FIFA 20 Player Data Visualization

In this notebook, I will demonstrate data visualizaton techniques using the FIFA 19 complete player dataset. So, let's get started.

Table of Contents

1.Introduction to FIFA 20 complete player dataset 2.Import libraries 3.Read dataset 4.Data Exploration 5.Data Visualization 6.Analyse players based on Nationality 7.Analyse players based on Club 8.Profiling top players 9.Data Analysis

1. Introduction to FIFA 20 complete player dataset

The dataset includes lastest edition FIFA 2020 players attributes which are as follows -

'id', 'name', 'full_name', 'birth_date', 'age', 'height_cm', 'weight_kgs', 'positions', 'nationality', 'overall_rating', 'potential','value_euro', 'wage_euro', 'preferred_foot', 'international_reputation(1-5)', 'weak_foot(1-5)', 'skill_moves(1-5)','work_rate', 'body_type', 'release_clause_euro', 'club_team','club_rating', 'club_position', 'club_jersey_number', 'club_join_date','contract_end_year', 'national_team', 'national_rating','national_team_position', 'national_jersey_number', 'crossing','finishing', 'heading_accuracy', 'short_passing', 'volleys','dribbling', 'curve', 'freekick_accuracy', 'long_passing','ball_control', 'acceleration', 'sprint_speed', 'agility', 'reactions','balance', 'shot_power', 'jumping', 'stamina', 'strength', 'long_shots','aggression', 'interceptions', 'positioning', 'vision', 'penalties','composure', 'marking', 'standing_tackle', 'sliding_tackle','GK_diving', 'GK_handling', 'GK_kicking', 'GK_positioning','GK_reflexes', 'tags', 'traits', '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'

2. Import libraries

In [14]:
```
import numpy as np # linear algebra
import pandas as pd # data processing,(CSV file)

# for visualizations
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
```

3. Read dataset

In [3]:
```
d=pd.read_csv("C:/Users/Ajay/Downloads/fifa_cleaned.csv")
```

In [4]:
```
d
```

Out[4]:

	id	name	full_name	birth_date	age	height_cm	weight_kgs	positions	nationality	overall_rating	...	LWB	LDM	CDM	RDM	RWB	LB	LCB	CB	RCB	RB
0	158023	L. Messi	Lionel Andrés Messi Cuccittini	1987-06-24	31	170.18	72.1	CF,RW,ST	Argentina	94	...	64+2	61+2	61+2	61+2	64+2	59+2	48+2	48+2	48+2	59+2
1	190460	C. Eriksen	Christian Dannemann Eriksen	1992-02-14	27	154.94	76.2	CAM,RM,CM	Denmark	88	...	71+3	71+3	71+3	71+3	71+3	66+3	57+3	57+3	57+3	66+3
2	195864	P. Pogba	Paul Pogba	1993-03-15	25	190.50	83.9	CM,CAM	France	88	...	76+3	77+3	77+3	77+3	76+3	74+3	72+3	72+3	72+3	74+3
3	198219	L. Insigne	Lorenzo Insigne	1991-06-04	27	162.56	59.0	LW,ST	Italy	88	...	63+3	58+3	58+3	58+3	63+3	58+3	44+3	44+3	44+3	58+3
4	201024	K. Koulibaly	Kalidou Koulibaly	1991-06-20	27	187.96	88.9	CB	Senegal	88	...	73+3	77+3	77+3	77+3	73+3	76+3	85+3	85+3	85+3	76+3
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
17949	204322	R. McKenzie	Rory McKenzie	1993-10-07	25	175.26	74.8	RM,CAM,CM	Scotland	67	...	53+2	52+2	52+2	52+2	53+2	50+2	46+2	46+2	46+2	50+2
17950	239762	M. Sipľak	Michal Sipľak	1996-02-02	23	182.88	79.8	LB	Slovakia	59	...	57+2	55+2	55+2	55+2	57+2	57+2	58+2	58+2	58+2	57+2
17951	235155	J. Bekkema	Jan Bekkema	1996-04-09	22	185.42	89.8	GK	Netherlands	59	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
17952	244883	A. Al Yami	Abdulrahman Al Yami	1997-06-19	21	175.26	64.9	ST,LM	Saudi Arabia	59	...	41+2	35+2	35+2	35+2	41+2	39+2	32+2	32+2	32+2	39+2
17953	247187	Júnior Brumado	José Francisco dos Santos Júnior	1999-05-15	19	190.50	79.8	ST	Brazil	59	...	41+2	40+2	40+2	40+2	41+2	40+2	40+2	40+2	40+2	40+2

17954 rows × 92 columns

4. Data Exploration

In [5]:
```
# shape of the data
d.shape
```

Out[5]: (17954, 92)

In [6]:
```
# preview dataset
d.head()
```

Out[6]:

	id	name	full_name	birth_date	age	height_cm	weight_kgs	positions	nationality	overall_rating	...	LWB	LDM	CDM	RDM	RWB	LB	LCB	CB	RCB	RB
0	158023	L. Messi	Lionel Andrés Messi Cuccittini	1987-06-24	31	170.18	72.1	CF,RW,ST	Argentina	94	...	64+2	61+2	61+2	61+2	64+2	59+2	48+2	48+2	48+2	59+2
1	190460	C. Eriksen	Christian Dannemann Eriksen	1992-02-14	27	154.94	76.2	CAM,RM,CM	Denmark	88	...	71+3	71+3	71+3	71+3	71+3	66+3	57+3	57+3	57+3	66+3
2	195864	P. Pogba	Paul Pogba	1993-03-15	25	190.50	83.9	CM,CAM	France	88	...	76+3	77+3	77+3	77+3	76+3	74+3	72+3	72+3	72+3	74+3
3	198219	L. Insigne	Lorenzo Insigne	1991-06-04	27	162.56	59.0	LW,ST	Italy	88	...	63+3	58+3	58+3	58+3	63+3	58+3	44+3	44+3	44+3	58+3
4	201024	K. Koulibaly	Kalidou Koulibaly	1991-06-20	27	187.96	88.9	CB	Senegal	88	...	73+3	77+3	77+3	77+3	73+3	76+3	85+3	85+3	85+3	76+3

5 rows × 92 columns

In [8]:
```
# view column names
d.columns
```

Out[8]:
```
Index(['id', 'name', 'full_name', 'birth_date', 'age', 'height_cm',
       'weight_kgs', 'positions', 'nationality', 'overall_rating', 'potential',
       'value_euro', 'wage_euro', 'preferred_foot',
       'international_reputation(1-5)', 'weak_foot(1-5)', 'skill_moves(1-5)',
       'work_rate', 'body_type', 'release_clause_euro', 'club_team',
       'club_rating', 'club_position', 'club_jersey_number', 'club_join_date',
       'contract_end_year', 'national_team', 'national_rating',
       'national_team_position', 'national_jersey_number', 'crossing',
       'finishing', 'heading_accuracy', 'short_passing', 'volleys',
       'dribbling', 'curve', 'freekick_accuracy', 'long_passing',
       'ball_control', 'acceleration', 'sprint_speed', 'agility', 'reactions',
       'balance', 'shot_power', 'jumping', 'stamina', 'strength', 'long_shots',
       'aggression', 'interceptions', 'positioning', 'vision', 'penalties',
       'composure', 'marking', 'standing_tackle', 'sliding_tackle',
       'GK_diving', 'GK_handling', 'GK_kicking', 'GK_positioning',
       'GK_reflexes', 'tags', 'traits', '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'],
      dtype='object')
```

