Importing the Libraries

In [1]:
```python
# basic operations
import numpy as np

# for dataframe manipulations
import pandas as pd

# for data visualizations
import matplotlib.pyplot as plt
import seaborn as sns

# for missing values
import missingno as mno

# for date time manipulation
import datetime

# for interactivity
import ipywidgets as widgets
from ipywidgets import interact
from ipywidgets import interact_manual

# setting up the background style for the plots
plt.style.use('fivethirtyeight')
```

Reading the Data

In [3]:
```python
# reading the data and also checking the computation time
%time data = pd.read_csv('data-1.csv')

# lets also check the shape of the dataset
print(data.shape)
```

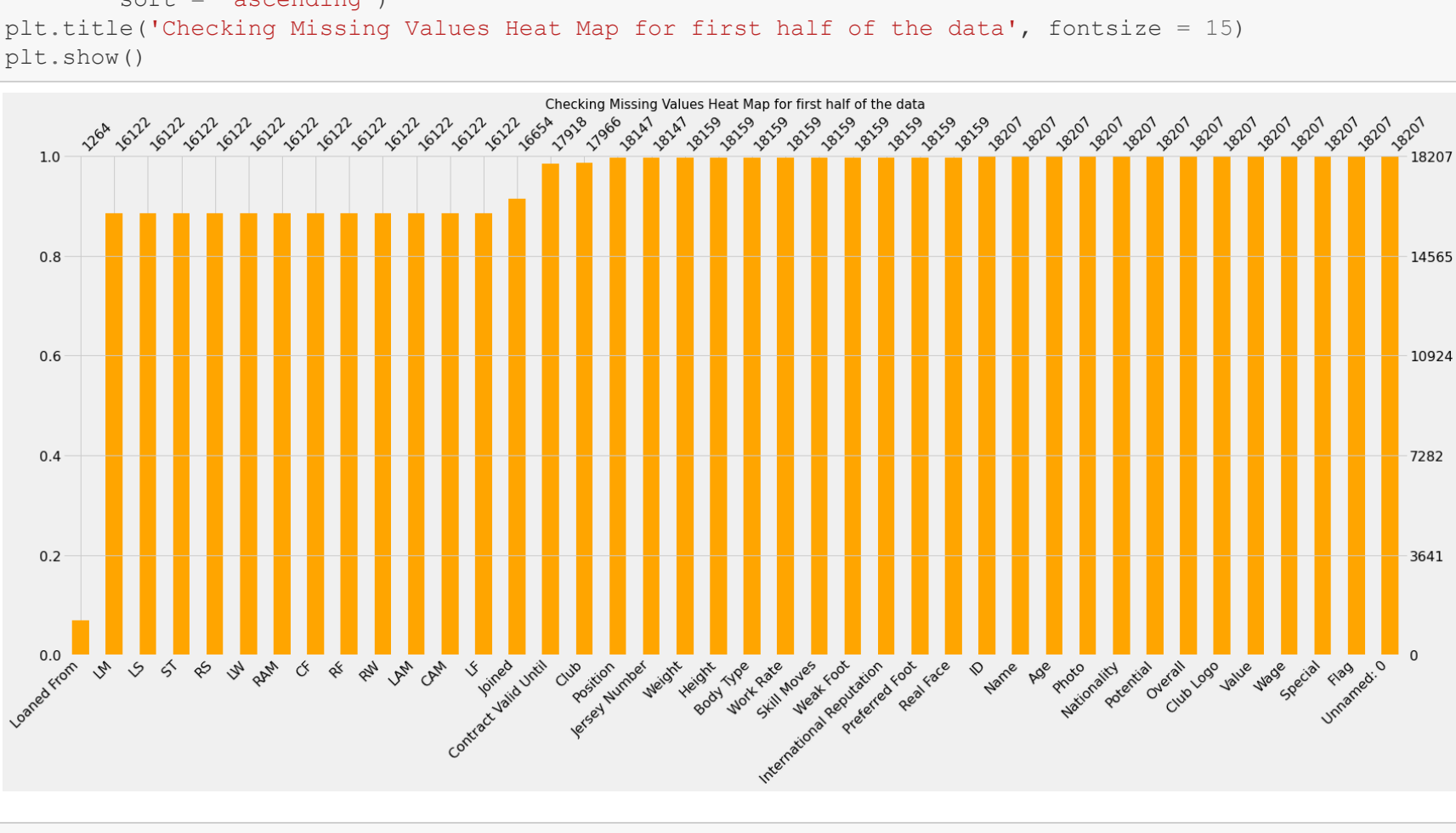
```
Wall time: 269 ms
(18207, 89)
```

In [4]:
```python
# lets check the column names present in the data
data.columns
```

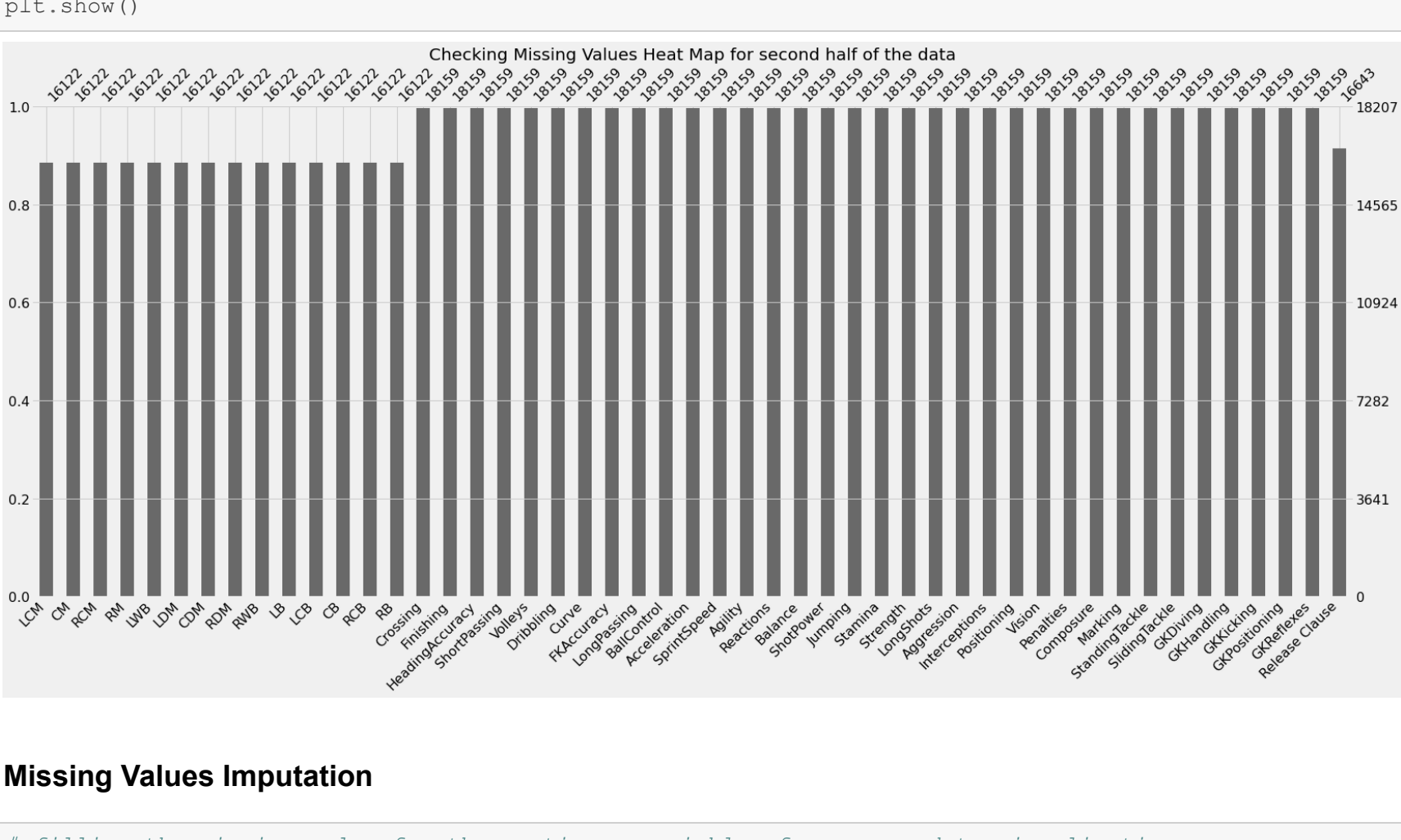
Out[4]:
```
Index(['Unnamed: 0', 'ID', 'Name', 'Age', 'Photo', 'Nationality', 'Flag',
       'Overall', 'Potential', 'Club', 'Club Logo', 'Value', 'Wage', 'Special',
       'Preferred Foot', 'International Reputation', 'Weak Foot',
       'Skill Moves', 'Work Rate', 'Body Type', 'Real Face', 'Position',
       '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', 'Dribbling',
       'Curve', 'FKAccuracy', 'LongPassing', 'BallControl', 'Acceleration',
       'SprintSpeed', 'Agility', 'Reactions', 'Balance', 'ShotPower',
       'Jumping', 'Stamina', 'Strength', 'LongShots', 'Aggression',
       'Interceptions', 'Positioning', 'Vision', 'Penalties', 'Composure',
       'Marking', 'StandingTackle', 'SlidingTackle', 'GKDiving', 'GKHandling',
       'GKKicking', 'GKPositioning', 'GKReflexes', 'Release Clause'],
      dtype='object')
```

Cleaning Data

In [5]:
```python
# checking if the data contains any NULL value
# Visualize missing values as a matrix
mno.bar(data.iloc[:, :40],
        color = 'orange',
        sort = 'ascending')
plt.title('Checking Missing Values Heat Map for first half of the data', fontsize = 15)
plt.show()
```



In [6]:
```python
# Visualize missing values as a matrix
mno.bar(data.iloc[:, 40:])
plt.title('Checking Missing Values Heat Map for second half of the data')
plt.show()
```



