

# fifa-2019

May 4, 2023

## 0.1 FIFA(2019) Dataset

### 0.1.1 Exploratory data analysis (EDA)

```
[1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

```
[2]: sns.set(style="ticks") #for grid line
flatui = ["#9b59b6", "#3498db", "#95a5a6", "#e74c3c", "#34495e", "#2ecc71"] #_
    ↪defining the colour palette
flatui = sns.color_palette(flatui)
```

```
[3]: #Wordcloud uses the text sizes to represent the frequency of the text
from wordcloud import WordCloud
```

```
[4]: # reading the data
df= pd.read_csv('FIFA_data.csv')
df.head()
```

```
[4]: Unnamed: 0      ID      Name  Age  \
0      0  158023      L. Messi   31
1      1  20801  Cristiano Ronaldo  33
2      2  190871      Neymar Jr   26
3      3  193080      De Gea     27
4      4  192985      K. De Bruyne  27

                                     Photo Nationality  \
0  https://cdn.sofifa.org/players/4/19/158023.png  Argentina
1  https://cdn.sofifa.org/players/4/19/20801.png    Portugal
2  https://cdn.sofifa.org/players/4/19/190871.png     Brazil
3  https://cdn.sofifa.org/players/4/19/193080.png     Spain
4  https://cdn.sofifa.org/players/4/19/192985.png     Belgium
```

	Flag	Overall	Potential	\
0	https://cdn.sofifa.org/flags/52.png	94	94	
1	https://cdn.sofifa.org/flags/38.png	94	94	
2	https://cdn.sofifa.org/flags/54.png	92	93	
3	https://cdn.sofifa.org/flags/45.png	91	93	
4	https://cdn.sofifa.org/flags/7.png	91	92	

	Club	... Composure	Marking	StandingTackle	SlidingTackle	\
0	FC Barcelona	...	96.0	33.0	28.0	26.0
1	Juventus	...	95.0	28.0	31.0	23.0
2	Paris Saint-Germain	...	94.0	27.0	24.0	33.0
3	Manchester United	...	68.0	15.0	21.0	13.0
4	Manchester City	...	88.0	68.0	58.0	51.0

	GK Diving	GK Handling	GK Kicking	GK Positioning	GK Reflexes	Release Clause
0	6.0	11.0	15.0	14.0	8.0	€226.5M
1	7.0	11.0	15.0	14.0	11.0	€127.1M
2	9.0	9.0	15.0	15.0	11.0	€228.1M
3	90.0	85.0	87.0	88.0	94.0	€138.6M
4	15.0	13.0	5.0	10.0	13.0	€196.4M

[5 rows x 89 columns]

```
[5]: # checking the number of rows and columns in the dataset
df.shape
```

```
[5]: (18207, 89)
```

```
[6]: # Printing a concise summary of the DataFrame.
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18207 entries, 0 to 18206
Data columns (total 89 columns):
#   Column              Non-Null Count  Dtype
---  -
0   Unnamed: 0          18207 non-null  int64
1   ID                  18207 non-null  int64
2   Name               18207 non-null  object
3   Age                18207 non-null  int64
4   Photo              18207 non-null  object
5   Nationality        18207 non-null  object
6   Flag               18207 non-null  object
7   Overall            18207 non-null  int64
8   Potential          18207 non-null  int64
9   Club               17966 non-null  object
10  Club Logo          18207 non-null  object
```

11	Value	18207	non-null	object
12	Wage	18207	non-null	object
13	Special	18207	non-null	int64
14	Preferred Foot	18159	non-null	object
15	International Reputation	18159	non-null	float64
16	Weak Foot	18159	non-null	float64
17	Skill Moves	18159	non-null	float64
18	Work Rate	18159	non-null	object
19	Body Type	18159	non-null	object
20	Real Face	18159	non-null	object
21	Position	18147	non-null	object
22	Jersey Number	18147	non-null	float64
23	Joined	16654	non-null	object
24	Loaned From	1264	non-null	object
25	Contract Valid Until	17918	non-null	object
26	Height	18159	non-null	object
27	Weight	18159	non-null	object
28	LS	16122	non-null	object
29	ST	16122	non-null	object
30	RS	16122	non-null	object
31	LW	16122	non-null	object
32	LF	16122	non-null	object
33	CF	16122	non-null	object
34	RF	16122	non-null	object
35	RW	16122	non-null	object
36	LAM	16122	non-null	object
37	CAM	16122	non-null	object
38	RAM	16122	non-null	object
39	LM	16122	non-null	object
40	LCM	16122	non-null	object
41	CM	16122	non-null	object
42	RCM	16122	non-null	object
43	RM	16122	non-null	object
44	LWB	16122	non-null	object
45	LDM	16122	non-null	object
46	CDM	16122	non-null	object
47	RDM	16122	non-null	object
48	RWB	16122	non-null	object
49	LB	16122	non-null	object
50	LCB	16122	non-null	object
51	CB	16122	non-null	object
52	RCB	16122	non-null	object
53	RB	16122	non-null	object
54	Crossing	18159	non-null	float64
55	Finishing	18159	non-null	float64
56	HeadingAccuracy	18159	non-null	float64
57	ShortPassing	18159	non-null	float64
58	Volleys	18159	non-null	float64

59	Dribbling	18159	non-null	float64
60	Curve	18159	non-null	float64
61	FKAccuracy	18159	non-null	float64
62	LongPassing	18159	non-null	float64
63	BallControl	18159	non-null	float64
64	Acceleration	18159	non-null	float64
65	SprintSpeed	18159	non-null	float64
66	Agility	18159	non-null	float64
67	Reactions	18159	non-null	float64
68	Balance	18159	non-null	float64
69	ShotPower	18159	non-null	float64
70	Jumping	18159	non-null	float64
71	Stamina	18159	non-null	float64
72	Strength	18159	non-null	float64
73	LongShots	18159	non-null	float64
74	Aggression	18159	non-null	float64
75	Interceptions	18159	non-null	float64
76	Positioning	18159	non-null	float64
77	Vision	18159	non-null	float64
78	Penalties	18159	non-null	float64
79	Composure	18159	non-null	float64
80	Marking	18159	non-null	float64
81	StandingTackle	18159	non-null	float64
82	SlidingTackle	18159	non-null	float64
83	GKDividing	18159	non-null	float64
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