In [9]:
```
# view dataframe summary
d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17954 entries, 0 to 17953
Data columns (total 92 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   id                              17954 non-null  int64
 1   name                            17954 non-null  object
 2   full_name                       17954 non-null  object
 3   birth_date                      17954 non-null  object
 4   age                             17954 non-null  int64
 5   height_cm                       17954 non-null  float64
 6   weight_kgs                      17954 non-null  float64
 7   positions                       17954 non-null  object
 8   nationality                     17954 non-null  object
 9   overall_rating                  17954 non-null  int64
 10  potential                       17954 non-null  int64
 11  value_euro                      17699 non-null  float64
 12  wage_euro                       17708 non-null  float64
 13  preferred_foot                  17954 non-null  object
 14  international_reputation(1-5)   17954 non-null  int64
 15  weak_foot(1-5)                  17954 non-null  int64
 16  skill_moves(1-5)                17954 non-null  int64
 17  work_rate                       17954 non-null  object
 18  body_type                       17954 non-null  object
 19  release_clause_euro             16117 non-null  float64
 20  club_team                       17940 non-null  object
 21  club_rating                     17940 non-null  float64
 22  club_position                   17940 non-null  object
 23  club_jersey_number              17940 non-null  float64
 24  club_join_date                  16018 non-null  object
 25  contract_end_year               17593 non-null  object
 26  national_team                   857 non-null    object
 27  national_rating                 857 non-null    float64
 28  national_team_position          857 non-null    object
 29  national_jersey_number          857 non-null    float64
 30  crossing                        17954 non-null  int64
 31  finishing                       17954 non-null  int64
 32  heading_accuracy                17954 non-null  int64
 33  short_passing                   17954 non-null  int64
 34  volleys                         17954 non-null  int64
 35  dribbling                       17954 non-null  int64
 36  curve                           17954 non-null  int64
 37  freekick_accuracy               17954 non-null  int64
 38  long_passing                    17954 non-null  int64
 39  ball_control                    17954 non-null  int64
 40  acceleration                    17954 non-null  int64
 41  sprint_speed                    17954 non-null  int64
 42  agility                         17954 non-null  int64
 43  reactions                       17954 non-null  int64
 44  balance                         17954 non-null  int64
 45  shot_power                      17954 non-null  int64
 46  jumping                         17954 non-null  int64
 47  stamina                         17954 non-null  int64
 48  strength                        17954 non-null  int64
 49  long_shots                      17954 non-null  int64
 50  aggression                      17954 non-null  int64
 51  interceptions                   17954 non-null  int64
 52  positioning                     17954 non-null  int64
```

```
 53  vision                    17954 non-null  int64
 54  penalties                 17954 non-null  int64
 55  composure                 17954 non-null  int64
 56  marking                   17954 non-null  int64
 57  standing_tackle           17954 non-null  int64
 58  sliding_tackle            17954 non-null  int64
 59  GK_diving                 17954 non-null  int64
 60  GK_handling               17954 non-null  int64
 61  GK_kicking                17954 non-null  int64
 62  GK_positioning            17954 non-null  int64
 63  GK_reflexes               17954 non-null  int64
 64  tags                      1417 non-null   object
 65  traits                    8137 non-null   object
 66  LS                        15889 non-null  object
 67  ST                        15889 non-null  object
 68  RS                        15889 non-null  object
 69  LW                        15889 non-null  object
 70  LF                        15889 non-null  object
 71  CF                        15889 non-null  object
 72  RF                        15889 non-null  object
 73  RW                        15889 non-null  object
 74  LAM                       15889 non-null  object
 75  CAM                       15889 non-null  object
 76  RAM                       15889 non-null  object
 77  LM                        15889 non-null  object
 78  LCM                       15889 non-null  object
 79  CM                        15889 non-null  object
 80  RCM                       15889 non-null  object
 81  RM                        15889 non-null  object
 82  LWB                       15889 non-null  object
 83  LDM                       15889 non-null  object
 84  CDM                       15889 non-null  object
 85  RDM                       15889 non-null  object
 86  RWB                       15889 non-null  object
 87  LB                        15889 non-null  object
 88  LCB                       15889 non-null  object
 89  CB                        15889 non-null  object
 90  RCB                       15889 non-null  object
 91  RB                        15889 non-null  object
dtypes: float64(9), int64(41), object(42)
memory usage: 12.6+ MB
```

In [11]:
```
# check for missing values
d.isnull().sum()
```

Out[11]:
```
id              0
name            0
full_name       0
birth_date      0
age             0
               ...
LB           2065
LCB          2065
CB           2065
RCB          2065
RB           2065
Length: 92, dtype: int64
```

5. Data Visualization

Distribution of preferred foot

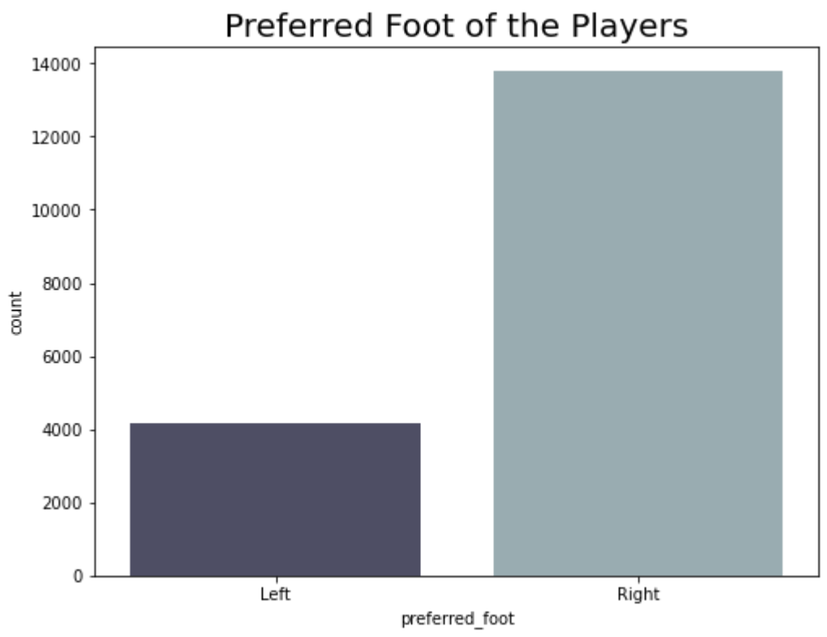
The distribution of preferred foot can be obtained as follows -

In [12]:
```
d['preferred_foot'].value_counts()
```

Out[12]:
```
Right    13781
Left      4173
Name: preferred_foot, dtype: int64
```