Missing Values Imputation

In [7]:
```python
# filling the missing value for the continous variables for proper data visualization

data['ShortPassing'].fillna(data['ShortPassing'].mean(), inplace = True)
data['Volleys'].fillna(data['Volleys'].mean(), inplace = True)
data['Dribbling'].fillna(data['Dribbling'].mean(), inplace = True)
data['Curve'].fillna(data['Curve'].mean(), inplace = True)
data['FKAccuracy'].fillna(data['FKAccuracy'], inplace = True)
data['LongPassing'].fillna(data['LongPassing'].mean(), inplace = True)
data['BallControl'].fillna(data['BallControl'].mean(), inplace = True)
data['HeadingAccuracy'].fillna(data['HeadingAccuracy'].mean(), inplace = True)
data['Finishing'].fillna(data['Finishing'].mean(), inplace = True)
data['Crossing'].fillna(data['Crossing'].mean(), inplace = True)

data['Weight'].fillna('200lbs', inplace = True)
data['Contract Valid Until'].fillna(2019, inplace = True)
data['Height'].fillna("5'11", inplace = True)
data['Loaned From'].fillna('None', inplace = True)
data['Joined'].fillna('Jul 1, 2018', inplace = True)
data['Jersey Number'].fillna(8, inplace = True)
data['Body Type'].fillna('Normal', inplace = True)
data['Position'].fillna('ST', inplace = True)
data['Club'].fillna('No Club', inplace = True)
data['Work Rate'].fillna('Medium/ Medium', inplace = True)
data['Skill Moves'].fillna(data['Skill Moves'].median(), inplace = True)
data['Weak Foot'].fillna(3, inplace = True)
data['Preferred Foot'].fillna('Right', inplace = True)
data['International Reputation'].fillna(1, inplace = True)
data['Wage'].fillna('€200K', inplace = True)
```

In [8]:
```python
pd.set_option('max_rows', 100)
data.isnull().sum()
```

Out[8]:
```
Unnamed: 0                     0
ID                             0
Name                           0
Age                            0
Photo                          0
Nationality                    0
Flag                           0
Overall                        0
Potential                      0
Club                           0
Club Logo                      0
Value                          0
Wage                           0
Special                        0
Preferred Foot                 0
International Reputation       0
Weak Foot                      0
Skill Moves                    0
Work Rate                      0
Body Type                      0
Real Face                     48
Position                       0
Jersey Number                  0
Joined                         0
Loaned From                    0
Contract Valid Until           0
Height                         0
Weight                         0
LS                          2085
ST                          2085
RS                          2085
LW                          2085
LF                          2085
CF                          2085
RF                          2085
RW                          2085
LAM                         2085
CAM                         2085
RAM                         2085
LM                          2085
LCM                         2085
CM                          2085
RCM                         2085
RM                          2085
LWB                         2085
LDM                         2085
CDM                         2085
RDM                         2085
RWB                         2085
LB                          2085
LCB                         2085
CB                          2085
RCB                         2085
RB                          2085
Crossing                       0
Finishing                      0
HeadingAccuracy                0
ShortPassing                   0
Volleys                        0
Dribbling                      0
Curve                          0
FKAccuracy                    48
LongPassing                    0
BallControl                    0
Acceleration                  48
SprintSpeed                   48
Agility                       48
Reactions                     48
Balance                       48
ShotPower                     48
Jumping                       48
Stamina                       48
Strength                      48
LongShots                     48
Aggression                    48
Interceptions                 48
Positioning                   48
Vision                        48
Penalties                     48
Composure                     48
Marking                       48
StandingTackle                48
SlidingTackle                 48
GKDiving                      48
GKHandling                    48
GKKicking                     48
GKPositioning                 48
GKReflexes                    48
Release Clause              1564
dtype: int64
```

In [9]:
```python
# impute with 0 for rest of the columns
data.fillna(0, inplace = True)

# lets check whether the data still has any missing values
data.isnull().sum().sum()
```

Out[9]: 0

Feature Engineering

In [10]:
```python
# creating new features by aggregating the features

def defending(data):
    return int(round((data[['Marking', 'StandingTackle',
                            'SlidingTackle']].mean()).mean()))

def general(data):
    return int(round((data[['HeadingAccuracy', 'Dribbling', 'Curve',
                            'BallControl']].mean()).mean()))

def mental(data):
    return int(round((data[['Aggression', 'Interceptions', 'Positioning',
                            'Vision','Composure']].mean()).mean()))

def passing(data):
    return int(round((data[['Crossing', 'ShortPassing',
                            'LongPassing']].mean()).mean()))

def mobility(data):
    return int(round((data[['Acceleration', 'SprintSpeed',
                            'Agility','Reactions']].mean()).mean()))
def power(data):
    return int(round((data[['Balance', 'Jumping', 'Stamina',
                            'Strength']].mean()).mean()))

def rating(data):
    return int(round((data[['Potential', 'Overall']].mean()).mean()))

def shooting(data):
    return int(round((data[['Finishing', 'Volleys', 'FKAccuracy',
                            'ShotPower','LongShots', 'Penalties']].mean()).mean()))
```

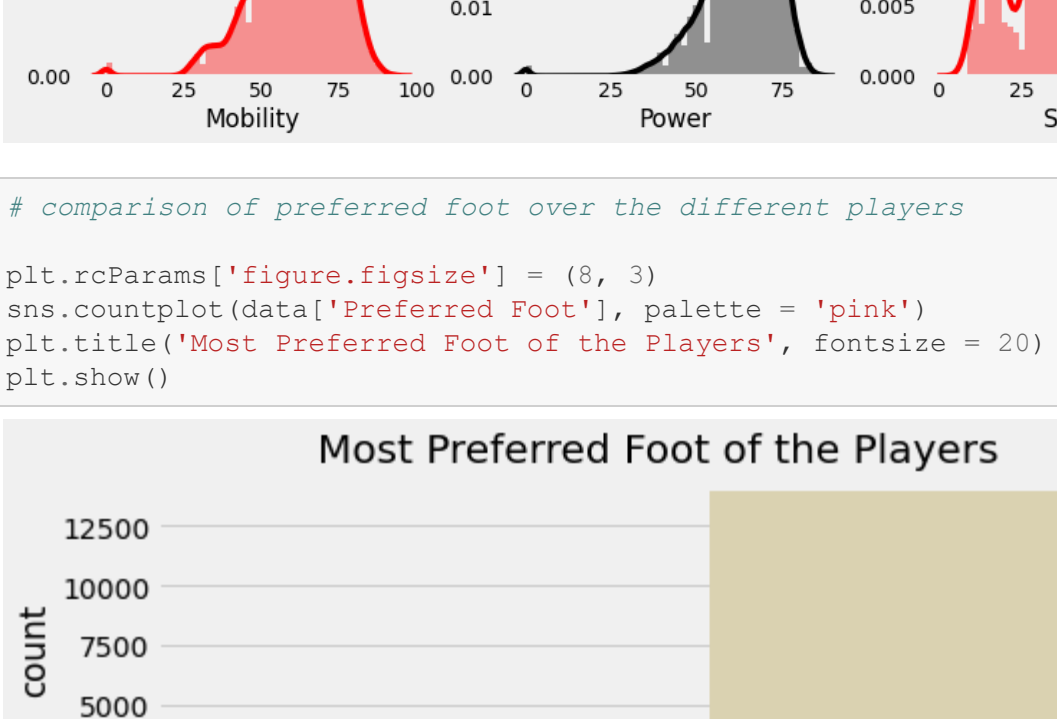
In [11]:
```python
# adding these categories to the data

data['Defending'] = data.apply(defending, axis = 1)
data['General'] = data.apply(general, axis = 1)
data['Mental'] = data.apply(mental, axis = 1)
data['Passing'] = data.apply(passing, axis = 1)
data['Mobility'] = data.apply(mobility, axis = 1)
data['Power'] = data.apply(power, axis = 1)
data['Rating'] = data.apply(rating, axis = 1)
data['Shooting'] = data.apply(shooting, axis = 1)

# lets check the column names in the data after adding new features
data.columns
```

Out[11]:
```
Index(['Unnamed: 0', 'ID', 'Name', 'Age', 'Photo', 'Nationality', 'Flag',
       'Overall', 'Potential', 'Club', 'Club Logo', 'Value', 'Wage', 'Special',
       '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', 'Dribbling',
       'Curve', 'FKAccuracy', 'LongPassing', 'BallControl', 'Acceleration',
       'SprintSpeed', 'Agility', 'Reactions', 'Balance', 'ShotPower',
       'Jumping', 'Stamina', 'Strength', 'LongShots', 'Aggression',
       'Interceptions', 'Positioning', 'Vision', 'Penalties', 'Composure',
       'Marking', 'StandingTackle', 'SlidingTackle', 'GKDiving', 'GKHandling',
       'GKKicking', 'GKPositioning', 'GKReflexes', 'Release Clause',
       'Defending', 'General', 'Mental', 'Passing', 'Mobility', 'Power',
       'Rating', 'Shooting'],
      dtype='object')
```