```
[7]: # checking null values
df.isnull().sum()
```

```
[7]: Unnamed: 0      0
ID              0
Name            0
Age             0
Photo           0
...
GKHandling      48
GKKicking       48
GKPositioning   48
GKReflexes      48
Release Clause  1564
```

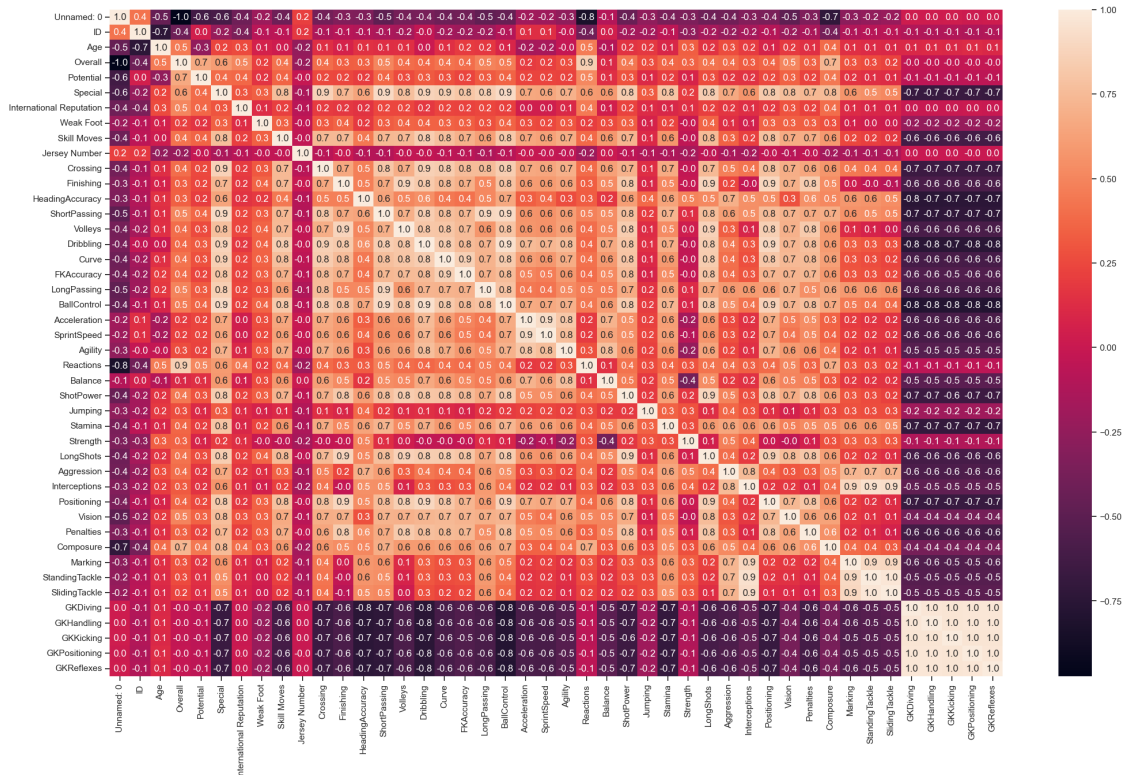
Length: 89, dtype: int64

```
[8]: # listing the columns
df.columns
```

```
[8]: 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', 'GKDividing', 'GKHandling',
        'GK Kicking', 'GK Positioning', 'GK Reflexes', 'Release Clause'],
        dtype='object')
```

```
[9]: # Plotting the Heatmap of the columns using correlation matrix
f, ax = plt.subplots(figsize=(25,15))
sns.heatmap(df.corr(),annot=True, fmt='.1f', ax=ax)
```

```
[9]: <Axes: >
```



```
[10]: # Nationality Text Size = Nationality Player Count
# Plotting the wordcloud for the Nationalit column
plt.subplots(figsize=(10,8))
wordcloud = WordCloud(
    background_color = 'black',
    width=1920,
    height=1080
).generate(" ".join(df.Nationality))

plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```



```

        'Stamina', 'Strength', 'LongShots', 'Aggression',
↪ 'Interceptions',
        'Positioning', 'Vision', 'Penalties', 'Composure',
↪ 'Marking',
        'StandingTackle', 'SlidingTackle', 'GKDividing',
↪ 'GKHandling',
        'GKkicking', 'GKPositioning', 'GKReflexes']]

```

```

[15]: # replacing the missing values with mean
for i in to_impute_by_mean.columns:
    df[i].fillna(df[i].mean(), inplace = True)

```

```

[16]: # These are categorical variables and will be imputed by mode.
to_impute_by_mode = df.loc[:, ['Body Type', 'International Reputation',
↪ 'Height', 'Weight', 'Preferred Foot', 'Jersey Number']]
for i in to_impute_by_mode.columns:
    df[i].fillna(df[i].mode(), inplace=True)

```

```

[17]: # the following variables are either discrete numerical or
# continuous numerical variables. So they will be imputed by median
to_impute_by_median = df.loc[:, ['Weak Foot', 'Skill Moves']]
for i in to_impute_by_median.columns:
    df[i].fillna(df[i].median(), inplace = True)

```

```

[18]: df.columns[df.isna().any()]

```

```

[18]: Index(['Preferred Foot', 'International Reputation', 'Work Rate', 'Body Type',
'Real Face', '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',
'Release Clause'],
dtype='object')

```

```

[19]: # Filling the remaining missing values with zero
df.fillna(0, inplace = True)

```

```

[20]: # functions to get the rounded values from different columns
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()))

```

```
[21]: # renaming columns
```

```
df.rename(columns={'Club Logo': 'Club_Logo'}, inplace=True)
```

```
[22]: df.columns
```

```

[22]: 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', 'GKDividing', 'GKHandling',
            'GKKicking', 'GKPositioning', 'GKReflexes', 'Release Clause'],
          dtype='object')

```

```
[23]: # adding these categories to the data

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

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