In [16]:
```
plt.figure(figsize=(8,6))
sns.countplot(d['preferred_foot'], palette = 'bone')
plt.title('Preferred Foot of the Players', fontsize = 20)
plt.show()
```

```
C:\New folder\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument wil
l be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [18]:
```
#Now the above countplots are great but they do not show up the exact percentages of players.
#We can show the exact percentages of players as follows-
d['preferred_foot'].value_counts()/len(d)
```

Out[18]:
```
Right    0.767573
Left     0.232427
Name: preferred_foot, dtype: float64
```

In [21]:
```
#We can use a pie chart to visualize the exact percentages.
labels = d['preferred_foot'].value_counts().index
size = d['preferred_foot'].value_counts()
colors=['cyan','red']
plt.pie(size, labels = labels, colors = colors, shadow = True, autopct='%1.1f%%',startangle = 90)
plt.title('Distribution of Preferred Foot among players', fontsize = 20)
plt.legend()
plt.show()
```

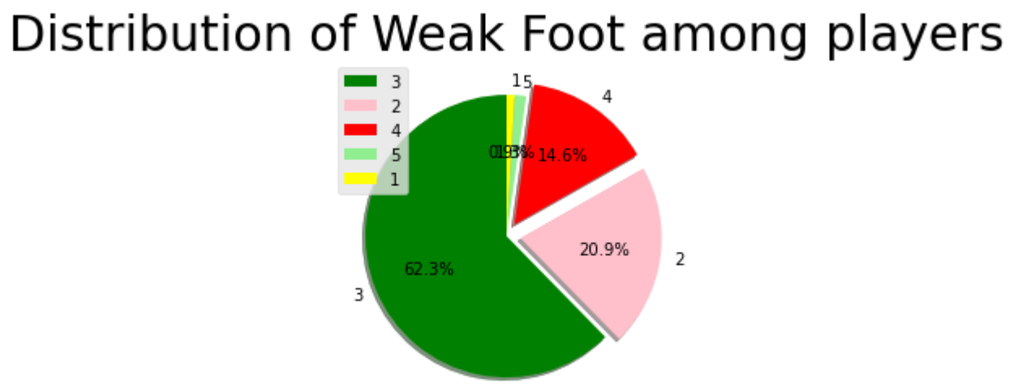
Distribution of weak foot

#The distribution of weak foot can be obtained as follows -

In [22]:
```
d['weak_foot(1-5)'].value_counts()
```

Out[22]:
```
3    11193
2     3746
4     2615
5      239
1      161
Name: weak_foot(1-5), dtype: int64
```

In [78]:
```
#We can visualize distribution of Weak foot as follows -
labels = d['weak_foot(1-5)'].value_counts().index
size = d['weak_foot(1-5)'].value_counts()
colors=['green','pink','red','lightgreen','yellow']
explode = [0, 0.1, 0.1, 0, 0]
plt.pie(size, labels = labels, colors = colors, explode = explode, shadow = True, autopct='%1.1f%%',startangle = 90)
plt.title('Distribution of Weak Foot among players', fontsize = 30)
plt.legend()
plt.show()
```

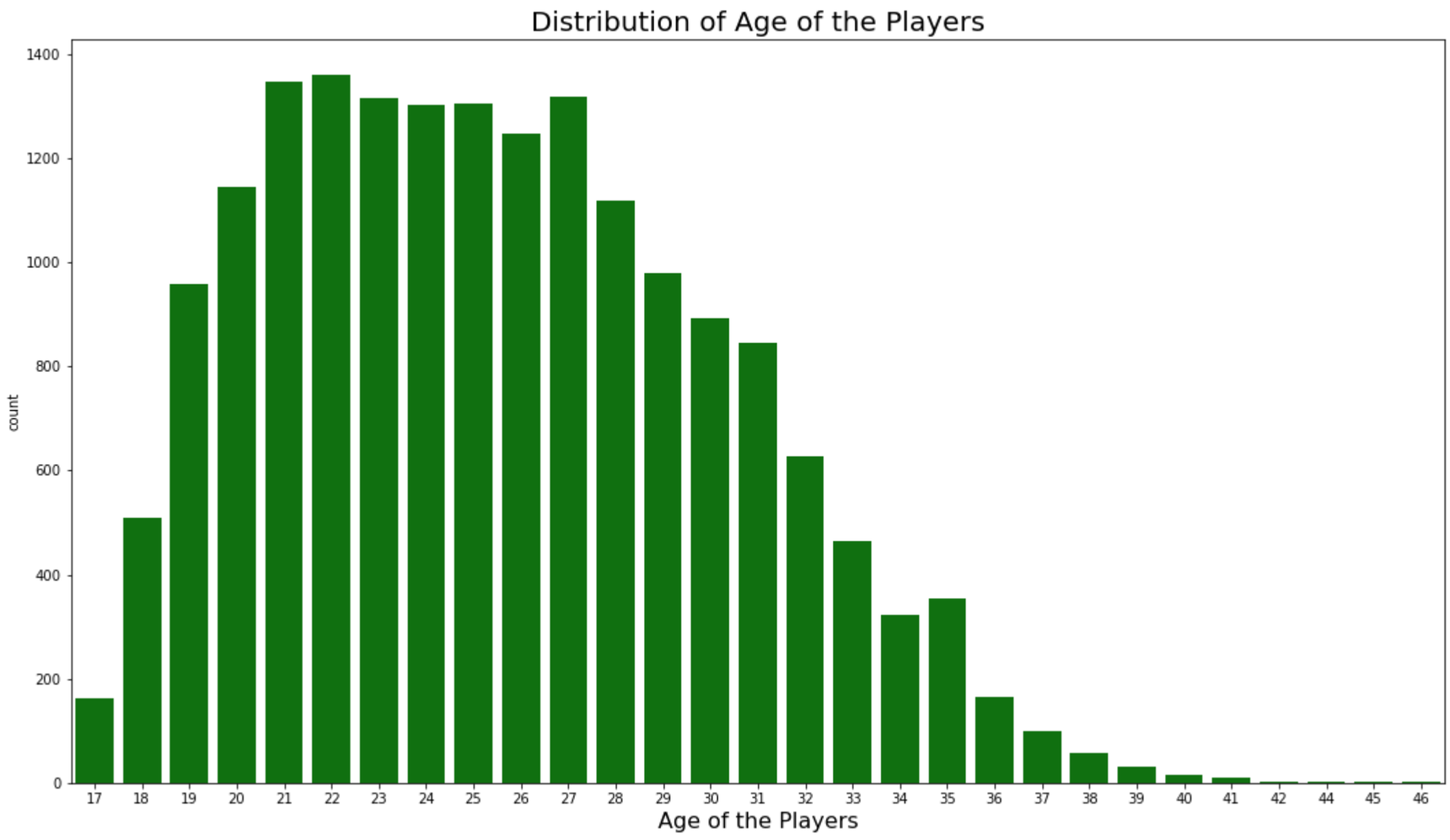


Age distribution of players

#We can use a histogram to visualize the age distribution of players

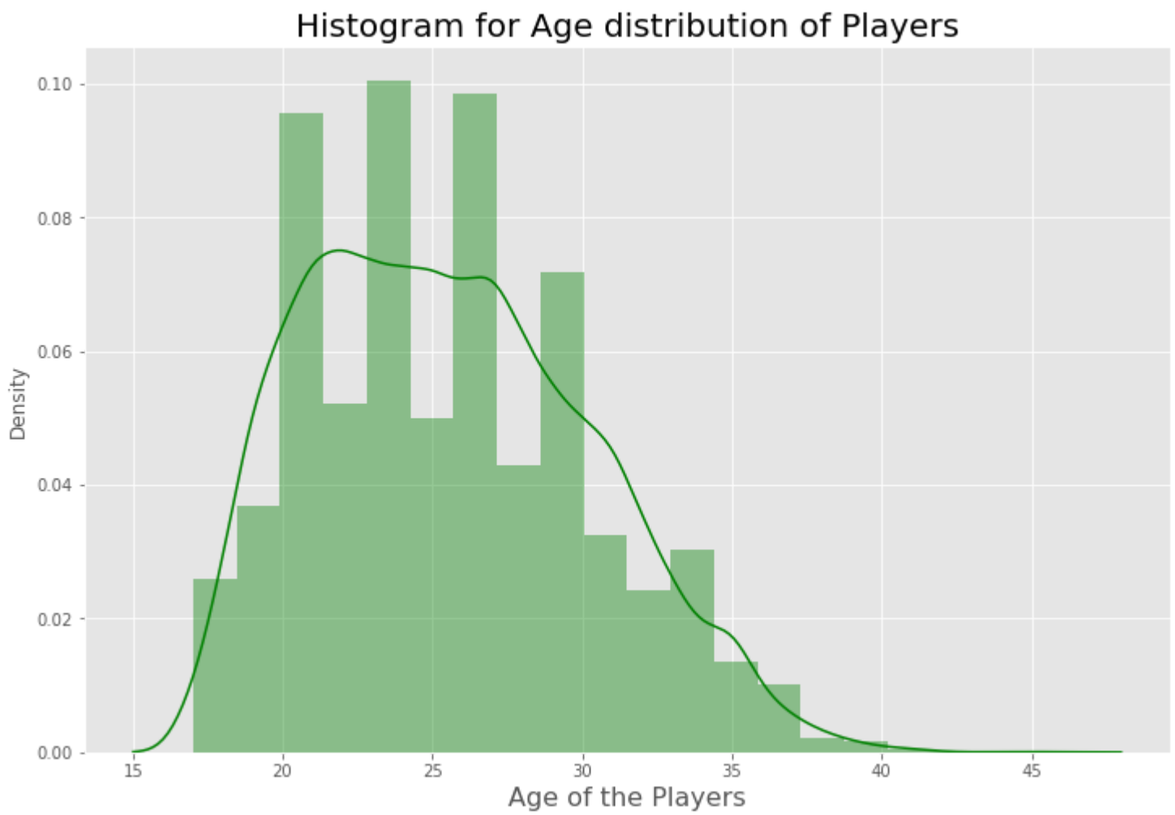
In [27]:
```
x = d['age']
plt.figure(figsize=(18,10))
ax = sns.countplot(x, color='g')
ax.set_xlabel(xlabel = 'Age of the Players', fontsize = 16)
ax.set_title(label = 'Distribution of Age of the Players', fontsize = 20)
plt.show()
```