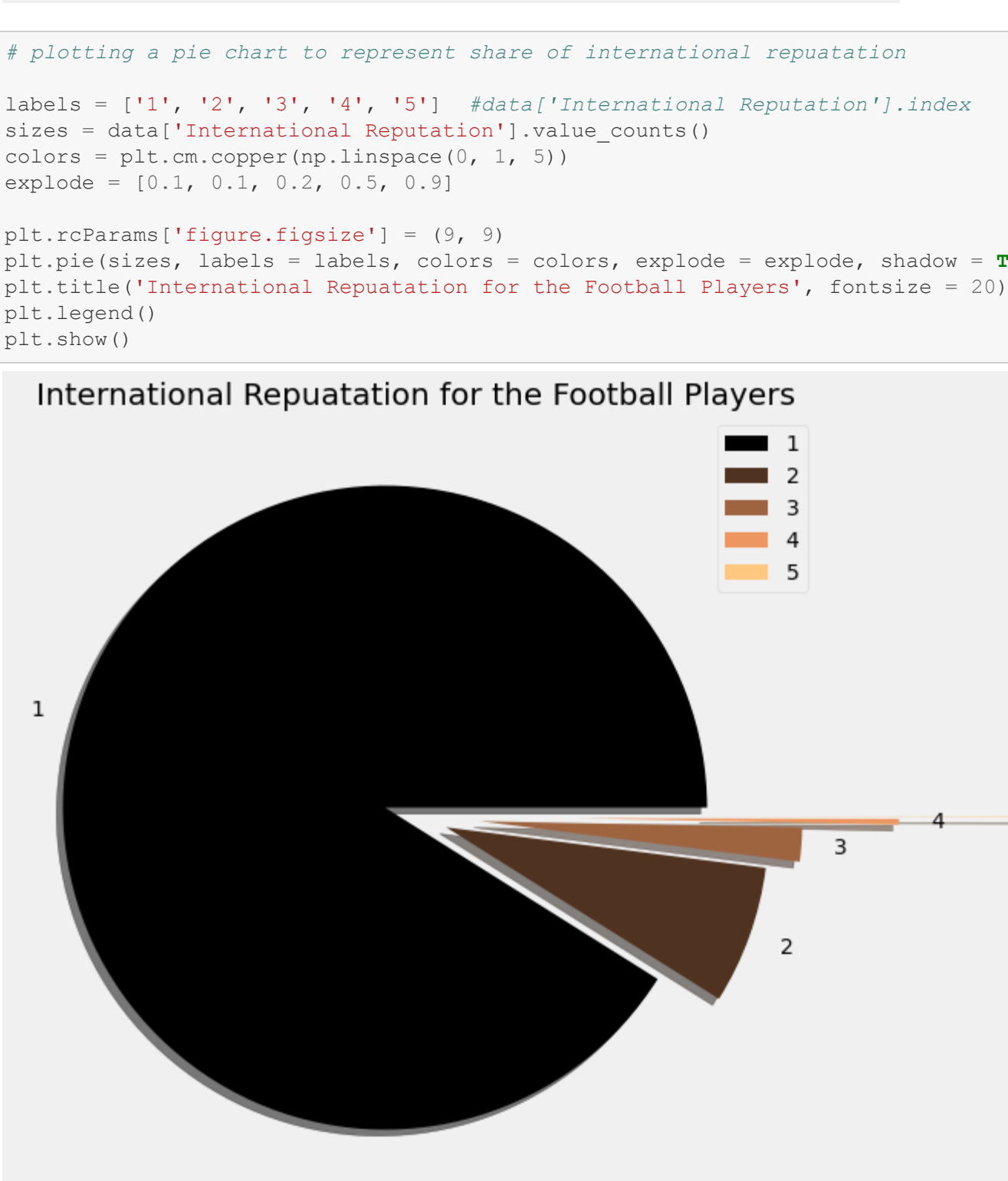
Data Visualization

In [12]:
```python
# lets check the Distribution of Scores of Different Skills

plt.rcParams['figure.figsize'] = (18, 12)
plt.subplot(2, 4, 1)
sns.distplot(data['Defending'], color = 'red')
plt.grid()

plt.subplot(2, 4, 2)
sns.distplot(data['General'], color = 'black')
plt.grid()

plt.subplot(2, 4, 3)
sns.distplot(data['Mental'], color = 'red')
plt.grid()

plt.subplot(2, 4, 4)
sns.distplot(data['Passing'], color = 'black')
plt.grid()

plt.subplot(2, 4, 5)
sns.distplot(data['Mobility'], color = 'red')
plt.grid()

plt.subplot(2, 4, 6)
sns.distplot(data['Power'], color = 'black')
plt.grid()

plt.subplot(2, 4, 7)
sns.distplot(data['Shooting'], color = 'red')
plt.grid()

plt.subplot(2, 4, 8)
sns.distplot(data['Rating'], color = 'black')
plt.grid()

plt.suptitle('Score Distributions for Different Abilities')
plt.show()
```

In [13]:
```python
# comparison of preferred foot over the different players

plt.rcParams['figure.figsize'] = (8, 3)
sns.countplot(data['Preferred Foot'], palette = 'pink')
plt.title('Most Preferred Foot of the Players', fontsize = 20)
plt.show()
```



In [14]:
```python
# plotting a pie chart to represent share of international repuatation

labels = ['1', '2', '3', '4', '5']  #data['International Reputation'].index
sizes = data['International Reputation'].value_counts()
colors = plt.cm.copper(np.linspace(0, 1, 5))
explode = [0.1, 0.1, 0.2, 0.5, 0.9]

plt.rcParams['figure.figsize'] = (9, 9)
plt.pie(sizes, labels = labels, colors = colors, explode = explode, shadow = True,)
plt.title('International Repuatation for the Football Players', fontsize = 20)
plt.legend()
plt.show()
```



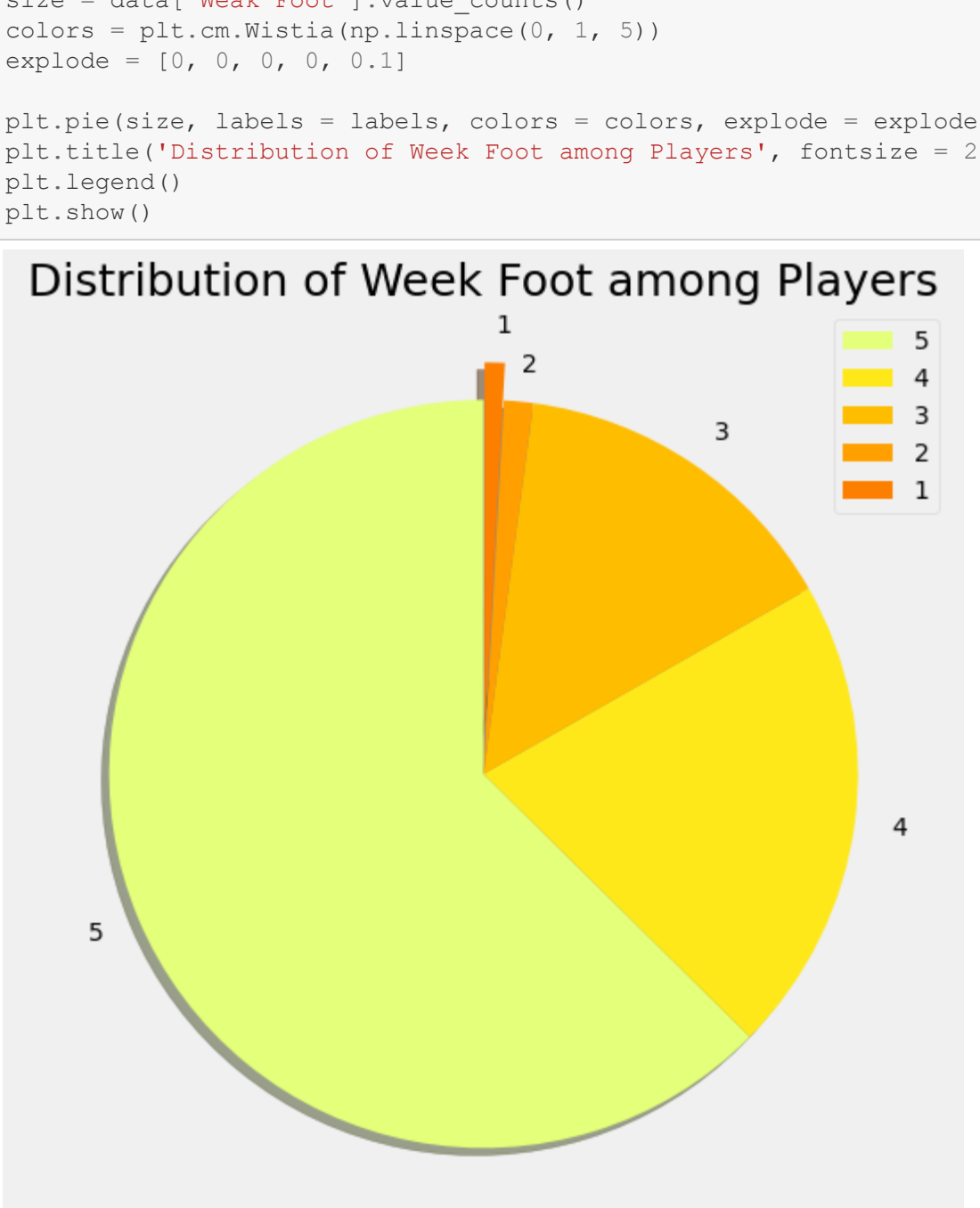
Let's check the Players with International Reputation as 5

In [15]:
```python
data[data['International Reputation'] == 5][['Name','Nationality',
                                              'Overall']].sort_values(by = 'Overall',
                                                         ascending = False).style.background_gradient(cmap = 'magma')
```