```
[24]:
```

	Unnamed: 0	ID	Name	Age	\
0	0	158023	L. Messi	31	
1	1	20801	Cristiano Ronaldo	33	
2	2	190871	Neymar Jr	26	
3	3	193080	De Gea	27	
4	4	192985	K. De Bruyne	27	

	Photo	Nationality	\
0	<a href="https://cdn.sofifa.org/players/4/19/158023.png">https://cdn.sofifa.org/players/4/19/158023.png</a>	Argentina	
1	<a href="https://cdn.sofifa.org/players/4/19/20801.png">https://cdn.sofifa.org/players/4/19/20801.png</a>	Portugal	
2	<a href="https://cdn.sofifa.org/players/4/19/190871.png">https://cdn.sofifa.org/players/4/19/190871.png</a>	Brazil	
3	<a href="https://cdn.sofifa.org/players/4/19/193080.png">https://cdn.sofifa.org/players/4/19/193080.png</a>	Spain	
4	<a href="https://cdn.sofifa.org/players/4/19/192985.png">https://cdn.sofifa.org/players/4/19/192985.png</a>	Belgium	

	Flag	Overall	Potential	\
0	<a href="https://cdn.sofifa.org/flags/52.png">https://cdn.sofifa.org/flags/52.png</a>	94	94	
1	<a href="https://cdn.sofifa.org/flags/38.png">https://cdn.sofifa.org/flags/38.png</a>	94	94	
2	<a href="https://cdn.sofifa.org/flags/54.png">https://cdn.sofifa.org/flags/54.png</a>	92	93	
3	<a href="https://cdn.sofifa.org/flags/45.png">https://cdn.sofifa.org/flags/45.png</a>	91	93	
4	<a href="https://cdn.sofifa.org/flags/7.png">https://cdn.sofifa.org/flags/7.png</a>	91	92	

	Club	...	GKReflexes	Release	Clause	Defending	General	\
0	FC Barcelona	...	8.0		€226.5M	29	89	
1	Juventus	...	11.0		€127.1M	27	88	
2	Paris Saint-Germain	...	11.0		€228.1M	28	85	
3	Manchester United	...	94.0		€138.6M	16	26	
4	Manchester City	...	13.0		€196.4M	59	79	

	Mental	Passing	Mobility	Power	Rating	Shooting
0	71	87	91	74	94	88
1	73	81	91	83	94	88
2	72	80	94	69	92	84
3	43	39	66	54	92	21
4	81	92	81	76	92	85

[5 rows x 97 columns]

```
[25]: # creating the players dataset
players = df[['Name', 'Defending', 'General', 'Mental', 'Passing',
              'Mobility', 'Power', 'Rating', 'Shooting', 'Flag', 'Age',
              'Nationality', 'Photo', 'Club_Logo', 'Club']]