```
C:\New folder\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument wil
l be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [28]:
```
x = d['age']
plt.figure(figsize = (12, 8))
plt.style.use('ggplot')
ax = sns.distplot(x, bins = 20, kde = True, color='g')
ax.set_xlabel(xlabel = 'Age of the Players', fontsize = 16)
ax.set_title(label = 'Histogram for Age distribution of Players', fontsize = 20)
plt.show()
```

```
C:\New folder\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



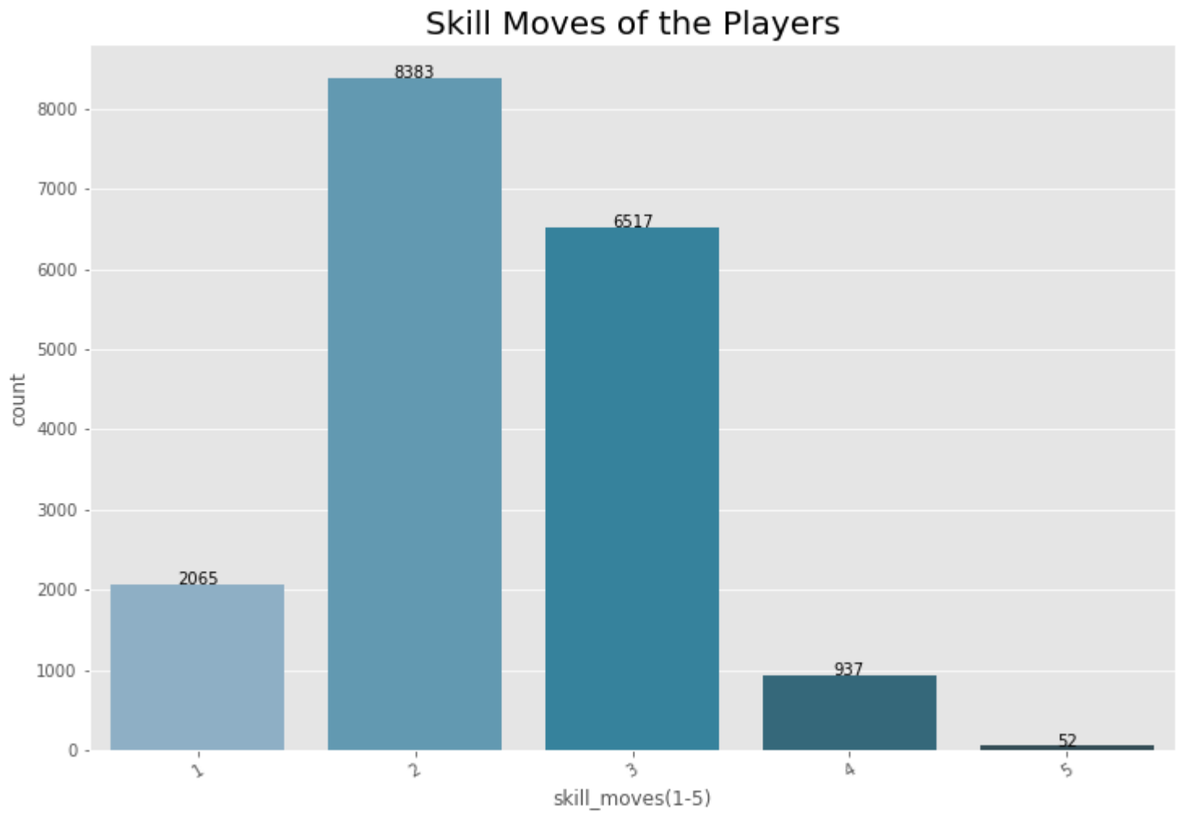
Skill Moves of the Players

##The different work rate of the players can be given as follows -

In [33]:
```
d['skill_moves(1-5)'].value_counts()
```

Out[33]:
```
2    8383
3    6517
1    2065
4     937
5      52
Name: skill_moves(1-5), dtype: int64
```

In [37]:
```
#We can visualize the skill moves of the players as follows-
fig, ax = plt.subplots(figsize=(12,8))
graph = sns.countplot(ax=ax,x=d['skill_moves(1-5)'], data=d, palette = 'PuBuGn_d')
graph.set_title('Skill Moves of the Players', fontsize = 20)
graph.set_xticklabels(graph.get_xticklabels(), rotation=30)
for p in graph.patches:
    height = p.get_height()
    graph.text(p.get_x()+p.get_width()/2., height + 0.1,height ,ha="center")
```

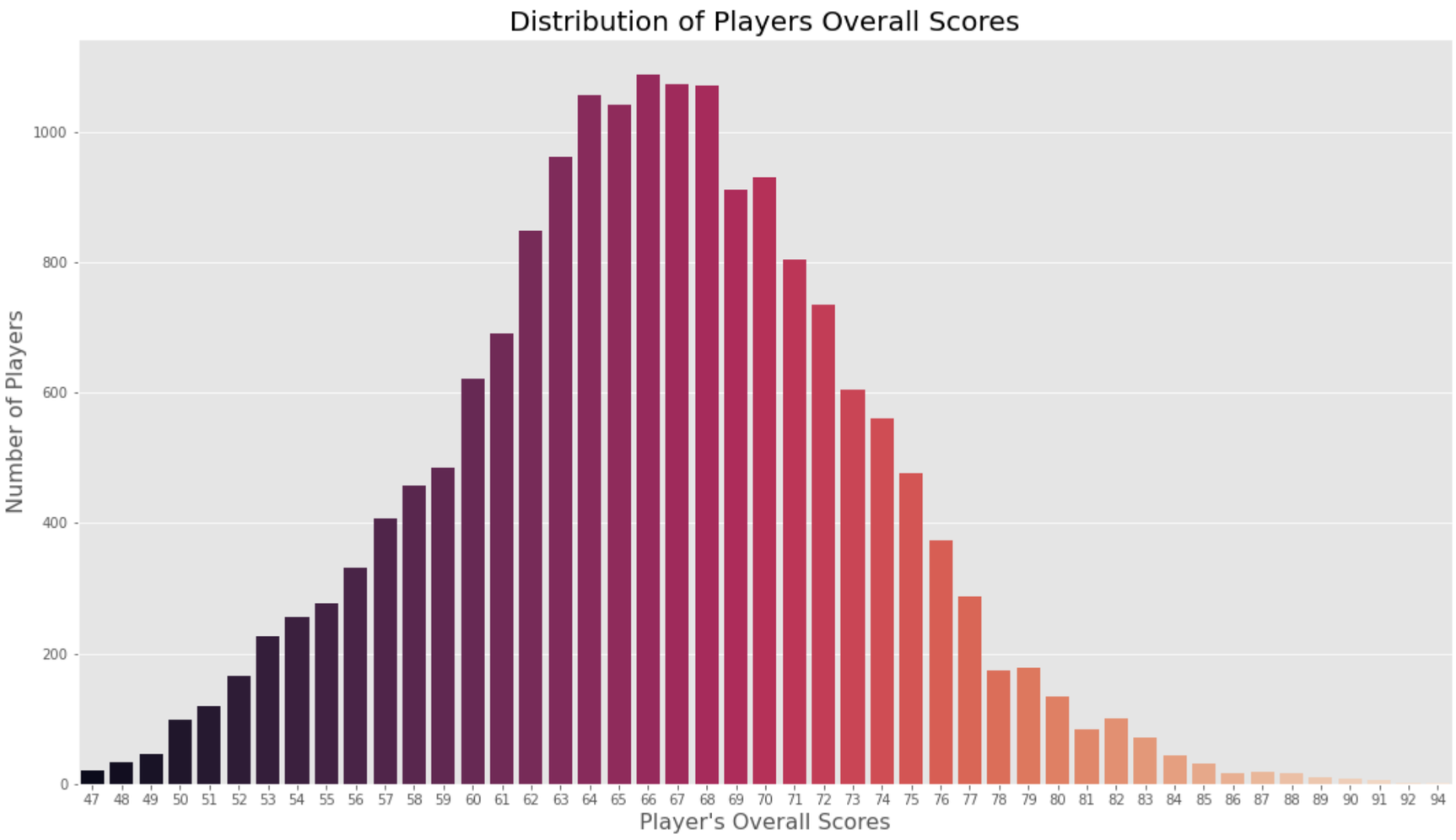


Overall Score of the Players

The overall score of the players can be visualized with histogram as follows -

In [39]:
```
x = d['overall_rating']
plt.figure(figsize=(18,10))
ax = sns.countplot(x, palette='rocket')
ax.set_xlabel(xlabel = "Player's Overall Scores", fontsize = 16)
ax.set_ylabel(ylabel = 'Number of Players', fontsize = 16)
ax.set_title(label = 'Distribution of Players Overall Scores', fontsize = 20)
plt.show()
```