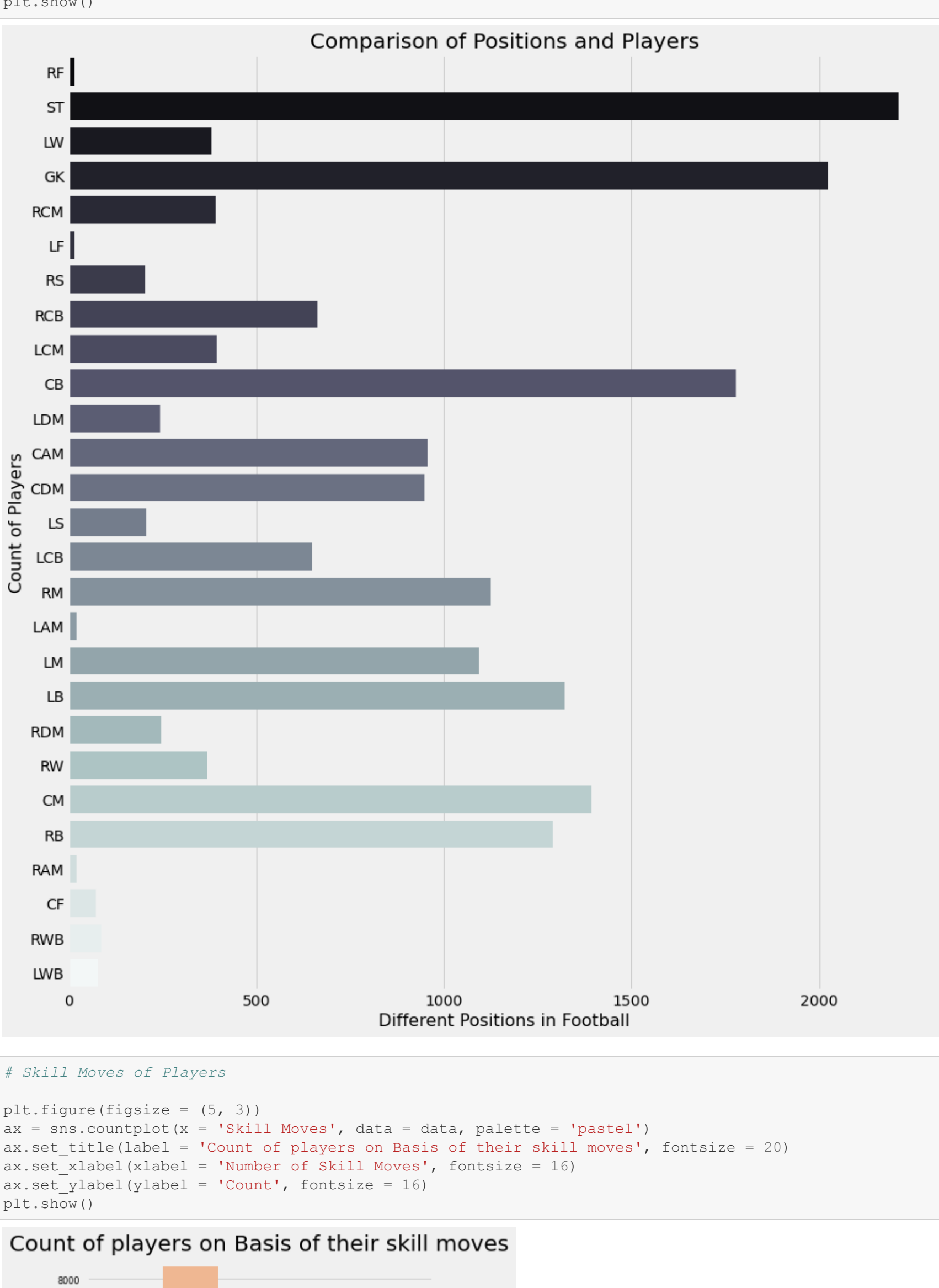
Out[15]:

	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

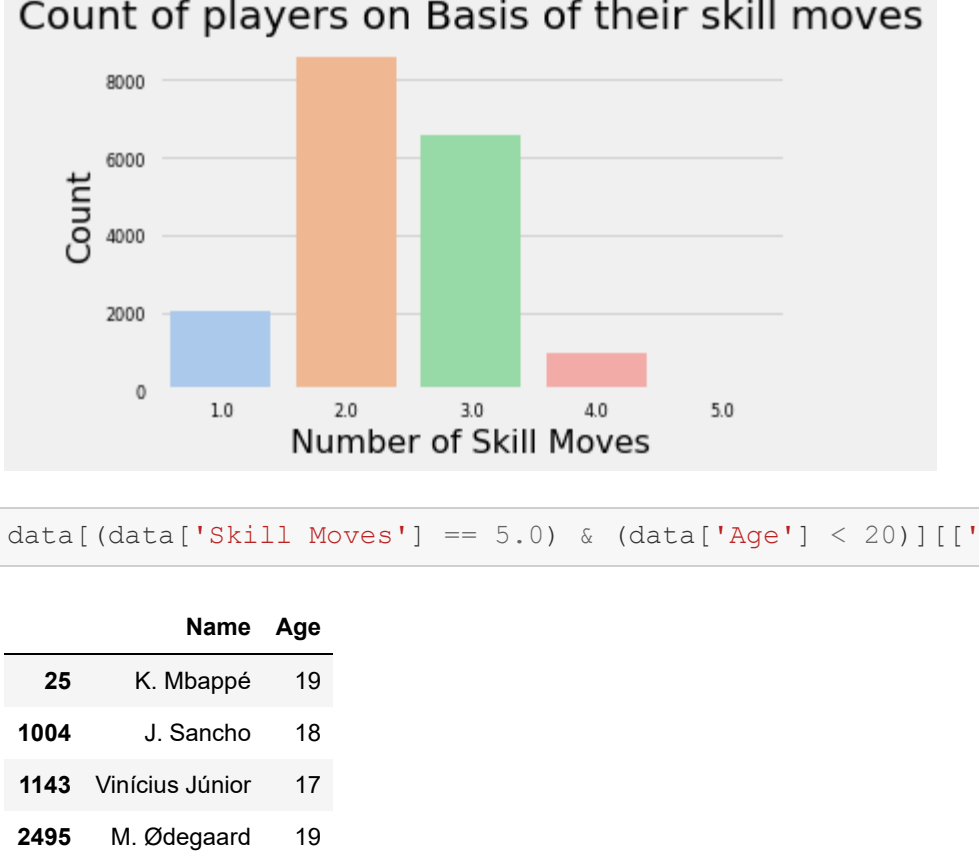
In [16]:
```python
# plotting a pie chart to represent the share of week foot players

labels = ['5', '4', '3', '2', '1']
size = data['Weak Foot'].value_counts()
colors = plt.cm.Wistia(np.linspace(0, 1, 5))
explode = [0, 0, 0, 0, 0.1]

plt.pie(size, labels = labels, colors = colors, explode = explode, shadow = True, startangle = 90)
plt.title('Distribution of Week Foot among Players', fontsize = 25)
plt.legend()
plt.show()
```



In [17]:
```python
# different positions acquired by the players

plt.figure(figsize = (13, 15))
plt.style.use('fivethirtyeight')
ax = sns.countplot(y = 'Position', data = data, palette = 'bone')
ax.set_xlabel(xlabel = 'Different Positions in Football', fontsize = 16)
ax.set_ylabel(ylabel = 'Count of Players', fontsize = 16)
ax.set_title(label = 'Comparison of Positions and Players', fontsize = 20)
plt.show()
```



In [36]:
```python
# Skill Moves of Players

plt.figure(figsize = (5, 3))
ax = sns.countplot(x = 'Skill Moves', data = data, palette = 'pastel')
ax.set_title(label = 'Count of players on Basis of their skill moves', fontsize = 20)
ax.set_xlabel(xlabel = 'Number of Skill Moves', fontsize = 16)
ax.set_ylabel(ylabel = 'Count', fontsize = 16)
plt.show()
```

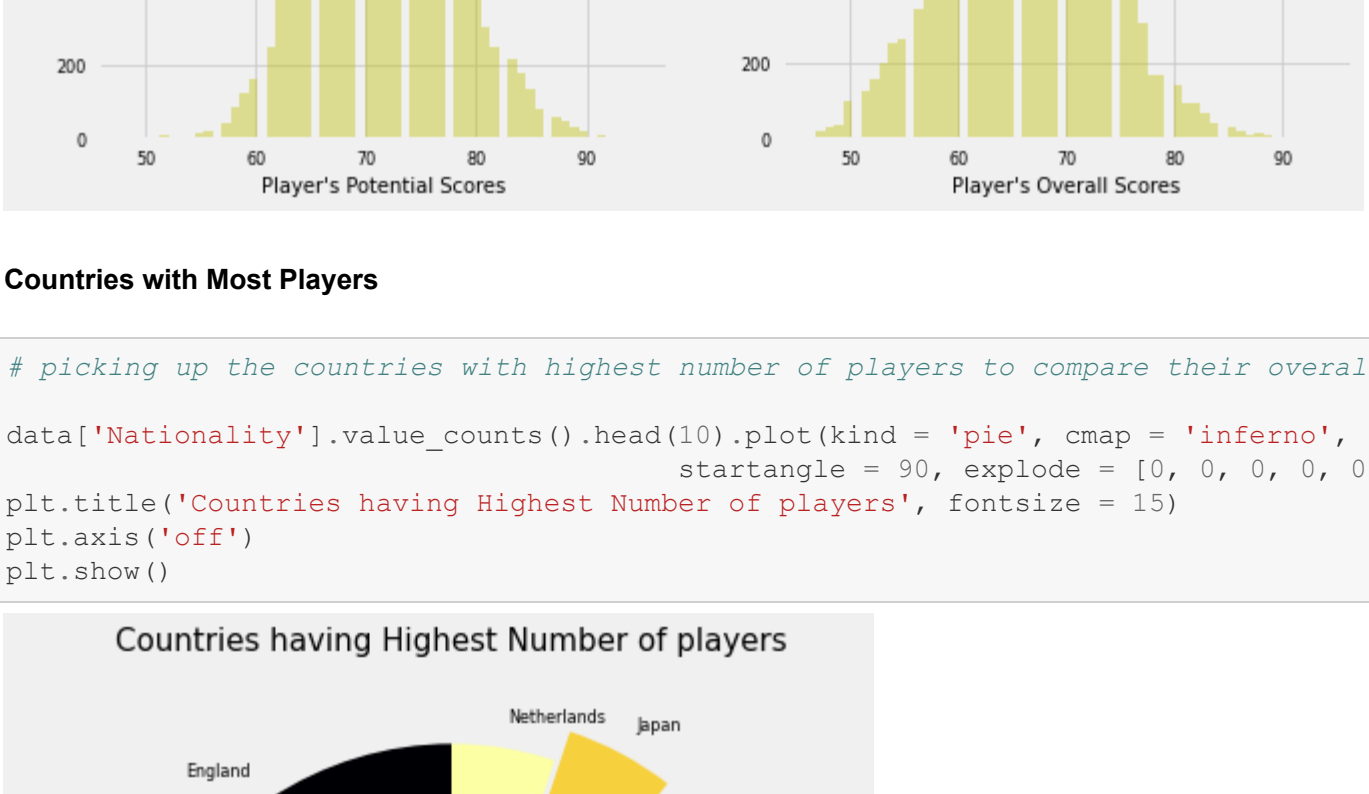


In [37]:
```python
data[(data['Skill Moves'] == 5.0) & (data['Age'] < 20)][['Name','Age']]
```