players.head()
```

```
[25]:
```

	Name	Defending	General	Mental	Passing	Mobility	Power	\
0	L. Messi	29	89	71	87	91	74	
1	Cristiano Ronaldo	27	88	73	81	91	83	
2	Neymar Jr	28	85	72	80	94	69	
3	De Gea	16	26	43	39	66	54	
4	K. De Bruyne	59	79	81	92	81	76	

	Rating	Shooting	Flag	Age	Nationality	\
0	94	88	<a href="https://cdn.sofifa.org/flags/52.png">https://cdn.sofifa.org/flags/52.png</a>	31	Argentina	
1	94	88	<a href="https://cdn.sofifa.org/flags/38.png">https://cdn.sofifa.org/flags/38.png</a>	33	Portugal	
2	92	84	<a href="https://cdn.sofifa.org/flags/54.png">https://cdn.sofifa.org/flags/54.png</a>	26	Brazil	
3	92	21	<a href="https://cdn.sofifa.org/flags/45.png">https://cdn.sofifa.org/flags/45.png</a>	27	Spain	
4	92	85	<a href="https://cdn.sofifa.org/flags/7.png">https://cdn.sofifa.org/flags/7.png</a>	27	Belgium	

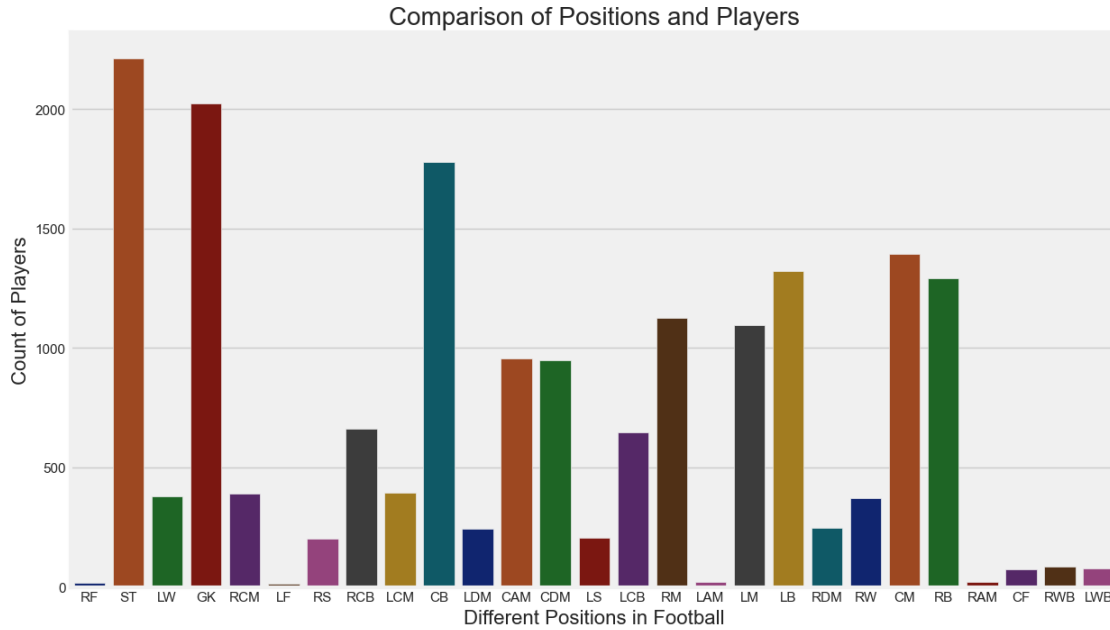
  

	Photo	\
0	<a href="https://cdn.sofifa.org/players/4/19/158023.png">https://cdn.sofifa.org/players/4/19/158023.png</a>	
1	<a href="https://cdn.sofifa.org/players/4/19/20801.png">https://cdn.sofifa.org/players/4/19/20801.png</a>	
2	<a href="https://cdn.sofifa.org/players/4/19/190871.png">https://cdn.sofifa.org/players/4/19/190871.png</a>	
3	<a href="https://cdn.sofifa.org/players/4/19/193080.png">https://cdn.sofifa.org/players/4/19/193080.png</a>	
4	<a href="https://cdn.sofifa.org/players/4/19/192985.png">https://cdn.sofifa.org/players/4/19/192985.png</a>	

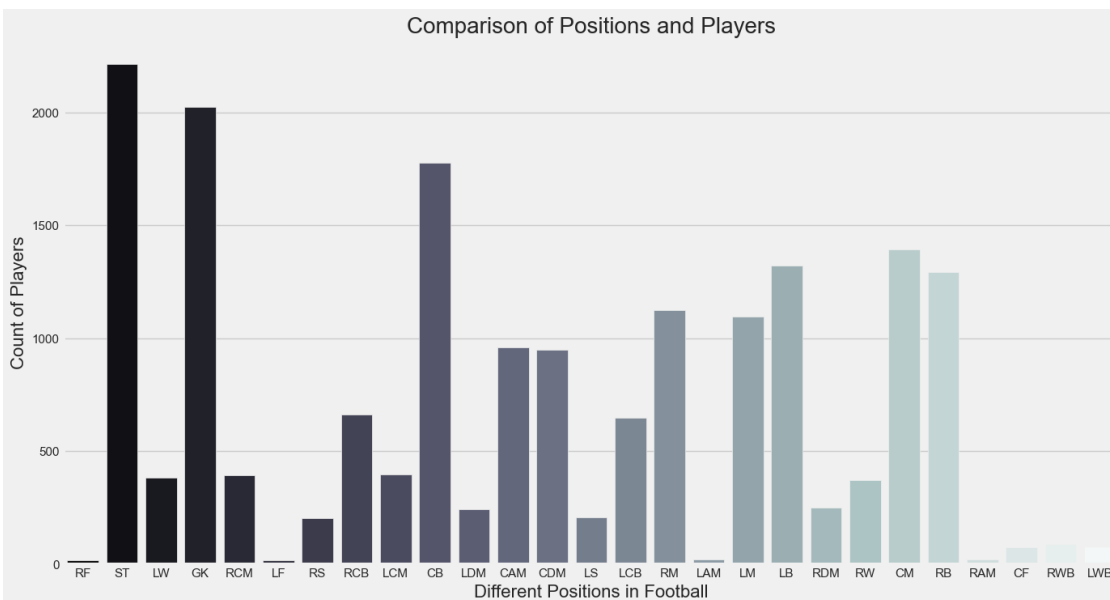
  

	Club_Logo	Club
0	<a href="https://cdn.sofifa.org/teams/2/light/241.png">https://cdn.sofifa.org/teams/2/light/241.png</a>	FC Barcelona
1	<a href="https://cdn.sofifa.org/teams/2/light/45.png">https://cdn.sofifa.org/teams/2/light/45.png</a>	Juventus
2	<a href="https://cdn.sofifa.org/teams/2/light/73.png">https://cdn.sofifa.org/teams/2/light/73.png</a>	Paris Saint-Germain
3	<a href="https://cdn.sofifa.org/teams/2/light/11.png">https://cdn.sofifa.org/teams/2/light/11.png</a>	Manchester United
4	<a href="https://cdn.sofifa.org/teams/2/light/10.png">https://cdn.sofifa.org/teams/2/light/10.png</a>	Manchester City

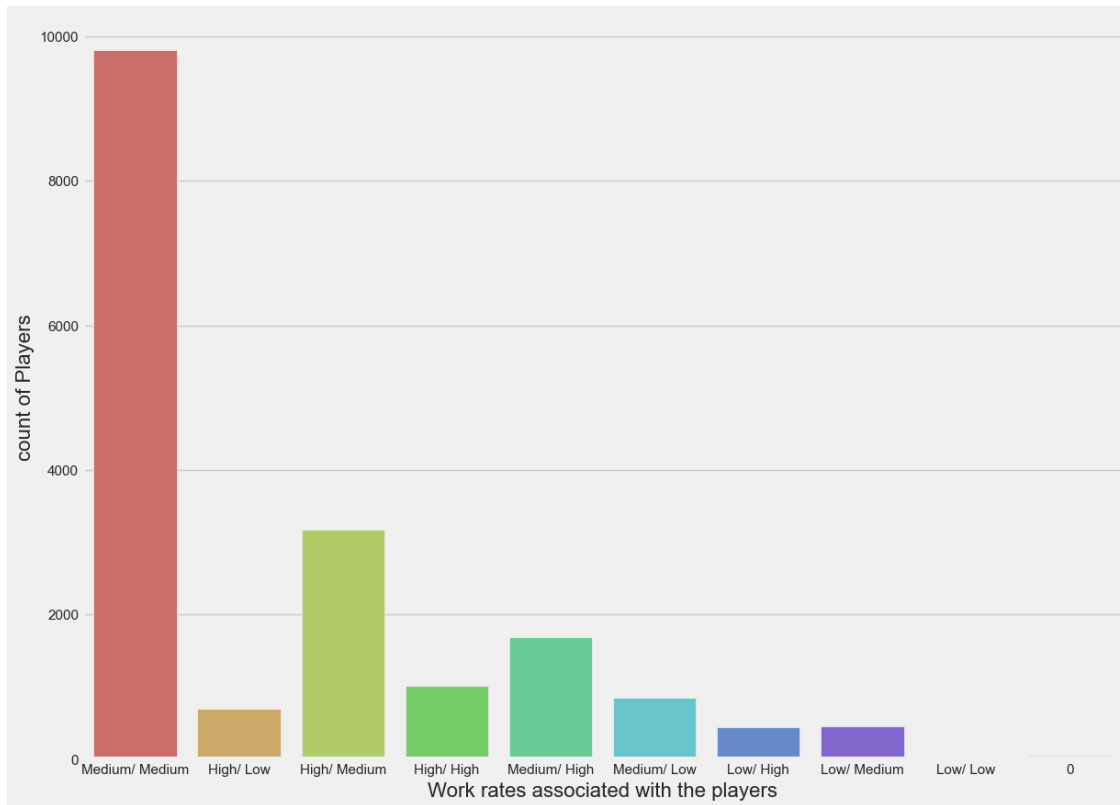
```
[26]: # different positions acquired by the players
plt.figure(figsize=(15,8))
plt.style.use('fivethirtyeight')
ax = sns.countplot(x= 'Position', data=df, palette='dark')
ax.set_xlabel(xlabel = 'Different Positions in Football', fontsize = 16)
ax.set_ylabel(ylabel = 'Count of Players', fontsize = 16)
ax.set_title(label = 'Comparison of Positions and Players', fontsize = 20)
plt.show()
```