```
C:\New folder\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument wil
l be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



Count of Players with Positions

##The number of players at different positions can be found as follows -

In [40]:
```
d['positions'].value_counts()
```

Out[40]:
```
CB              2243
GK              2065
ST              1747
CM               764
CDM,CM           709
                ...
RB,ST              1
CDM,CM,CB,LM       1
CM,RWB,CDM         1
RM,RW,CAM,ST       1
LW,CAM,LM,CM       1
Name: positions, Length: 890, dtype: int64
```

6. Analyse players based on nationality

We can calculate number of countries as follows -

In [47]:
```
d['nationality'].nunique()
```

Out[47]:
```
160
```

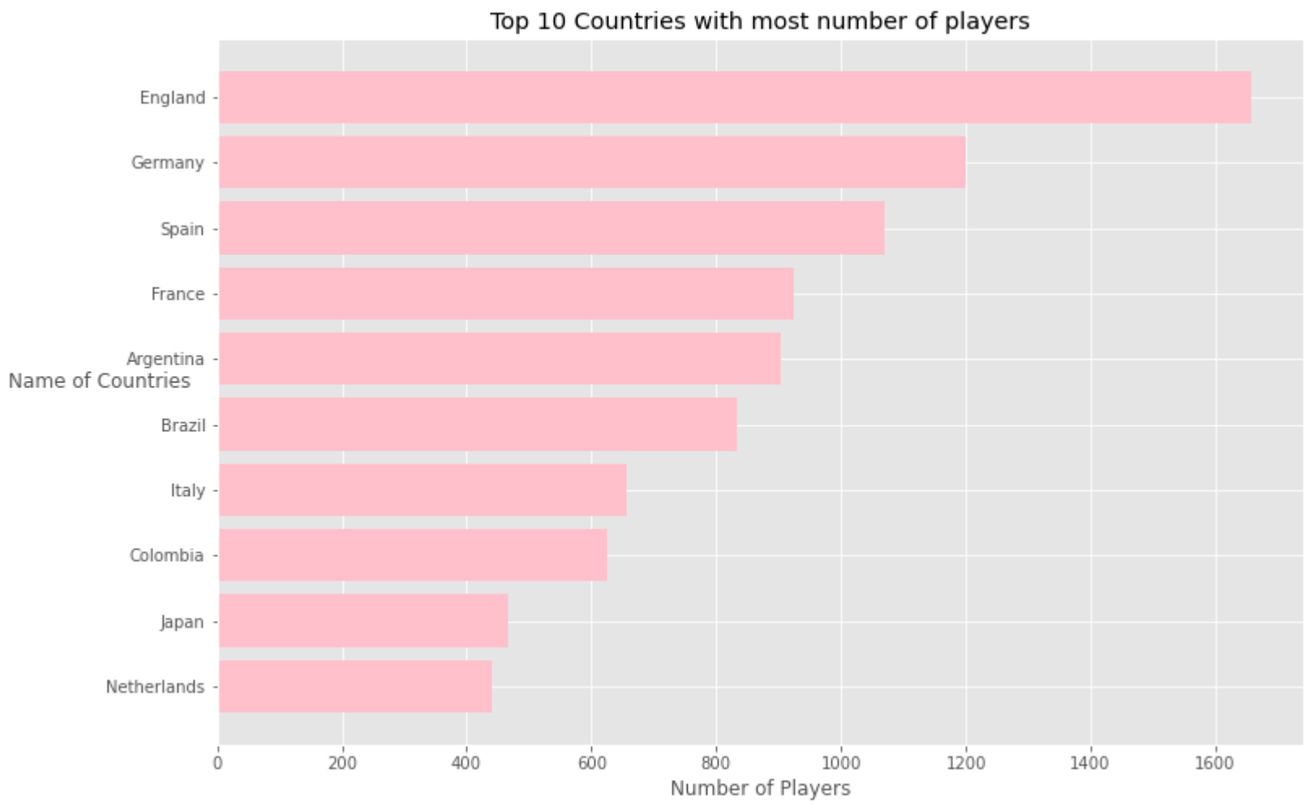
In [48]:
```
#The names of countries are as follows -
d['nationality'].unique()
```