Out[37]:

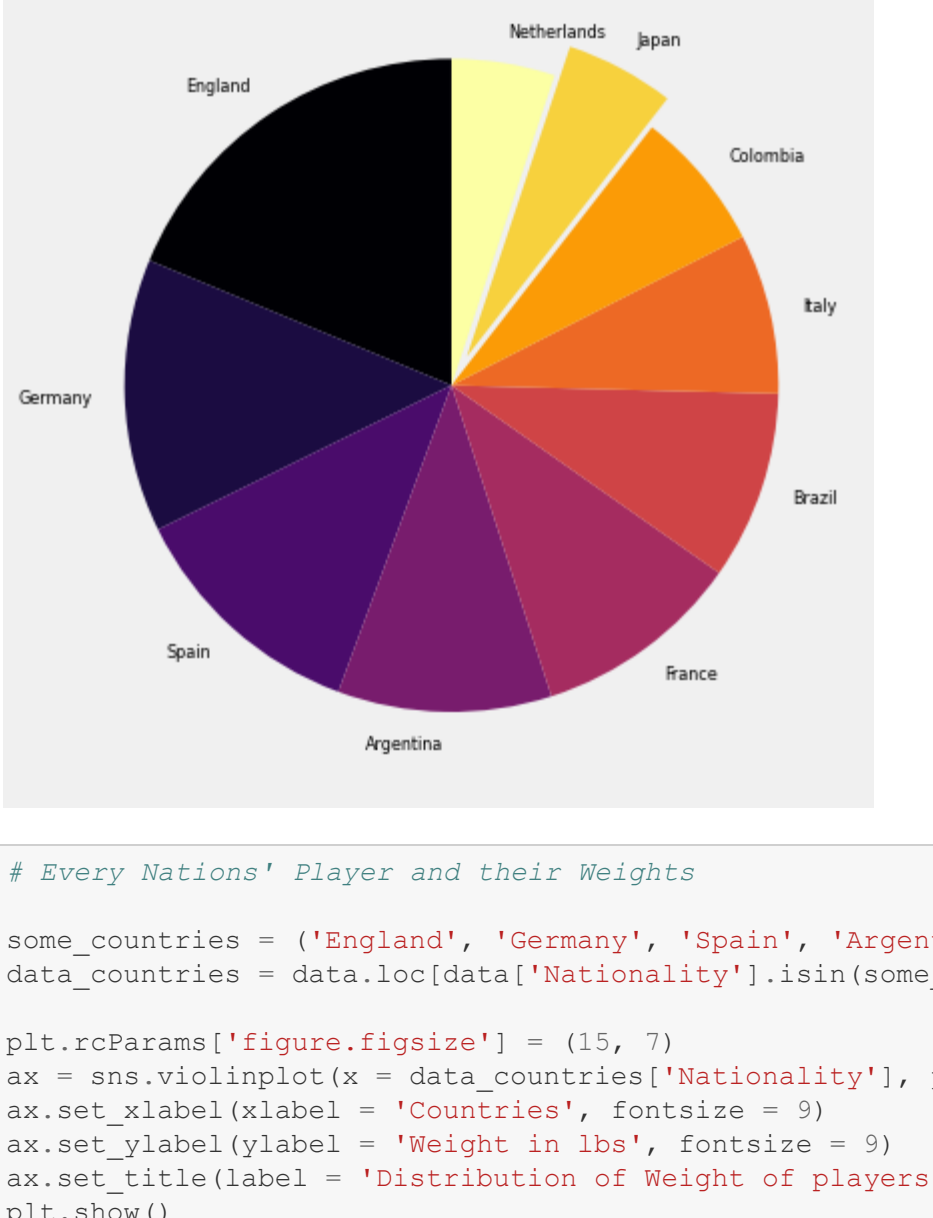
	Name	Age
25	K. Mbappé	19
1004	J. Sancho	18
1143	Vinícius Júnior	17
2495	M. Ødegaard	19

In [39]:
```python
# To show Different potential scores of the players participating in the FIFA 2019

plt.figure(figsize=(10, 4))
plt.style.use('seaborn-paper')

plt.subplot(1, 2, 1)
x = data.Potential
ax = sns.distplot(x, bins = 58, kde = False, color = 'y')
ax.set_xlabel(xlabel = "Player's Potential Scores", fontsize = 10)
ax.set_ylabel(ylabel = 'Number of players', fontsize = 10)
ax.set_title(label = 'Histogram of players Potential Scores', fontsize = 15)

plt.subplot(1, 2, 2)
y = data.Overall
ax = sns.distplot(y, bins = 58, kde = False, color = 'y')
ax.set_xlabel(xlabel = "Player's Overall Scores", fontsize = 10)
ax.set_ylabel(ylabel = 'Number of players', fontsize = 10)
ax.set_title(label = 'Histogram of players Overall Scores', fontsize = 15)
plt.show()
```

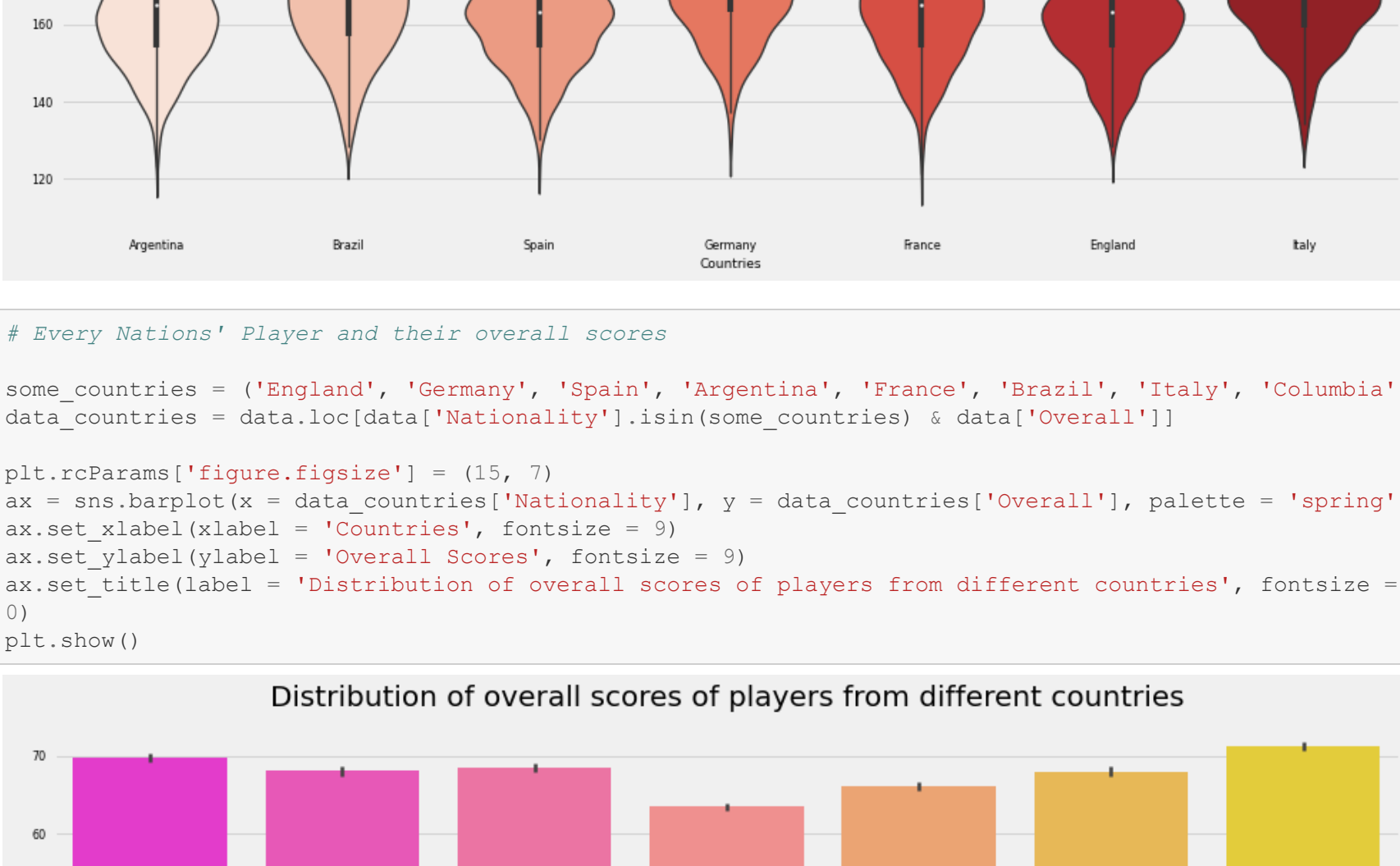


Countries with Most Players

In [41]:
```python
# picking up the countries with highest number of players to compare their overall scores

data['Nationality'].value_counts().head(10).plot(kind = 'pie', cmap = 'inferno',
                                         startangle = 90, explode = [0, 0, 0, 0, 0, 0, 0, 0, 0.1, 0])
plt.title('Countries having Highest Number of players', fontsize = 15)
plt.axis('off')
plt.show()
```