```
[27]: # plotting count of players based on their heights
plt.figure(figsize=(15,8))
ax = sns.countplot(x= 'Position',data=df,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()
```

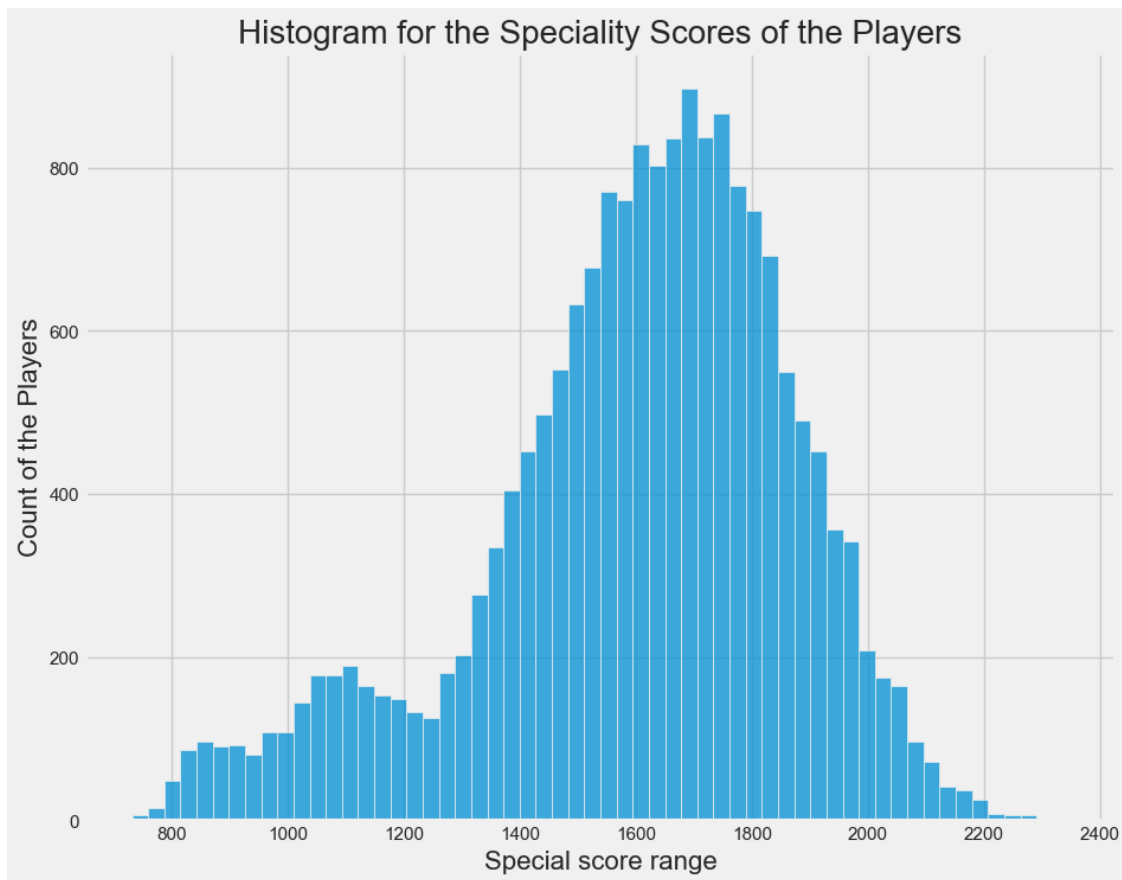


```
[28]: plt.figure(figsize=(13,10))
sns.countplot(x=df['Work Rate'],palette='hls')
plt.xlabel('Work rates associated with the players', fontsize = 16)
plt.ylabel('count of Players', fontsize = 16)
plt.show()
```



```
[29]: x = df.Special
plt.figure(figsize=(10,8))

ax = sns.histplot(x=x, bins=58)
ax.set_xlabel(xlabel = 'Special score range', fontsize = 16)
ax.set_ylabel(ylabel = 'Count of the Players',fontsize = 16)
ax.set_title(label = 'Histogram for the Speciality Scores of the Players',
             ↪ fontsize = 20)
plt.show()
```



```
[30]: # Every Nations' Player and their overall scores
# defining a tuple consisting of country names
some_countries = ('England', 'Germany', 'Spain', 'Argentina', 'France',
                  'Brazil', 'Italy', 'Columbia')

# extracting the overall data of the countries selected in the line above
data_country = df.loc[df['Nationality'].isin(some_countries) & df['Overall']]

data_country.head()
```

```
[30]: Unnamed: 0      ID      Name  Age  \
3          3  193080    De Gea   27
8          8  155862 Sergio Ramos  32
14         14  215914   N. Kanté   27
15         15  211110   P. Dybala   24
16         16  202126    H. Kane   24