Out[48]:
```
array(['Argentina', 'Denmark', 'France', 'Italy', 'Senegal',
       'Netherlands', 'Germany', 'Uruguay', 'Spain', 'Belgium', 'Egypt',
       'Slovakia', 'Brazil', 'Croatia', 'Costa Rica', 'Colombia',
       'Morocco', 'Portugal', 'Sweden', 'Bosnia Herzegovina', 'Mexico',
       'England', 'Austria', 'Iceland', 'Hungary', 'Wales', 'Ukraine',
       'Central African Rep.', 'Serbia', 'Ivory Coast', 'Cameroon',
       'Paraguay', 'Australia', 'Algeria', 'Romania', 'Russia', 'Israel',
       'Switzerland', 'Chile', 'Tunisia', 'Turkey', 'Nigeria', 'Peru',
       'Norway', 'Greece', 'United States', 'Venezuela', 'Iran',
       'Equatorial Guinea', 'Cape Verde', 'Tanzania', 'Scotland',
       'China PR', 'Kosovo', 'Montenegro', 'Canada', 'Madagascar', 'Mali',
       'Ghana', 'Guinea', 'Poland', 'Cuba', 'Northern Ireland', 'Japan',
       'New Zealand', 'South Africa', 'Republic of Ireland', 'Ecuador',
       'Burkina Faso', 'Czech Republic', 'Slovenia', 'Belarus', 'Gabon',
       'FYR Macedonia', 'Curacao', 'DR Congo', 'Honduras', 'Sierra Leone',
       'Guinea Bissau', 'Saudi Arabia', 'Fiji', 'Korea Republic',
       'Zambia', 'Syria', 'Armenia', 'Georgia', 'Angola', 'Zimbabwe',
       'Congo', 'Eritrea', 'Iraq', 'Albania', 'Bolivia', 'Gambia',
       'Jamaica', 'Burundi', 'Uganda', 'Benin', 'Suriname', 'Finland',
       'Lithuania', 'Togo', 'El Salvador', 'Korea DPR', 'Panama',
       'Moldova', 'Haiti', 'Comoros', 'Azerbaijan', 'Trinidad & Tobago',
       'Liberia', 'Bulgaria', 'Kenya', 'Chad', 'Faroe Islands',
       'Kazakhstan', 'Estonia', 'Montserrat', 'Dominican Republic',
       'Thailand', 'India', 'Andorra', 'Sudan', 'Hong Kong', 'Malta',
       'Afghanistan', 'Uzbekistan', 'Palestine', 'Yemen', 'Luxembourg',
       'Mauritania', 'Indonesia', 'Bermuda', 'Grenada', 'Philippines',
       'Liechtenstein', 'Guyana', 'St Kitts Nevis', 'Antigua & Barbuda',
       'Cyprus', 'Rwanda', 'South Sudan', 'Papua New Guinea', 'Latvia',
       'Vietnam', 'Nicaragua', 'Ethiopia', 'Barbados', 'Jordan',
       'St Lucia', 'Guatemala', 'Namibia', 'Guam', 'Mozambique', 'Kuwait',
       'Libya', 'New Caledonia', 'Oman', 'São Tomé & Príncipe',
       'United Arab Emirates'], dtype=object)
```

In [50]:
```
#The top 10 countries with most number of players are as follows -
top_countries = d['nationality'].value_counts().head(10)

top_countries
```

Out[50]:
```
England        1658
Germany        1199
Spain          1070
France          925
Argentina       904
Brazil          832
Italy           655
Colombia        624
Japan           466
Netherlands     441
Name: nationality, dtype: int64
```

In [53]:
```
fig, ax = plt.subplots(figsize=(12,8))
x = top_countries.values
y = top_countries.index
ax.barh(y, x, align='center', color='pink')
ax.invert_yaxis()  # labels read top-to-bottom
ax.set_xlabel('Number of Players')
ax.set_ylabel('Name of Countries', rotation=0)
ax.set_title('Top 10 Countries with most number of players')
plt.show()
```



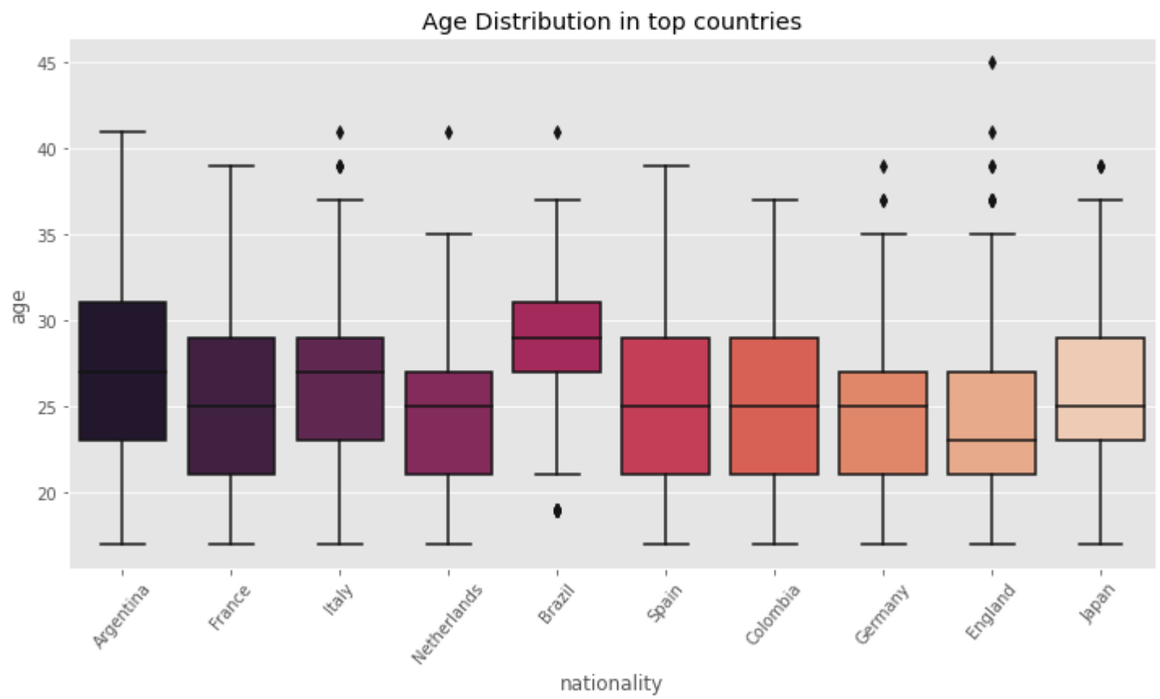
In [54]:
```
top_countries_name = top_countries.index

top_countries_name
```

Out[54]:
```
Index(['England', 'Germany', 'Spain', 'France', 'Argentina', 'Brazil', 'Italy',
       'Colombia', 'Japan', 'Netherlands'],
      dtype='object')
```

Age distribution from top countries

#We can draw a box plot to check the age distribution from top countries.