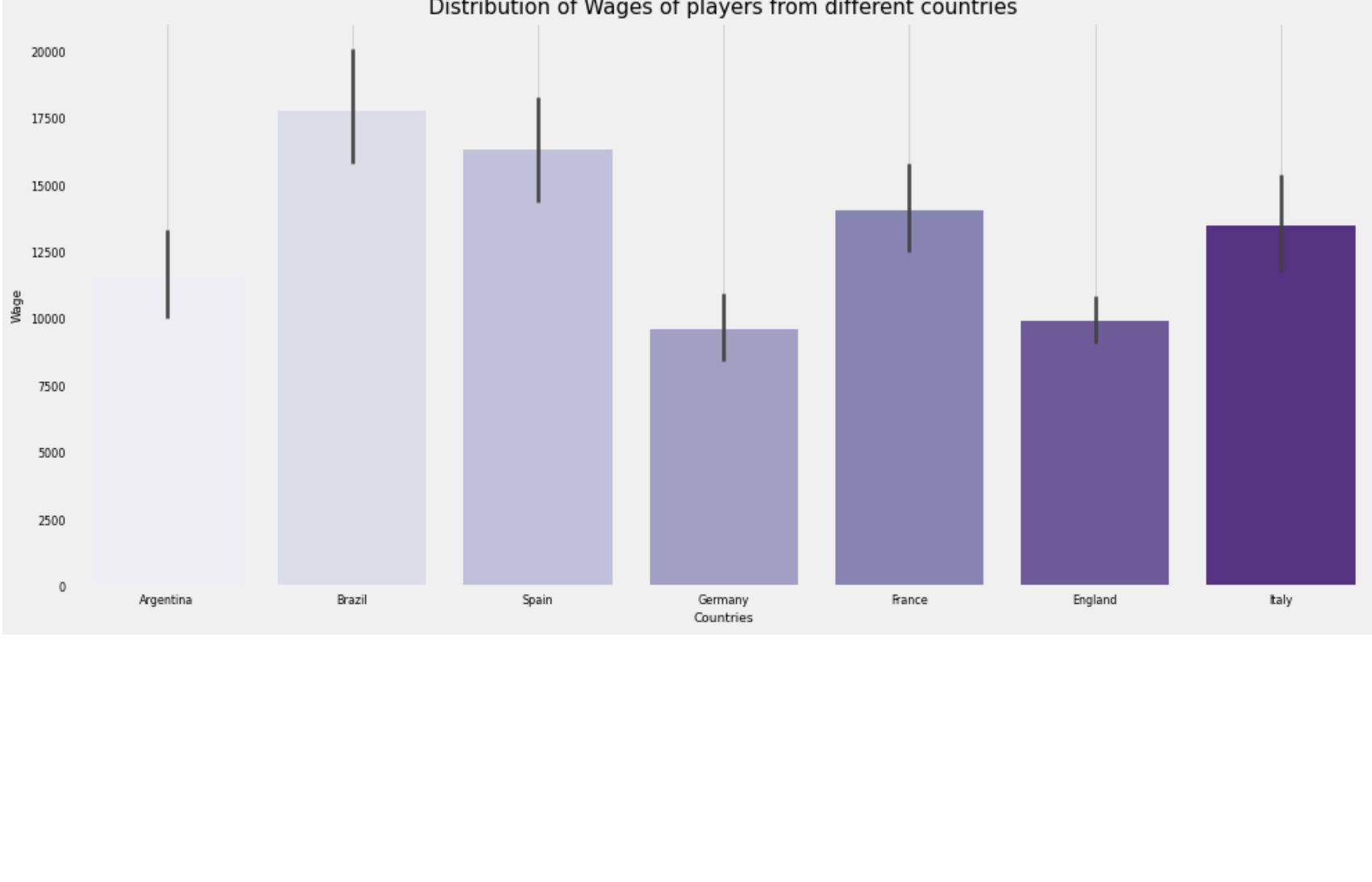


In [42]:
```python
# Every Nations' Player and their Weights

some_countries = ('England', 'Germany', 'Spain', 'Argentina', 'France', 'Brazil', 'Italy', 'Columbia')
data_countries = data.loc[data['Nationality'].isin(some_countries) & data['Weight']]

plt.rcParams['figure.figsize'] = (15, 7)
ax = sns.violinplot(x = data_countries['Nationality'], y = data_countries['Weight'], palette = 'Reds')
ax.set_xlabel(xlabel = 'Countries', fontsize = 9)
ax.set_ylabel(ylabel = 'Weight in lbs', fontsize = 9)
ax.set_title(label = 'Distribution of Weight of players from different countries', fontsize = 20)
plt.show()
```



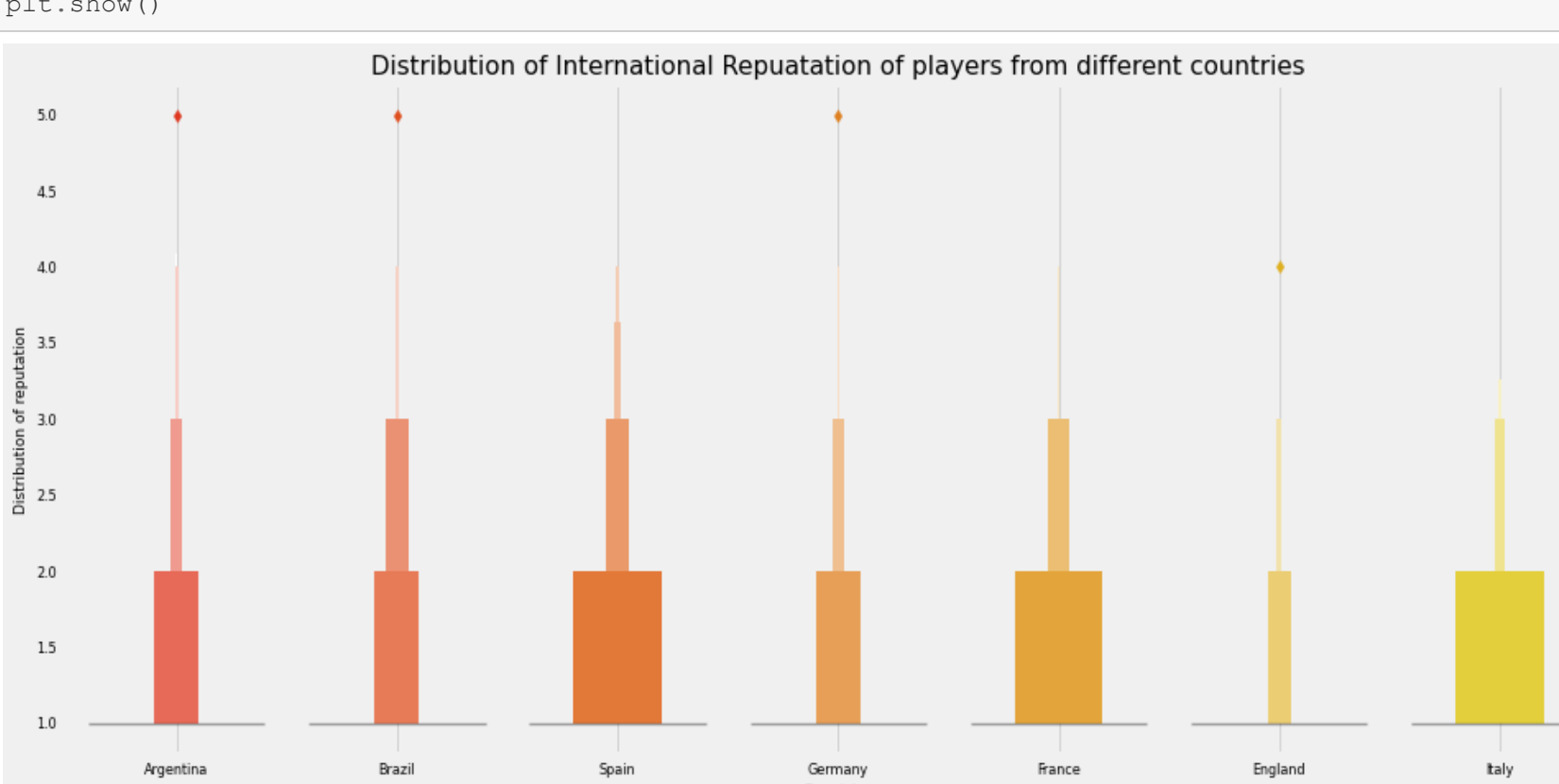
In [43]:
```python
# Every Nations' Player and their overall scores

some_countries = ('England', 'Germany', 'Spain', 'Argentina', 'France', 'Brazil', 'Italy', 'Columbia')
data_countries = data.loc[data['Nationality'].isin(some_countries) & data['Overall']]

plt.rcParams['figure.figsize'] = (15, 7)
ax = sns.barplot(x = data_countries['Nationality'], y = data_countries['Overall'], palette = 'spring')
ax.set_xlabel(xlabel = 'Countries', fontsize = 9)
ax.set_ylabel(ylabel = 'Overall Scores', fontsize = 9)
ax.set_title(label = 'Distribution of overall scores of players from different countries', fontsize = 2
0)
plt.show()
```

In [44]:
```python
# Every Nations' Player and their wages

some_countries = ('England', 'Germany', 'Spain', 'Argentina', 'France', 'Brazil', 'Italy', 'Columbia')
data_countries = data.loc[data['Nationality'].isin(some_countries) & data['Wage']]

plt.rcParams['figure.figsize'] = (15, 7)
ax = sns.barplot(x = data_countries['Nationality'], y = data_countries['Wage'], palette = 'Purples')
ax.set_xlabel(xlabel = 'Countries', fontsize = 9)
ax.set_ylabel(ylabel = 'Wage', fontsize = 9)
ax.set_title(label = 'Distribution of Wages of players from different countries', fontsize = 15)
plt.grid()
plt.show()
```