                                     Photo Nationality  \
3  https://cdn.sofifa.org/players/4/19/193080.png      Spain
8  https://cdn.sofifa.org/players/4/19/155862.png      Spain
```

14	<a href="https://cdn.sofifa.org/players/4/19/215914.png">https://cdn.sofifa.org/players/4/19/215914.png</a>	France
15	<a href="https://cdn.sofifa.org/players/4/19/211110.png">https://cdn.sofifa.org/players/4/19/211110.png</a>	Argentina
16	<a href="https://cdn.sofifa.org/players/4/19/202126.png">https://cdn.sofifa.org/players/4/19/202126.png</a>	England

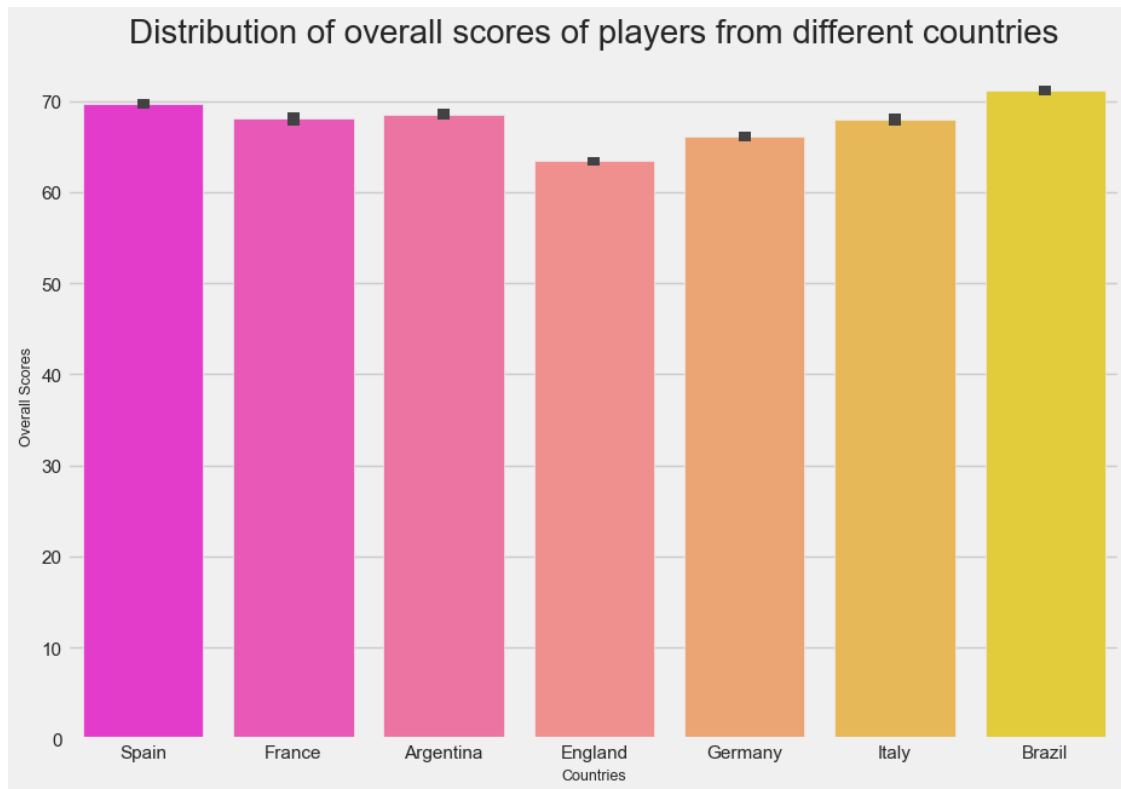
	Flag	Overall	Potential	\
3	<a href="https://cdn.sofifa.org/flags/45.png">https://cdn.sofifa.org/flags/45.png</a>	91	93	
8	<a href="https://cdn.sofifa.org/flags/45.png">https://cdn.sofifa.org/flags/45.png</a>	91	91	
14	<a href="https://cdn.sofifa.org/flags/18.png">https://cdn.sofifa.org/flags/18.png</a>	89	90	
15	<a href="https://cdn.sofifa.org/flags/52.png">https://cdn.sofifa.org/flags/52.png</a>	89	94	
16	<a href="https://cdn.sofifa.org/flags/14.png">https://cdn.sofifa.org/flags/14.png</a>	89	91	

	Club	...	GKReflexes	Release	Clause	Defending	General	\
3	Manchester United	...	94.0	€138.6M	16	26		
8	Real Madrid	...	11.0	€104.6M	90	78		
14	Chelsea	...	10.0	€121.3M	89	66		
15	Juventus	...	8.0	€153.5M	21	85		
16	Tottenham Hotspur	...	11.0	€160.7M	43	82		

	Mental	Passing	Mobility	Power	Rating	Shooting
3	43	39	66	54	92	21
8	77	74	78	82	91	68
14	83	78	84	85	90	61
15	67	81	87	76	92	86
16	75	79	76	80	90	85

[5 rows x 97 columns]

```
[31]: plt.rcParams['figure.figsize'] = (10,7)
ax= sns.
    ↳ barplot(x=data_country['Nationality'],y=data_country['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 = 20)
plt.show()
```



```
[32]: # finding the number of players in each club
df['Club'].value_counts().head(10)
```

```
[32]: No Club                241
      FC Barcelona           33
      Burnley                33
      AS Monaco              33
      Everton                33
      TSG 1899 Hoffenheim    33
      Wolverhampton Wanderers 33
      Eintracht Frankfurt    33
      Southampton            33
      Valencia CF            33
      Name: Club, dtype: int64
```

```
[33]: # copying the dataset
data = df.copy()
plt.style.use('fivethirtyeight')
sns.set(style="ticks")
```

```
[34]: # creating a tuple of club names
```



```

some_clubs = ('CD Leganés', 'Southampton', 'RC Celta', 'Empoli', 'Fortuna_
↳Düsseldorf', 'Manchester City',
              'Tottenham Hotspur', 'FC Barcelona', 'Valencia CF', 'Chelsea',_
↳'Real Madrid')

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

```

```

[34]:      Unnamed: 0      ID      Name  Age  \
5          5  183277    E. Hazard   27
6          6  177003    L. Modrić   32
7          7  176580    L. Suárez   31
8          8  155862  Sergio Ramos   32
14         14  215914     N. Kanté   27

      Photo Nationality  \
5  https://cdn.sofifa.org/players/4/19/183277.png    Belgium
6  https://cdn.sofifa.org/players/4/19/177003.png    Croatia
7  https://cdn.sofifa.org/players/4/19/176580.png    Uruguay
8  https://cdn.sofifa.org/players/4/19/155862.png      Spain
14 https://cdn.sofifa.org/players/4/19/215914.png    France

      Flag  Overall  Potential      Club  \
5  https://cdn.sofifa.org/flags/7.png      91      91    Chelsea
6  https://cdn.sofifa.org/flags/10.png     91      91  Real Madrid
7  https://cdn.sofifa.org/flags/60.png     91      91  FC Barcelona
8  https://cdn.sofifa.org/flags/45.png     91      91  Real Madrid
14 https://cdn.sofifa.org/flags/18.png     89      90    Chelsea

      ... GKReflexes Release Clause Defending  General Mental  Passing  \
5  ...      8.0      €172.1M      28      83      72      84
6  ...      9.0      €137.4M      70      81      80      89
7  ...     37.0      €164M      48      85      78      74
8  ...     11.0     €104.6M      90      78      77      74
14 ...     10.0     €121.3M      89      66      83      78

      Mobility  Power Rating Shooting
5          92      75      91      82
6          84      77      91      78
7          84      81      91      87
8          78      82      91      68
14         84      85      90      61

