 'I
4 data_countries = data.loc[data['Nationality'].isin(some_countries) & data['Wage']]
5
6 plt.rcParams['figure.figsize'] = (15, 7)
7 ax = sns.barplot(x = data_countries['Nationality'], y = data_countries['Wage'])
8 ax.set_xlabel(xlabel = 'Countries', fontsize = 9)
9 ax.set_ylabel(ylabel = 'Wage', fontsize = 9)
10 ax.set_title(label = 'Distribution of Wages of players from different countries', fo
11 plt.show()
```



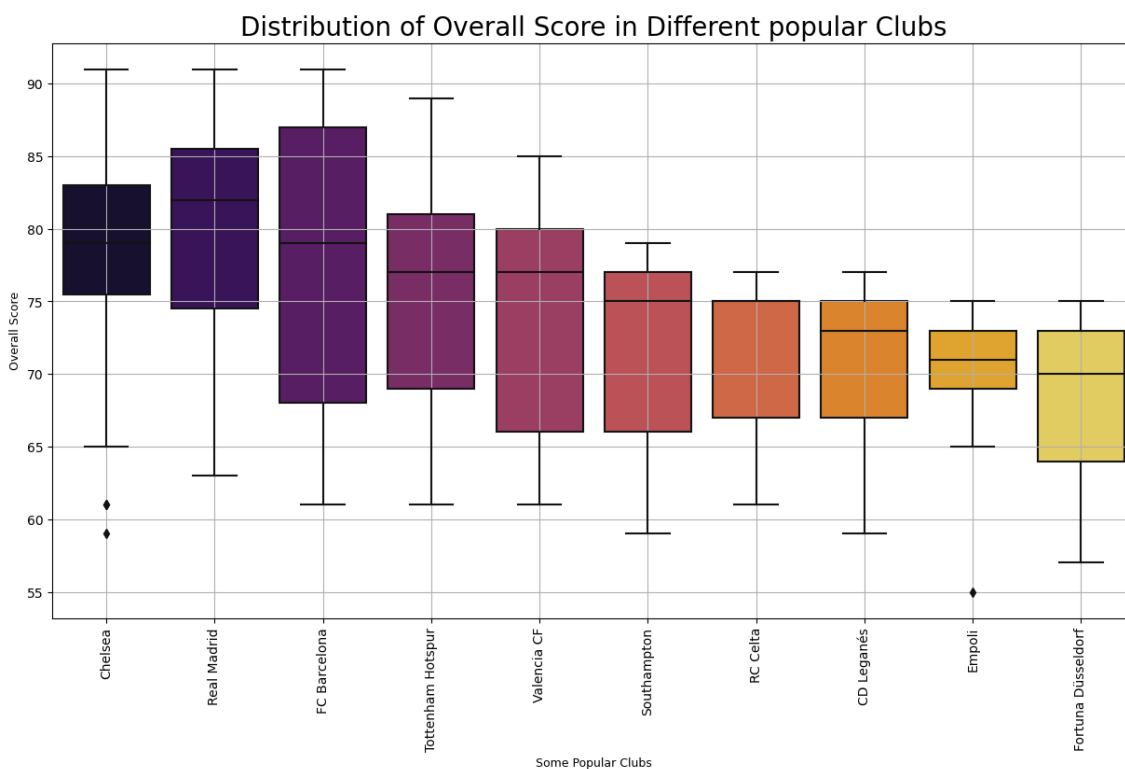
In [23]:

```
1 #Some Nations' Player and their International Reputation
2
3 some_countries = ('England', 'Germany', 'Spain', 'Argentina', 'France', 'Brazil', 'I
4 data_countries = data.loc[data['Nationality'].isin(some_countries) & data['Internati
5
6 plt.rcParams['figure.figsize'] = (15, 7)
7 ax = sns.boxenplot(x = data_countries['Nationality'], y = data_countries['Internatio
8 ax.set_xlabel(xlabel = 'Countries', fontsize = 9)
9 ax.set_ylabel(ylabel = 'Distribution of reputation', fontsize = 9)
10 ax.set_title(label = 'Distribution of International Reputatation of players from diff
11 plt.show()
```



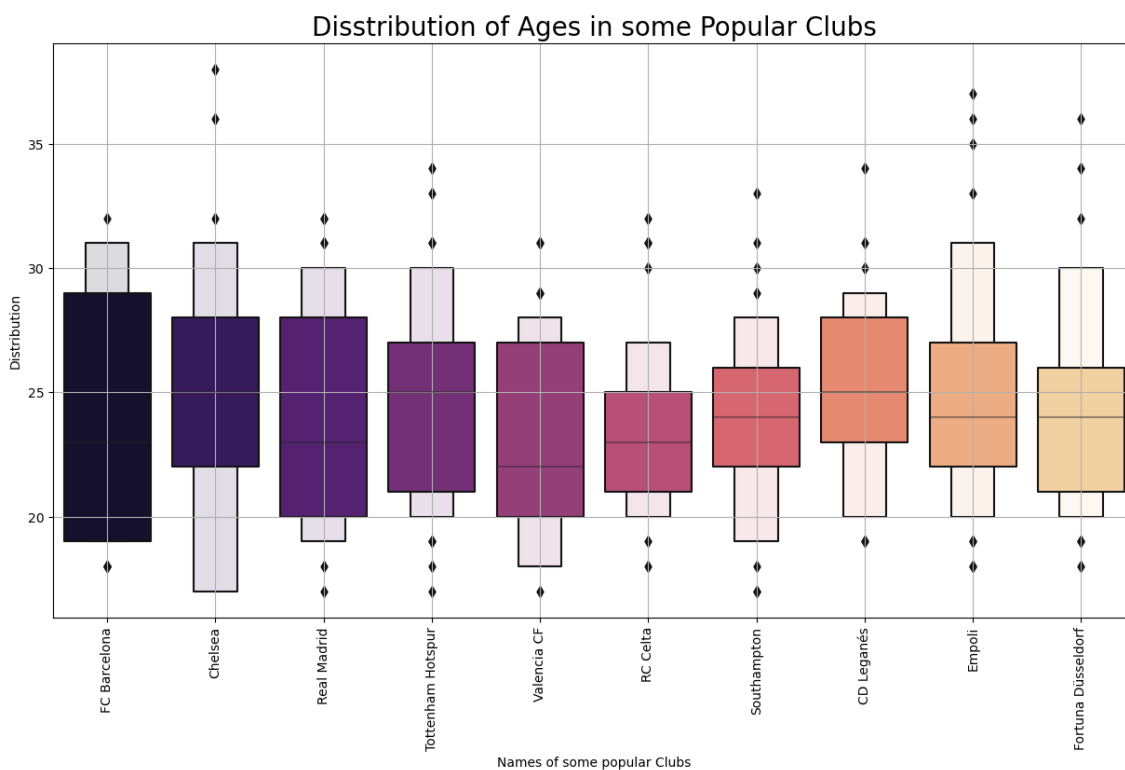
In [24]:

```
1 # Some clubs and their overall score
2
3 some_clubs = ('CD Leganés', 'Southampton', 'RC Celta', 'Empoli', 'Fortuna Düsseldorf',
4              'Tottenham Hotspur', 'FC Barcelona', 'Valencia CF', 'Chelsea', 'Real Ma
5
6 data_clubs = data.loc[data['Club'].isin(some_clubs) & data['Overall']]
7
8 plt.rcParams['figure.figsize'] = (15, 8)
9 ax = sns.boxplot(x = data_clubs['Club'], y = data_clubs['Overall'], palette = 'infer
10 ax.set_xlabel(xlabel = 'Some Popular Clubs', fontsize = 9)
11 ax.set_ylabel(ylabel = 'Overall Score', fontsize = 9)
12 ax.set_title(label = 'Distribution of Overall Score in Different popular Clubs', for
13 plt.xticks(rotation = 90)
14 plt.grid()
15 plt.show()
```



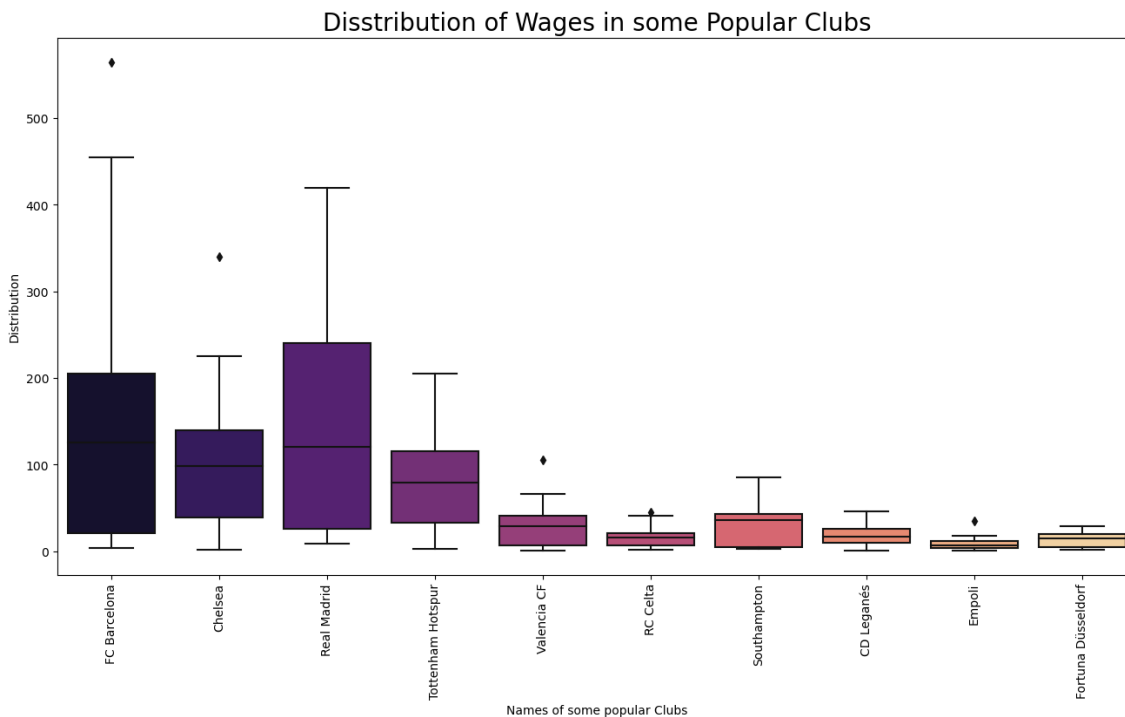
In [25]:

```
1 # Distribution of Ages in some Popular clubs
2
3 some_clubs = ('CD Leganés', 'Southampton', 'RC Celta', 'Empoli', 'Fortuna Düsseldorf',
4              'Tottenham Hotspur', 'FC Barcelona', 'Valencia CF', 'Chelsea', 'Real Madrid')
5
6 data_club = data.loc[data['Club'].isin(some_clubs) & data['Wage']]
7
8 plt.rcParams['figure.figsize'] = (15, 8)
9 ax = sns.boxenplot(x = 'Club', y = 'Age', data = data_club, palette = 'magma')
10 ax.set_xlabel(xlabel = 'Names of some popular Clubs', fontsize = 10)
11 ax.set_ylabel(ylabel = 'Distribution', fontsize = 10)
12 ax.set_title(label = 'Disribution of Ages in some Popular Clubs', fontsize = 20)
13 plt.xticks(rotation = 90)
14 plt.grid()
15 plt.show()
16
```



In [26]:

```
1 # Distribution of Wages in some Popular Clubs
2
3 some_clubs = ('CD Leganés', 'Southampton', 'RC Celta', 'Empoli', 'Fortuna Düsseldorf',
4               'Tottenham Hotspur', 'FC Barcelona', 'Valencia CF', 'Chelsea', 'Real Madrid')
5
6 data_club = data.loc[data['Club'].isin(some_clubs) & data['Wage']]
7
8 plt.rcParams['figure.figsize'] = (16, 8)
9 ax = sns.boxplot(x = 'Club', y = 'Wage', data = data_club, palette = 'magma')
10 ax.set_xlabel(xlabel = 'Names of some popular Clubs', fontsize = 10)
11 ax.set_ylabel(ylabel = 'Distribution', fontsize = 10)
12 ax.set_title(label = 'Disribution of Wages in some Popular Clubs', fontsize = 20)
13 plt.xticks(rotation = 90)
14 plt.show()
```



In [27]:

```
1 #best players per each position with their age, club, and nationality based on their
2
3 data.iloc[data.groupby(data['Position'])['Overall'].idxmax()][['Position', 'Name', '
4     'Nationality', 'Overall']].sort_values(by = 'Overall',
5     ascending = False)
```

Out[27]:

	Position	Name	Age	Club	Nationality	Overall
1	ST	Cristiano Ronaldo	33	Juventus	Portugal	94
0	RF	L. Messi	31	FC Barcelona	Argentina	94
2	LW	Neymar Jr	26	Paris Saint-Germain	Brazil	92
5	LF	E. Hazard	27	Chelsea	Belgium	91
7	RS	L. Suárez	31	FC Barcelona	Uruguay	91
4	RCM	K. De Bruyne	27	Manchester City	Belgium	91
3	GK	De Gea	27	Manchester United	Spain	91
8	RCB	Sergio Ramos	32	Real Madrid	Spain	91
12	CB	D. Godín	32	Atlético Madrid	Uruguay	90
11	LCM	T. Kroos	28	Real Madrid	Germany	90
17	CAM	A. Griezmann	27	Atlético Madrid	France	89
14	LDM	N. Kanté	27	Chelsea	France	89
24	LCB	G. Chiellini	33	Juventus	Italy	89
20	CDM	Sergio Busquets	29	FC Barcelona	Spain	89
21	LS	E. Cavani	31	Paris Saint-Germain	Uruguay	89
33	LM	P. Aubameyang	29	Arsenal	Gabon	88
35	LB	Marcelo	30	Real Madrid	Brazil	88
28	LAM	J. Rodríguez	26	FC Bayern München	Colombia	88
25	RM	K. Mbappé	19	Paris Saint-Germain	France	88
45	RDM	P. Pogba	25	Manchester United	France	87
69	RB	Azpilicueta	28	Chelsea	Spain	86
67	CM	Thiago	27	FC Bayern München	Spain	86
56	RW	Bernardo Silva	23	Manchester City	Portugal	86
129	RAM	J. Cuadrado	30	Juventus	Colombia	84
271	CF	Luis Alberto	25	Lazio	Spain	82
474	LWB	N. Schulz	25	TSG 1899 Hoffenheim	Germany	80
450	RWB	M. Ginter	24	Borussia Mönchengladbach	Germany	80

In [28]:

```
1 #Analyzig the Skills of Players
2
3 @interact
4 def skill(skills = ['Defending', 'General', 'Mental', 'Passing',
5                    'Mobility', 'Power', 'Rating','Shooting'], score = 35):
6     return data[data[skills] > score][['Name', 'Nationality', 'Club', 'Overall', ski
7                                         ascending = False).head(20)
```

skills

score

	Name	Nationality	Club	Overall	Defending
24	G. Chiellini	Italy	Juventus	89	92
8	Sergio Ramos	Spain	Real Madrid	91	90
12	D. Godín	Uruguay	Atlético Madrid	90	89
152	A. Barzagli	Italy	Juventus	84	89
14	N. Kanté	France	Chelsea	89	89
71	T. Alderweireld	Belgium	Tottenham Hotspur	86	89
34	M. Hummels	Germany	FC Bayern München	88	89
69	Azpilicueta	Spain	Chelsea	86	88
44	K. Koulibaly	Senegal	Napoli	87	88
42	S. Umtiti	France	FC Barcelona	87	88
104	Miranda	Brazil	Inter	85	88
27	Casemiro	Brazil	Real Madrid	88	88
51	J. Vertonghen	Belgium	Tottenham Hotspur	87	88
62	R. Varane	France	Real Madrid	86	87
77	M. Škriniar	Slovakia	Inter	85	87
88	K. Manolas	Greece	Roma	85	87
73	M. Benatia	Morocco	Juventus	86	87
59	V. van Dijk	Netherlands	Liverpool	86	87
111	J. Giménez	Uruguay	Atlético Madrid	84	87
54	Piqué	Spain	FC Barcelona	87	87

In [29]:

```
1 #interactive function for getting a report of the players country wise
2
3 @interact
4 def country(country = list(data['Nationality'].value_counts().index)):
5     return data[data['Nationality'] == country][['Name','Position','Overall',
6         'Potential']].sort_values(by = 'Overall',
7         ascending = False).head(15)
```

country

England

	Name	Position	Overall	Potential
16	H. Kane	ST	89	91
60	R. Sterling	RW	86	89
117	D. Alli	LCM	84	90
135	K. Walker	RB	84	84
180	J. Pickford	GK	83	88
182	J. Stones	RCB	83	87
259	H. Maguire	LCB	82	85
290	K. Trippier	RB	82	82
296	J. Henderson	RCM	82	82
254	J. Lingard	CAM	82	83
251	J. Vardy	ST	82	82
319	M. Rashford	LW	81	89
377	C. Smalling	RCB	81	82
396	D. Rose	LB	81	81
409	J. Milner	CM	81	81

In [30]:

```
1 #interactive function to get the list of top 15 players from each of the club
2
3 @interact
4 def club( club = list(data['Club'].value_counts().index[1:])):
5     return data[data['Club'] == club][['Name', 'Jersey Number', 'Position', 'Overall', '
6         'Value', 'Contract Valid Until']].sort_values(by = 'Overall',
7             ascending = False).head(15)
```

club

FC Barcelona

	Name	Jersey Number	Position	Overall	Nationality	Age	Wage	Value	Contract Valid Until
0	L. Messi	10.0	RF	94	Argentina	31	565.0	0.1105	2021
7	L. Suárez	9.0	RS	91	Uruguay	31	455.0	0.0800	2021
18	M. ter Stegen	22.0	GK	89	Germany	26	240.0	0.0580	2022
20	Sergio Busquets	5.0	CDM	89	Spain	29	315.0	0.0515	2023
32	Coutinho	7.0	LW	88	Brazil	26	340.0	0.0695	2023
42	S. Umtiti	23.0	CB	87	France	24	205.0	0.0570	2023
49	Jordi Alba	18.0	LB	87	Spain	29	250.0	0.0380	2020
53	I. Rakitić	4.0	RCM	87	Croatia	30	260.0	0.0465	2021
54	Piqué	3.0	RCB	87	Spain	31	240.0	0.0340	2022
96	A. Vidal	8.0	CAM	85	Chile	31	205.0	0.0260	2021
155	O. Dembélé	11.0	RW	83	France	21	155.0	0.0400	2022
192	Sergi Roberto	20.0	RB	83	Spain	26	170.0	0.0265	2022
239	C. Lenglet	15.0	LCB	82	France	23	135.0	0.0260	2023
263	Rafinha	12.0	CAM	82	Brazil	25	165.0	0.0295	2020
266	J. Cillessen	1.0	GK	82	Netherlands	29	135.0	0.0180	2021

In [31]:

```
1 # finding 15 eldest players from the dataset
2
3 data.sort_values('Age', ascending = False)[['Name', 'Age', 'Club',
4                                             'Nationality', 'Overall']].head(15)
```

Out[31]:

	Name	Age	Club	Nationality	Overall
4741	O. Pérez	45	Pachuca	Mexico	71
18183	K. Pilkington	44	Cambridge United	England	48
17726	T. Warner	44	Accrington Stanley	Trinidad & Tobago	53
10545	S. Narazaki	42	Nagoya Grampus	Japan	65
7225	C. Muñoz	41	CD Universidad de Concepción	Argentina	68
1120	J. Villar	41	No Club	Paraguay	77
12192	H. Sulaimani	41	Ohod Club	Saudi Arabia	63
15426	M. Tyler	41	Peterborough United	England	59
4228	B. Nivet	41	ESTAC Troyes	France	71
10356	F. Kippe	40	Lillestrøm SK	Norway	65
16264	P. van der Vlag	40	FC Emmen	Netherlands	58
9484	B. Castillo	40	Atlético Huila	Colombia	66
4187	C. Lucchetti	40	Atlético Tucumán	Argentina	71
2821	S. Bertoli	40	Patronato	Argentina	73
3550	S. Nakamura	40	Júbilo Iwata	Japan	72

In [32]:

```

1 # finding 15 eldest players from the dataset
2
3 data.sort_values('Age', ascending = True)[['Name', 'Age', 'Club',
4                                             'Nationality', 'Overall']].head(15)

```

Out[32]:

	Name	Age	Club	Nationality	Overall
18206	G. Nugent	16	Tranmere Rovers	England	46
17743	J. Olstad	16	Sarpsborg 08 FF	Norway	52
13293	H. Massengo	16	AS Monaco	France	62
16081	J. Italiano	16	Perth Glory	Australia	58
18166	N. Ayéva	16	Örebro SK	Sweden	48
17712	K. Broda	16	Wisła Kraków	Poland	53
17354	L. D'Arrigo	16	Adelaide United	Australia	54
13567	Y. Verschaeren	16	RSC Anderlecht	Belgium	62
16927	B. Nygren	16	IFK Göteborg	Sweden	56
18018	B. O'Gorman	16	Bray Wanderers	Republic of Ireland	50
17091	A. Doğan	16	Kayserispor	Turkey	55
17115	C. Bassett	16	Colorado Rapids	United States	55
15793	J. Lahne	16	IF Brommapojkarna	Sweden	59
15746	Y. Begraoui	16	AJ Auxerre	France	59
17175	B. Mumba	16	Sunderland	England	55

In [33]:

```
1 # Top 10 Left footed footballers
2
3 data[data['Preferred Foot'] == 'Left'][['Name', 'Age', 'Club',
4     'Nationality', 'Overall']].sort_values(by = 'Overall',
5     ascending = False).head(10)
```

Out[33]:

	Name	Age	Club	Nationality	Overall
0	L. Messi	31	FC Barcelona	Argentina	94
13	David Silva	32	Manchester City	Spain	90
15	P. Dybala	24	Juventus	Argentina	89
17	A. Griezmann	27	Atlético Madrid	France	89
19	T. Courtois	26	Real Madrid	Belgium	89
24	G. Chiellini	33	Juventus	Italy	89
35	Marcelo	30	Real Madrid	Brazil	88
37	H. Lloris	31	Tottenham Hotspur	France	88
36	G. Bale	28	Real Madrid	Wales	88
28	J. Rodríguez	26	FC Bayern München	Colombia	88

In [34]:

```
1 # Top 10 Right footed footballers
2
3 data[data['Preferred Foot'] == 'Right'][['Name', 'Age', 'Club',
4     'Nationality', 'Overall']].sort_values(by = 'Overall',
5     ascending = False).head(10)
```

Out[34]:

	Name	Age	Club	Nationality	Overall
1	Cristiano Ronaldo	33	Juventus	Portugal	94
2	Neymar Jr	26	Paris Saint-Germain	Brazil	92
3	De Gea	27	Manchester United	Spain	91
4	K. De Bruyne	27	Manchester City	Belgium	91
5	E. Hazard	27	Chelsea	Belgium	91
6	L. Modrić	32	Real Madrid	Croatia	91
7	L. Suárez	31	FC Barcelona	Uruguay	91
8	Sergio Ramos	32	Real Madrid	Spain	91
9	J. Oblak	25	Atlético Madrid	Slovenia	90
10	R. Lewandowski	29	FC Bayern München	Poland	90

In []:

1	
---	--