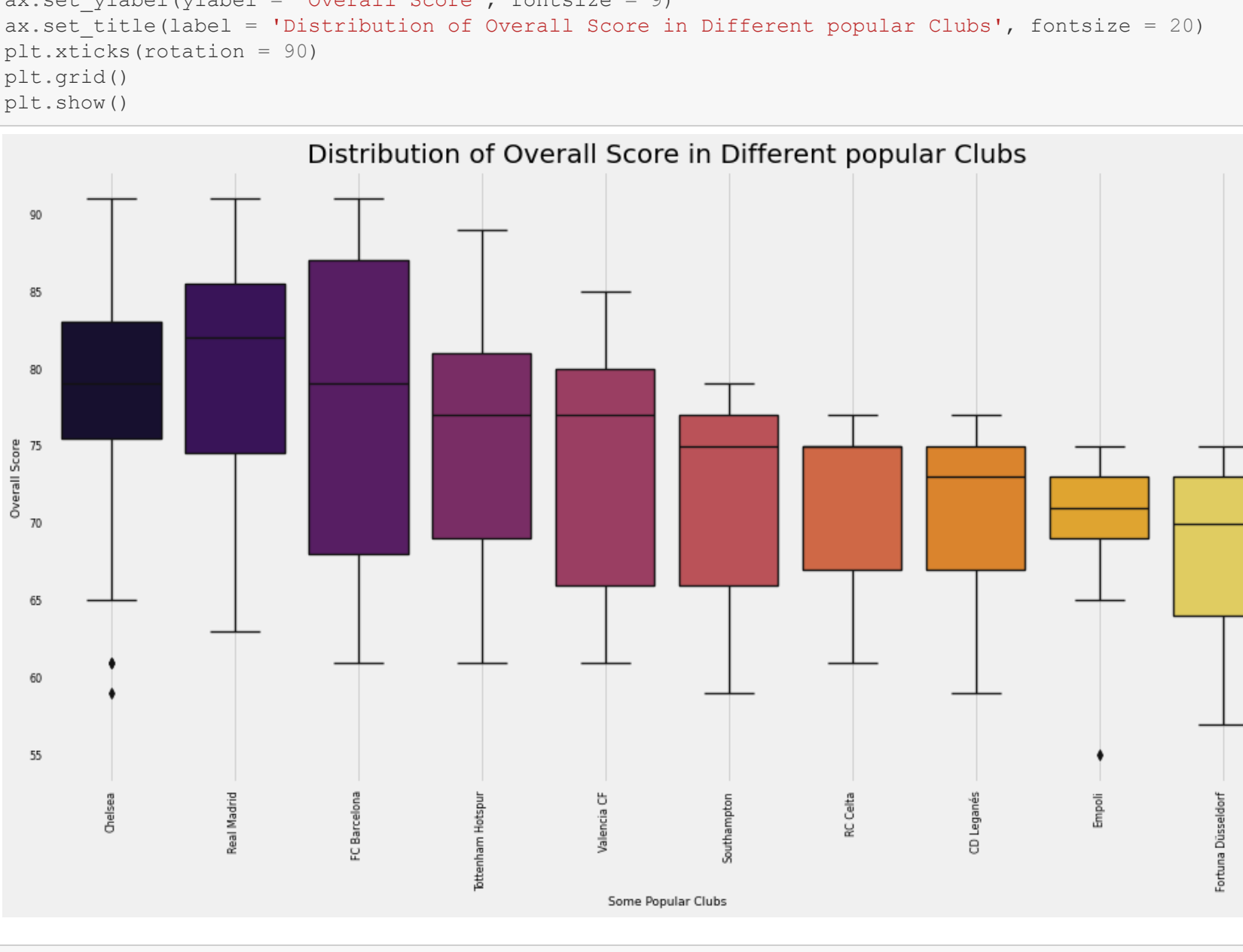
In [45]:

```python
# Every Nations' Player and their International Reputation

some_countries = ('England', 'Germany', 'Spain', 'Argentina', 'France', 'Brazil', 'Italy', 'Columbia')
data_countries = data.loc[data['Nationality'].isin(some_countries) & data['International Reputation']]

plt.rcParams['figure.figsize'] = (15, 7)
ax = sns.boxenplot(x = data_countries['Nationality'], y = data_countries['International Reputation'], p
alette = 'autumn')
ax.set_xlabel(xlabel = 'Countries', fontsize = 9)
ax.set_ylabel(ylabel = 'Distribution of reputation', fontsize = 9)
ax.set_title(label = 'Distribution of International Repuatation of players from different countries', f
ontsize = 15)
plt.grid()
plt.show()
```



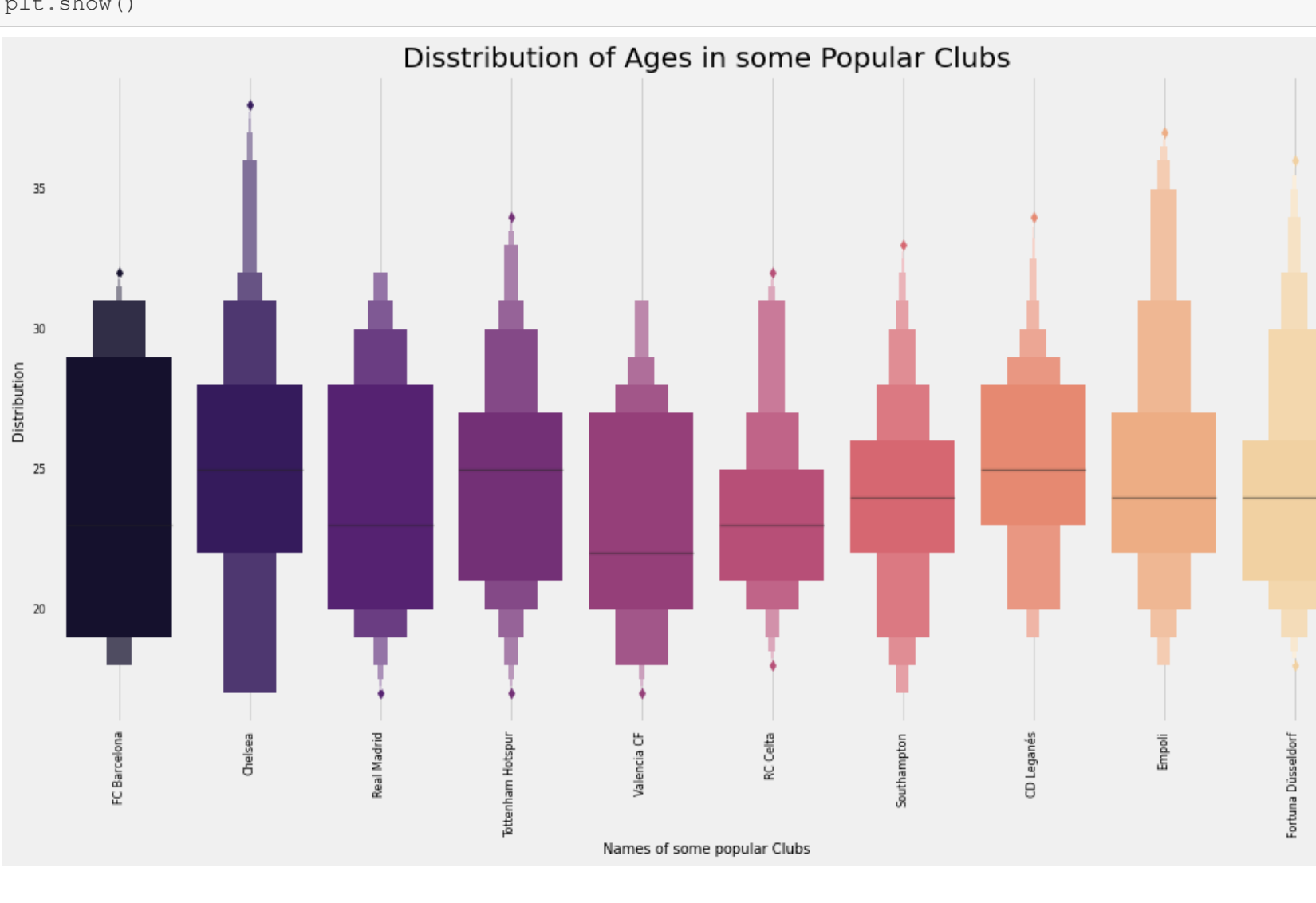
In [46]:

```python
some_clubs = ('CD Leganés', 'Southampton', 'RC Celta', 'Empoli', 'Fortuna Düsseldorf', 'Manchestar Cit
y',
             'Tottenham Hotspur', 'FC Barcelona', 'Valencia CF', 'Chelsea', 'Real Madrid')

data_clubs = data.loc[data['Club'].isin(some_clubs) & data['Overall']]

plt.rcParams['figure.figsize'] = (15, 8)
ax = sns.boxplot(x = data_clubs['Club'], y = data_clubs['Overall'], palette = 'inferno')
ax.set_xlabel(xlabel = 'Some Popular Clubs', fontsize = 9)
ax.set_ylabel(ylabel = 'Overall Score', fontsize = 9)
ax.set_title(label = 'Distribution of Overall Score in Different popular Clubs', fontsize = 20)
plt.xticks(rotation = 90)
plt.grid()
plt.show()
```



In [47]:

```python
# Distribution of Ages in some Popular clubs

some_clubs = ('CD Leganés', 'Southampton', 'RC Celta', 'Empoli', 'Fortuna Düsseldorf', 'Manchestar Cit
y',
             'Tottenham Hotspur', 'FC Barcelona', 'Valencia CF', 'Chelsea', 'Real Madrid')

data_club = data.loc[data['Club'].isin(some_clubs) & data['Wage']]

plt.rcParams['figure.figsize'] = (15, 8)
ax = sns.boxenplot(x = 'Club', y = 'Age', data = data_club, palette = 'magma')
ax.set_xlabel(xlabel = 'Names of some popular Clubs', fontsize = 10)
ax.set_ylabel(ylabel = 'Distribution', fontsize = 10)
ax.set_title(label = 'Disstribution of Ages in some Popular Clubs', fontsize = 20)
plt.xticks(rotation = 90)
plt.grid()
plt.show()
```



Query Analysis

Best Players per each position with their age, club, and nationality based on their Overall Scores

In [50]:

```python
# best players per each position with their age, club, and nationality based on their overall scores

data.iloc[data.groupby(data['Position'])['Overall'].idxmax()][['Position', 'Name', 'Age', 'Club',
                     'Nationality','Overall']].sort_values(by = 'Overall',
                                        ascending = False).style.background_gradient(cmap = 'pink')
```

Out[50]:

	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

Best Players for each Skill

In [51]:

```python
## Let's Analyze the Skills of Players

@interact
def skill(skills = ['Defending', 'General', 'Mental', 'Passing',
                    'Mobility', 'Power', 'Rating','Shooting'], score = 75):
    return data[data[skills] > score][['Name', 'Nationality', 'Club', 'Overall', skills]].sort_values(b
y = skills,
                                ascending = False).head(20).style.background_gradient(cmap = 'Blue
s')
```