[5 rows x 97 columns]

```

```

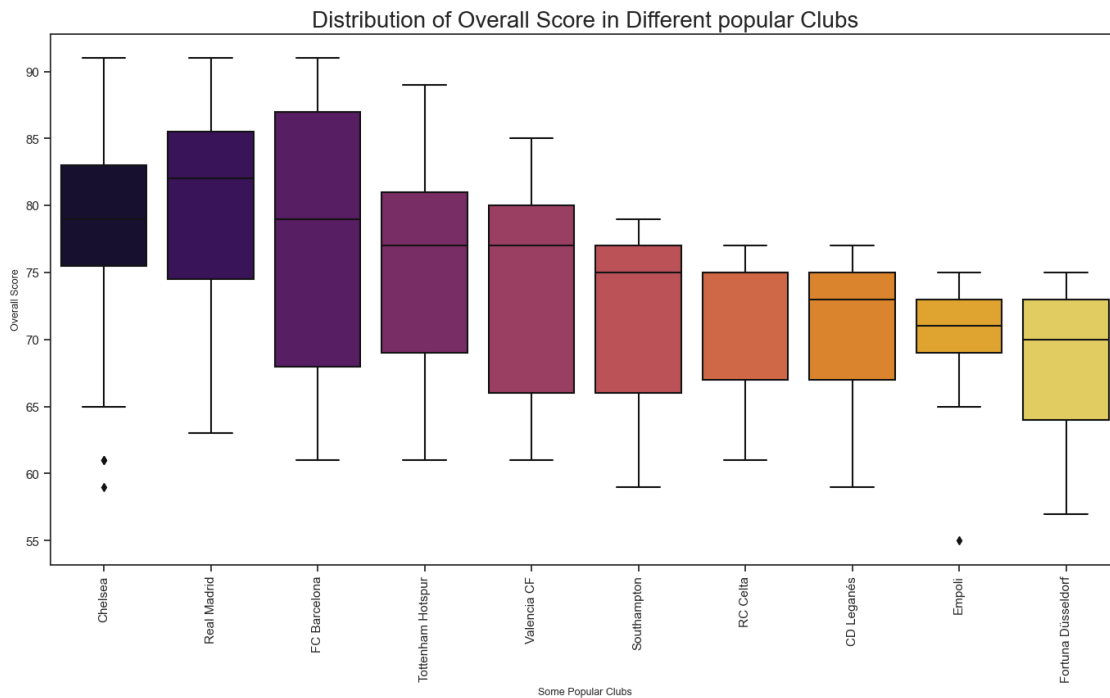
[35]: plt.rcParams['figure.figsize'] = (15,8)
ax = sns.boxplot(x=data_clubs['Club'],y=data_clubs['Overall'],palette='inferno')

```

```

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.show()

```



```

[36]: # finding out the top 10 left footed footballers
left = data[data['Preferred Foot']=='Left'][['Name', 'Age', 'Club', '
↳Nationality']].head(10)
left

```

```

[36]:
      Name  Age      Club Nationality
0    L. Messi  31   FC Barcelona  Argentina
13  David Silva  32 Manchester City    Spain
15    P. Dybala  24      Juventus  Argentina
17  A. Griezmann  27 Atlético Madrid    France
19    T. Courtois  26     Real Madrid    Belgium
24    G. Chiellini  33      Juventus    Italy
26    M. Salah  26     Liverpool    Egypt
28  J. Rodríguez  26 FC Bayern München  Colombia
35    Marcelo  30     Real Madrid    Brazil
36    G. Bale  28     Real Madrid    Wales

```

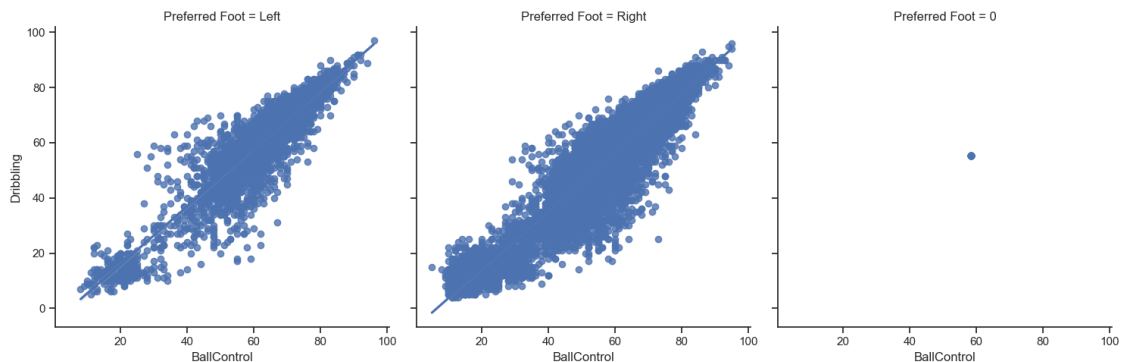
```
[37]: right = data[data['Preferred Foot']=='Right'][['Name', 'Age', 'Club', 'Nationality']].head(10)
right
```

```
[37]:
```

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

```
[38]: # 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()
```



```
[39]: # checking the clubs where players from the most number of nations play
data.groupby(data['Club'])['Nationality'].nunique().
sort_values(ascending=False).head(10)
```

```
[39]: Club
```

No Club	28
Brighton & Hove Albion	21
Fulham	19
Udinese	18
West Ham United	18
Empoli	18
AS Monaco	18

```
Eintracht Frankfurt      18
Lazio                    18
Napoli                   18
Name: Nationality, dtype: int64
```

```
[40]: # checking the clubs where players from the least number of nations play
data.groupby(data['Club'])['Nationality'].nunique().sort_values(ascending =
↳ True).head(10)
```

```
[40]: Club
Santos      1
Ceará Sporting Club  1
América FC (Minas Gerais)  1
Paraná      1
Chapecoense  1
Padova      1
Cittadella  1
Sangju Sangmu FC  1
Ranheim Fotball  1
CA Osasuna   1
Name: Nationality, dtype: int64
```

```
[41]: # dropping the unnamed column
df.drop(['Unnamed: 0'],axis=1,inplace=True)
```

```
[42]: #Player with maximum Potential and Overall Performance
player = str(df.loc[df['Potential'].idxmax()][1])
print('Maximum Potential : '+str(df.loc[df['Potential'].idxmax()][1]))
print('Maximum Overall Performamnce : '+str(df.loc[df['Overall'].idxmax()][1]))
```