In [55]:
```
df_country_age = d.loc[d['nationality'].isin(top_countries_name) & d['age']]
plt.figure(1 , figsize = (12,6))
sns.boxplot(x = 'nationality' , y = 'age' , data = df_country_age, palette='rocket')
plt.title('Age Distribution in top countries')
plt.xticks(rotation = 50)
plt.show()
```



7. Analyse players based on club

In [58]:
```
# How many clubs are there?
d['club_team'].nunique()
```

Out[58]: 687

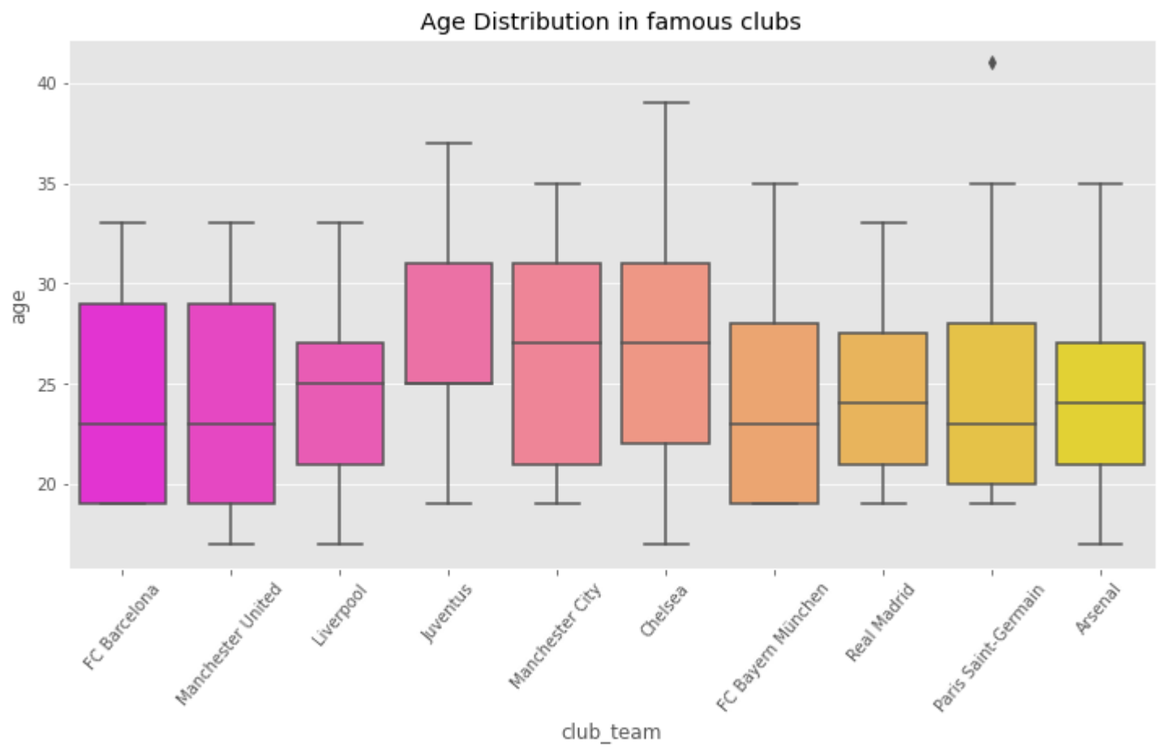
In [61]:
```
#What are the club names?

d['club_team'].unique()
```

Out[61]:
```
array(['FC Barcelona', 'Tottenham Hotspur', 'Manchester United', 'Napoli',
       'Liverpool', 'Paris Saint-Germain', 'Manchester City',
       'FC Bayern München', 'Real Madrid', 'Atlético Madrid', 'Juventus',
       'Inter', 'Chelsea', 'Lazio', 'Ajax', 'RB Leipzig', 'Portugal',
       'FC Porto', 'Olympique Lyonnais', 'Vissel Kobe', 'LA Galaxy',
       'Medipol Başakşehir FK', 'Real Betis', 'Belgium', 'Sevilla FC',
       'Milan', 'Guangzhou Evergrande Taobao FC', 'Bayer 04 Leverkusen',
       'PSV', 'Arsenal', 'Valencia CF', 'Grêmio', 'Cruzeiro',
       'Atlético Mineiro', 'SV Werder Bremen', 'Dalian YiFang FC',
       'West Ham United', 'Italy', 'Everton', 'Al Nassr',
       'Olympique de Marseille', 'Shakhtar Donetsk', 'VfL Wolfsburg',
       'Torino', 'Borussia Mönchengladbach', 'FC Schalke 04',
       'Villarreal CF', 'Watford', 'Borussia Dortmund', 'Leicester City',
       'LOSC Lille', 'Real Sociedad', 'Fluminense', 'RC Celta',
       'Sporting CP', 'Brighton & Hove Albion', 'Cameroon', 'Fiorentina',
       'Beşiktaş JK', 'Athletic Club de Bilbao', 'FC Nantes',
       'Stoke City', 'Chievo Verona', 'DC United', 'Racing Club',
       'Hebei China Fortune FC', 'RCD Espanyol', 'Lokomotiv Moscow',
       'PFC CSKA Moscow', 'Tigres U.A.N.L.', 'Sweden', 'OGC Nice',
```

Age distribution in famous clubs

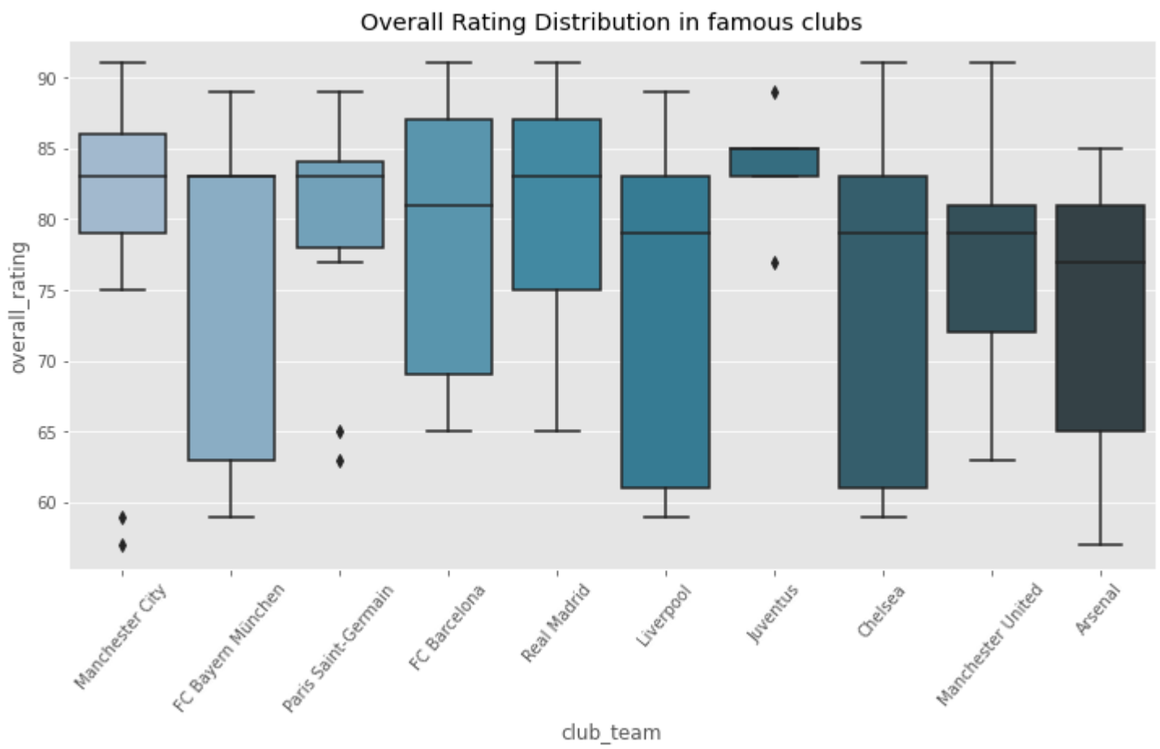
In [62]:
```
# What are some popular clubs?
clubs = ['FC Barcelona','Real Madrid','Juventus','Liverpool','Manchester United',
         'Chelsea','Arsenal','Paris Saint-Germain' ,'FC Bayern München','Manchester City']
```