Let's make a function to get the list of Top 15 Players from each Country

In [52]:

```python
# lets make an interactive function for getting a report of the players country wise

# lets make a function to see the list of top 15 players from each country
@interact
def country(country = list(data['Nationality'].value_counts().index)):
    return data[data['Nationality'] == country][['Name','Position','Overall',
                 'Potential']].sort_values(by = 'Overall',
                           ascending = False).head(15).style.background_gradient(cmap = 'magma')
```

Let's make a function to get the list of Top 15 Players from each Club

In [53]:

```python
# lets make an interactive function to get the list of top 15 players from each of the club

# lets define a function
@interact
def club(club = list(data['Club'].value_counts().index[1:])):
    return data[data['Club'] == club][['Name','Jersey Number','Position','Overall','Nationality','Age',
'Wage',
                   'Value','Contract Valid Until']].sort_values(by = 'Overall',
                              ascending  = False).head(15).style.background_gradient(cmap = 'inferno'
)
```

youngest Players from the FIFA 2019

In [54]:

```python
# finding 5 youngest Players from the dataset

youngest = data[data['Age'] == 16][['Name', 'Age', 'Club', 'Nationality', 'Overall']]
youngest.sort_values(by = 'Overall', ascending = False).head().style.background_gradient(cmap = 'magma'
)
```

Out[54]:

	Name	Age	Club	Nationality	Overall
11457	W. Geubbels	16	AS Monaco	France	64
11732	A. Taoui	16	Toulouse Football Club	France	64
12496	Pelayo Morilla	16	Real Sporting de Gijón	Spain	63
12828	Guerrero	16	CF Rayo Majadahonda	Spain	63
13293	H. Massengo	16	AS Monaco	France	62

15 Eldest Players from FIFA 2019

In [55]:

```python
# finding 15 eldest players from the dataset

data.sort_values('Age', ascending = False)[['Name', 'Age', 'Club',
                         'Nationality', 'Overall']].head(15).style.background_gradient(cmap = 'Wis
tia')
```

Out[55]:

	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

The longest membership in the club

In [56]:

```python
# The longest membership in the club

now = datetime.datetime.now()
data['Join_year'] = data.Joined.dropna().map(lambda x: x.split(',')[1].split(' ')[1])
data['Years_of_member'] = (data.Join_year.dropna().map(lambda x: now.year - int(x))).astype('int')
membership = data[['Name', 'Club', 'Years_of_member']].sort_values(by = 'Years_of_member', ascending =
False).head(10)
membership.set_index('Name', inplace=True)
membership.style.background_gradient(cmap = 'Reds')
```

Out[56]:

Name	Club	Years_of_member
O. Pérez	Pachuca	31
M. Al Shalhoub	Al Hilal	24
H. Sogahata	Kashima Antlers	24
M. Ogasawara	Kashima Antlers	24
S. Narazaki	Nagoya Grampus	23
M. Wölfli	BSC Young Boys	22
K. Kitamoto	Vissel Kobe	22
C. Källqvist	BK Häcken	21
Y. Endo	Gamba Osaka	21
S. Pellissier	Chievo Verona	20

In [57]:

```python
import ipywidgets as widgets
from ipywidgets import interact
@interact
def check(column = 'Years_of_member',
          club = ['FC Barcelona', 'Real Madrid', 'Chelsea'], x = 4):
    return data[(data[column] > x) & (data['Club'] == club)][['Name', 'Club',
                                  'Years_of_member']].sort_values(by = 'Years_of_member',
                                              ascending = False).style.background_gra
dient(cmap = 'magma')
```

Defining the features of players

In [58]:

```python
# defining the features of players

player_features = ('Acceleration', 'Aggression', 'Agility',
                   'Balance', 'BallControl', 'Composure',
                   'Crossing', 'Dribbling', 'FKAccuracy',
                   'Finishing', 'GKDiving', 'GKHandling',
                   'GKKicking', 'GKPositioning', 'GKReflexes',
                   'HeadingAccuracy', 'Interceptions', 'Jumping',
                   'LongPassing', 'LongShots', 'Marking', 'Penalties')

# Top four features for every position in football

for i, val in data.groupby(data['Position'])[player_features].mean().iterrows():
    print('Position {}: {}, {}, {}'.format(i, *tuple(val.nlargest(4).index)))
```

```
Position CAM: Balance, Agility, Acceleration
Position CB: Jumping, Aggression, HeadingAccuracy
Position CDM: Aggression, Jumping, Balance
Position CF: Agility, Balance, Acceleration
Position CM: Balance, Agility, Acceleration
Position GK: GKReflexes, GKDiving, GKPositioning
Position LAM: Agility, Balance, Acceleration
Position LB: Acceleration, Balance, Agility
Position LCB: Jumping, Aggression, HeadingAccuracy
Position LCM: Balance, Agility, BallControl
Position LDM: Aggression, BallControl, LongPassing
Position LF: Balance, Agility, Acceleration
Position LM: Acceleration, Agility, Balance
Position LS: Acceleration, Agility, Finishing
Position LW: Acceleration, Agility, Balance
Position LWB: Acceleration, Agility, Balance
Position RAM: Agility, Balance, Acceleration
Position RB: Acceleration, Balance, Jumping
Position RCB: Jumping, Aggression, HeadingAccuracy
Position RCM: Agility, Balance, BallControl
Position RDM: Aggression, Jumping, BallControl
Position RF: Agility, Acceleration, Balance
Position RM: Acceleration, Agility, Balance
Position RS: Acceleration, Agility, Jumping
Position RW: Acceleration, Agility, Balance
Position RWB: Acceleration, Agility, Balance
Position ST: Acceleration, Jumping, Finishing
```

```
<ipython-input-58-058bc0b5a945>:13: FutureWarning: Indexing with multiple keys (implicitly converted
to a tuple of keys) will be deprecated, use a list instead.
  for i, val in data.groupby(data['Position'])[player_features].mean().iterrows():
```

Top 10 left footed footballers

In [59]:

```python
# Top 10 left footed footballers

data[data['Preferred Foot'] == 'Left'][['Name', 'Age', 'Club',
        'Nationality', 'Overall']].sort_values(by = 'Overall',
            ascending = False).head(10).style.background_gradient(cmap = 'bone')
```

Out[59]:

	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

Top 10 Right footed footballers

In [60]:

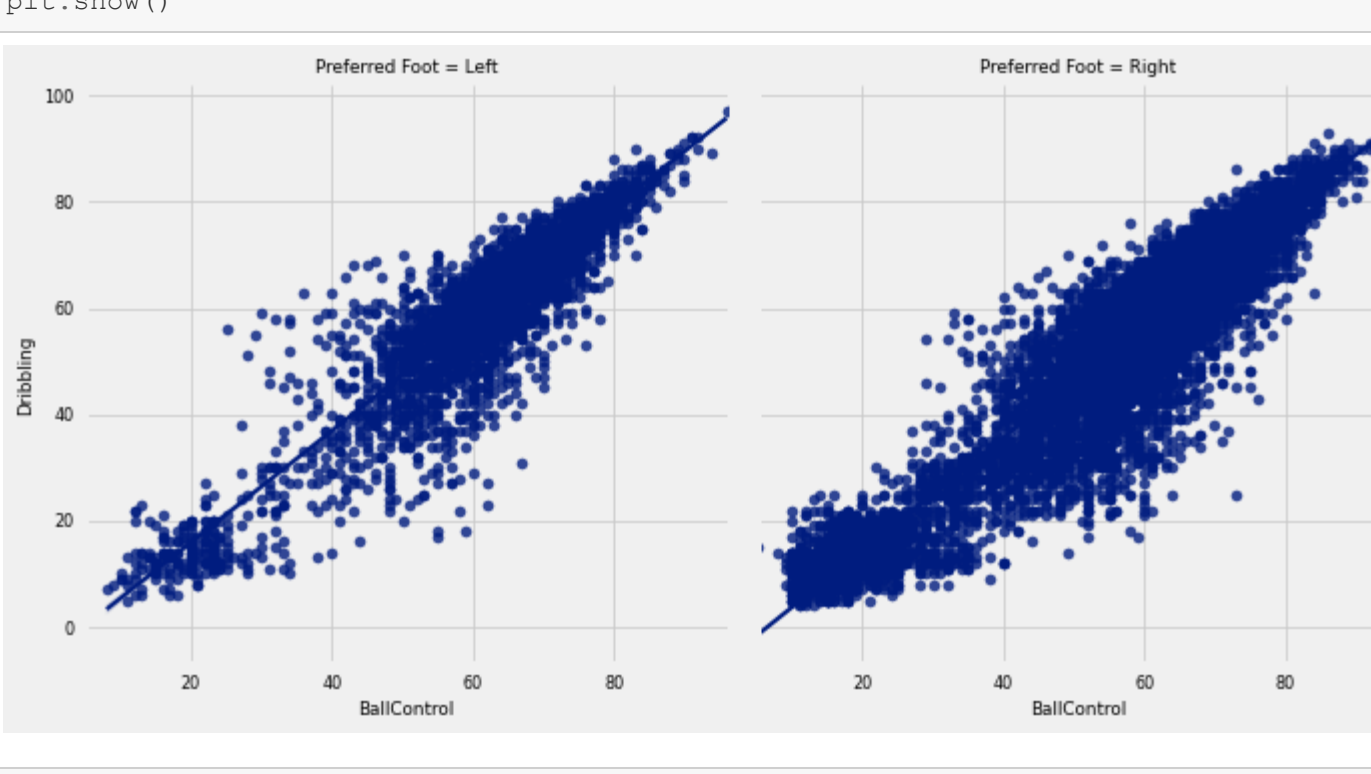
```python
# Top 10 Right footed footballers

data[data['Preferred Foot'] == 'Right'][['Name', 'Age', 'Club',
                    'Nationality', 'Overall']].sort_values(by = 'Overall',
                                ascending = False).head(10).style.background_gradient(cmap = 'copper')
```

Out[60]:

	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 [61]:

```python
# comparing the performance of left-footed and right-footed footballers
# ballcontrol vs dribbling

sns.lmplot(x = 'BallControl', y = 'Dribbling', data = data, col = 'Preferred Foot')
plt.show()
```



In [ ]:

Thank You