```
Maximum Potential : K. Mbappé
Maximum Overall Performamnce : L. Messi
```

```
[43]: # finding the best players for each performance criteria

pr_cols=['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', 'GKDividing',
'GKHandling', 'GKKicking', 'GKPositioning', 'GKReflexes']

i=0
while i < len(pr_cols):
    print('best {0}:{1}'.format(pr_cols[i],df.loc[df[pr_cols[i]].idxmax()][1]))
    i += 1
```

best Crossing:K. De Bruyne  
 best Finishing:L. Messi  
 best HeadingAccuracy:Naldo  
 best ShortPassing:L. Modrić  
 best Volleys:E. Cavani  
 best Dribbling:L. Messi  
 best Curve:Quaresma  
 best FKAccuracy:L. Messi  
 best LongPassing:T. Kroos  
 best BallControl:L. Messi  
 best Acceleration:Douglas Costa  
 best SprintSpeed:K. Mbappé  
 best Agility:Neymar Jr  
 best Reactions:Cristiano Ronaldo  
 best Balance:Bernard  
 best ShotPower:Cristiano Ronaldo  
 best Jumping:Cristiano Ronaldo  
 best Stamina:N. Kanté  
 best Strength:A. Akinfenwa  
 best LongShots:L. Messi  
 best Aggression:B. Pearson  
 best Interceptions:N. Kanté  
 best Positioning:Cristiano Ronaldo  
 best Vision:L. Messi  
 best Penalties:M. Balotelli  
 best Composure:L. Messi  
 best Marking:A. Barzagli  
 best StandingTackle:G. Chiellini  
 best SlidingTackle:Sergio Ramos  
 best GKDividing:De Gea  
 best GKHandling:J. Oblak  
 best GKKicking:M. Neuer  
 best GKPositioning:G. Buffon  
 best GKReflexes:De Gea

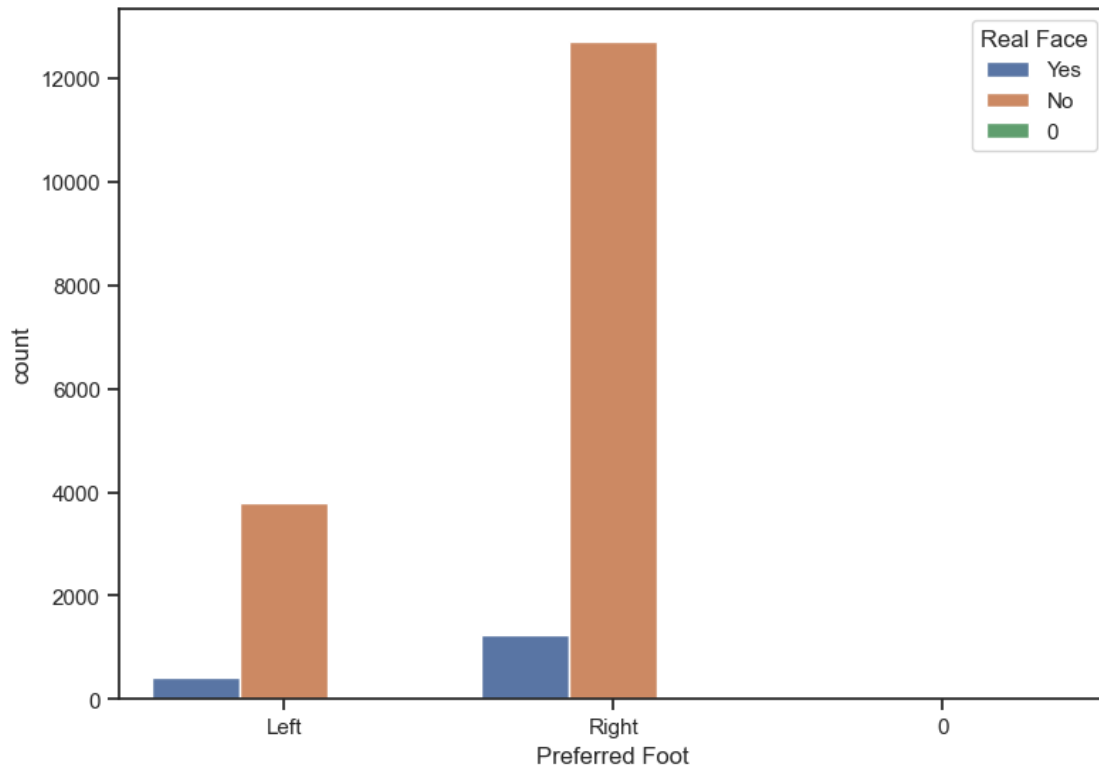
```

[44]: # creating a list of best players in each of the pr_cols criteria
i=0
best = []
while i < len(pr_cols):
    best.append(df.loc[df[pr_cols[i]].idxmax()][1])
    i +=1
print(best)
  
```

['K. De Bruyne', 'L. Messi', 'Naldo', 'L. Modrić', 'E. Cavani', 'L. Messi',  
 'Quaresma', 'L. Messi', 'T. Kroos', 'L. Messi', 'Douglas Costa', 'K. Mbappé',  
 'Neymar Jr', 'Cristiano Ronaldo', 'Bernard', 'Cristiano Ronaldo', 'Cristiano  
 Ronaldo', 'N. Kanté', 'A. Akinfenwa', 'L. Messi', 'B. Pearson', 'N. Kanté',  
 'Cristiano Ronaldo', 'L. Messi', 'M. Balotelli', 'L. Messi', 'A. Barzagli', 'G.

Chiellini', 'Sergio Ramos', 'De Gea', 'J. Oblak', 'M. Neuer', 'G. Buffon', 'De Gea']

```
[45]: # Plot to show the preferred foot choice of different players
f, ax = plt.subplots(figsize=(8, 6))
sns.countplot(x="Preferred Foot", hue="Real Face", data=df)
plt.show()
```



```
[46]: # Finding the player with the maximum potential
df.loc[df['Potential'].idxmax()][1]
```

[46]: 'K. Mbappé'

```
[47]: # showing the name of the players which occurs the most number of times from
      ↳ the first 20 names
plt.subplots(figsize=(10,8))
wordcloud = WordCloud(
    background_color='black',
    width=1920,
    height=1080
).generate(" ".join(df.Name[0:20]))
plt.imshow(wordcloud)
```

```
plt.axis('off')
plt.show()
```



```
[48]: # checking which clubs have been mentioned the most
plt.subplots(figsize=(10,8))
wordcloud = WordCloud(
    background_color='black',
    width=1920,
    height=1080
).generate(" ".join(df.Club))

plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```



```
[49]: # showing the name of the players which occurs the most number of times(left
      ↪join)
plt.subplots(figsize=(10,8))
wordcloud = WordCloud(
    background_color='black',
    width=1920,
    height=1080
).generate(" ".join(left.Name))

plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```





```
[50]: # showing the name of the players which occurs the most number of times(right_
      ↪join)
plt.subplots(figsize=(10,8))
wordcloud = WordCloud(
    background_color='black',
    width=1920,
    height=1080
).generate(" ".join(right.Name))
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```



```
[51]: # Checking which player has been mentioned the most in the 'best' list that we
      ↪ have prepared
plt.subplots(figsize=(10,8))
wordcloud = WordCloud(
    background_color='black',
    width=1920,
    height=1080
).generate(" ".join(best))
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```