In [64]:
```
df_club_age = d.loc[d['club_team'].isin(clubs) & d['age']]
plt.figure(1 , figsize = (12,6))
sns.boxplot(x = 'club_team', y = 'age' , data = df_club_age, palette='spring')
plt.title('Age Distribution in famous clubs')
plt.xticks(rotation = 50)
plt.show()
```



Overall Rating in famous clubs

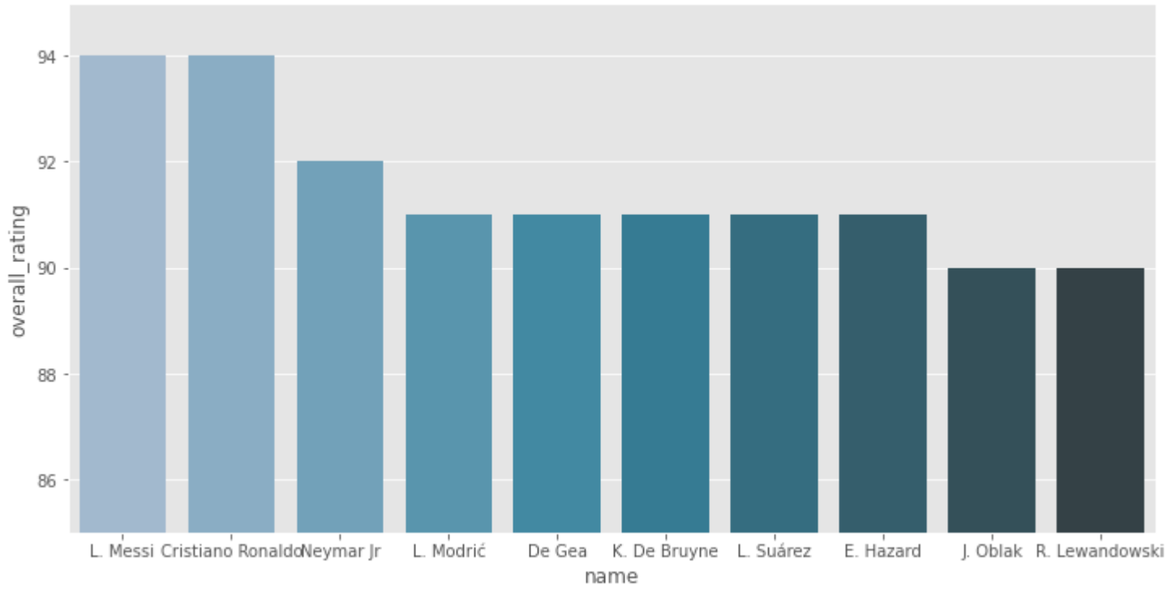
In [65]:
```
df_club_rating = d.loc[d['club_team'].isin(clubs) & d['overall_rating']]
plt.figure(1 , figsize = (12,6))
sns.boxplot(x = 'club_team' , y = 'overall_rating' , data = df_club_rating, palette='PuBuGn_d')
plt.title('Overall Rating Distribution in famous clubs')
plt.xticks(rotation = 50)
plt.show()
```



8. Profiling top players

The 10 best players are-

In [66]:
```
df_best_players = pd.DataFrame.copy(d.sort_values(by ='overall_rating',ascending = False ).head(10))
plt.figure(1,figsize = (12,6))
sns.barplot(x ='name' , y = 'overall_rating' , data = df_best_players, palette='PuBuGn_d')
plt.ylim(85,95)
plt.show()
```



9. Data Analysis

##Who are the top 10 eldest players?

In [68]:
```
d.sort_values(by = 'age' , ascending = False)[['name','club_team','nationality','overall_rating', 'age' ]].head()
```

Out[68]:

	name	club_team	nationality	overall_rating	age
1304	O. Pérez	Pachuca	Mexico	71	46
4865	K. Pilkington	Cambridge United	England	48	45
5226	T. Warner	Accrington Stanley	Trinidad & Tobago	53	44
9950	H. Sulaimani	Al Ahli	Saudi Arabia	64	42
13635	B. Nivet	ESTAC Troyes	France	69	42

The top 10 youngest players are given by -

d.sort_values(by = 'age' , ascending = True)[['name','club_team','nationality','overall_rating', 'age']].head()

The Best Freekick Takers

The best free-kick takers are given by-

In [73]:
```
d.sort_values(by = 'freekick_accuracy' , ascending = False)[['name','club_team','nationality','age','freekick_accuracy']].head()
```

Out[73]:

	name	club_team	nationality	age	freekick_accuracy
0	L. Messi	FC Barcelona	Argentina	31	94
84	S. Giovinco	Italy	Italy	32	93
17901	M. Pjanić	Juventus	Bosnia Herzegovina	28	92
17160	E. Bardhi	Levante UD	FYR Macedonia	23	91
17623	H. Çalhanoğlu	Milan	Turkey	25	90

The Best Dribbler

The best dribbler is-

In [74]:
```
d.sort_values(by = 'dribbling' , ascending = False)[['name','club_team','nationality','overall_rating', 'age','dribbling']].head()
```

Out[74]:

	name	club_team	nationality	overall_rating	age	dribbling
0	L. Messi	FC Barcelona	Argentina	94	31	97
17943	Neymar Jr	Paris Saint-Germain	Brazil	92	27	96
17940	E. Hazard	Chelsea	Belgium	91	28	95
28	Isco	Real Madrid	Spain	87	26	94
17884	Y. Brahimi	FC Porto	Algeria	85	29	93

The Best Finisher

In [ ]:
```
Who is the best finisher?
```

In [75]:
```
d.sort_values(by = 'finishing' , ascending = False)[['name','club_team','nationality','overall_rating', 'age','finishing']].head()
```

Out[75]:

	name	club_team	nationality	overall_rating	age	finishing
0	L. Messi	FC Barcelona	Argentina	94	31	95
17937	H. Kane	Tottenham Hotspur	England	90	25	94
17944	Cristiano Ronaldo	Juventus	Portugal	94	34	94
7	S. Agüero	Manchester City	Argentina	89	30	93
17938	L. Suárez	FC Barcelona	Uruguay	91	32	92

Fastest Players

In [77]:
```
#The fastest players on the planet are -
d.sort_values(by = 'sprint_speed' , ascending = False)[['name','club_team','nationality','overall_rating', 'age','sprint_speed']].head()
```

Out[77]:

	name	club_team	nationality	overall_rating	age	sprint_speed
6	K. Mbappé	Paris Saint-Germain	France	88	20	96
17915	L. Sané	Manchester City	Germany	86	23	96
16157	Adama	Wolverhampton Wanderers	Spain	74	23	96
17924	G. Bale	Real Madrid	Wales	88	29	95
16708	J. Damm	Tigres U.A.N.L.	Mexico	75	26	95