

Slack Task:- Matplotlib and Seaborn

02-05-2023 Tuesday

Exercise 1:

Create a line plot using matplotlib pyplot that displays the population of four different cities over time. Each city should have its own line, and the x-axis should represent years (e.g. 2010, 2011, 2012, etc.) while the y-axis should represent the population. The data for the four cities is provided below:

City A: [500000, 550000, 600000, 650000, 700000, 750000, 800000] City B: [800000, 850000, 900000, 950000, 1000000, 1050000, 1100000] City C: [1000000, 1050000, 1100000, 1150000, 1200000, 1250000, 1300000] City D: [1200000, 1250000, 1300000, 1350000, 1400000, 1450000, 1500000]

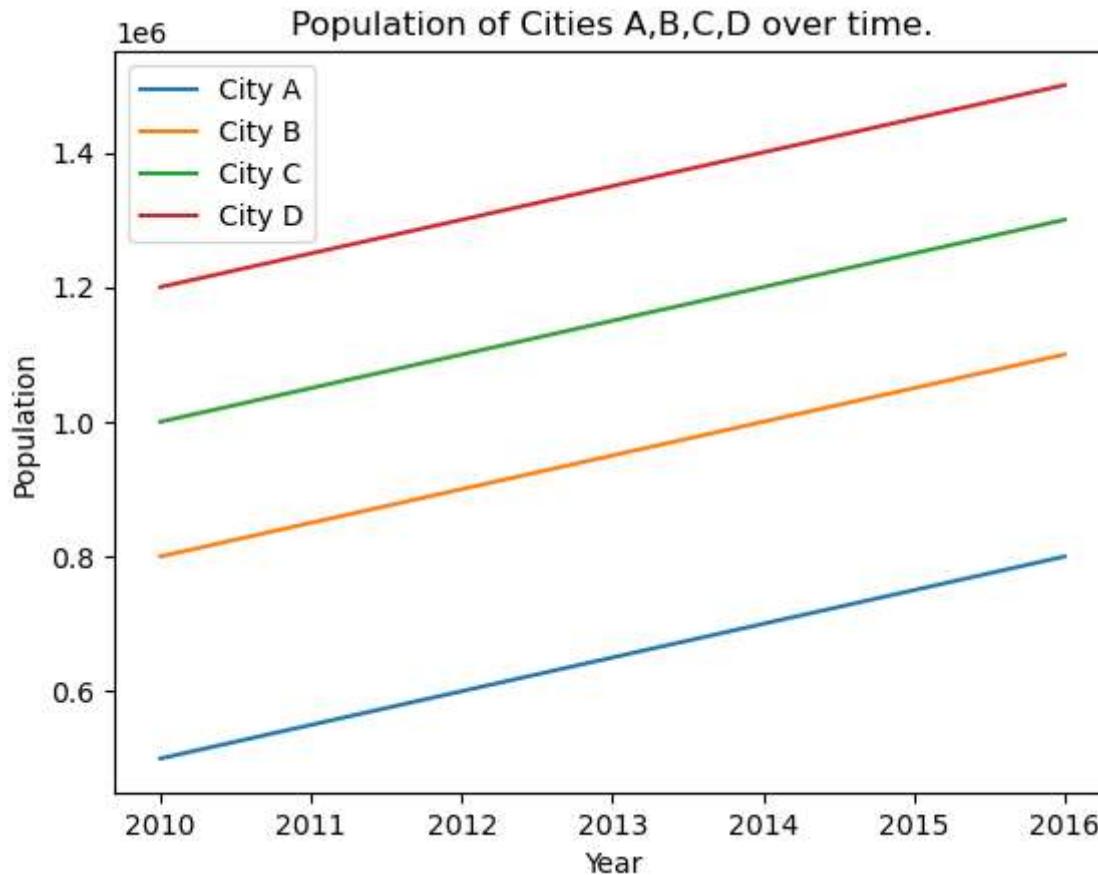
```
In [16]: import matplotlib.pyplot as plt

years=[2010,2011,2012,2013,2014,2015,2016]
city_a=[500000, 550000, 600000, 650000, 700000, 750000, 800000]
city_b=[800000, 850000, 900000, 950000, 1000000, 1050000, 1100000]
city_c=[1000000, 1050000, 1100000, 1150000, 1200000, 1250000, 1300000]
city_d=[1200000, 1250000, 1300000, 1350000, 1400000, 1450000, 1500000]

plt.plot(years,city_a,label='City A')
plt.plot(years,city_b,label='City B')
plt.plot(years,city_c,label='City C')
plt.plot(years,city_d,label='City D')

plt.xlabel('Year')
plt.ylabel('Population')
plt.title("Population of Cities A,B,C,D over time.")
plt.legend()

plt.show()
```



Exercise 2:

Create a scatter plot using matplotlib pyplot that shows the relationship between the number of hours studied and the test scores obtained by a group of students. Use the following data:

Hours Studied: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] Test Scores: [93, 57, 61, 54, 51, 53, 87, 81, 83, 85]

In [252]:

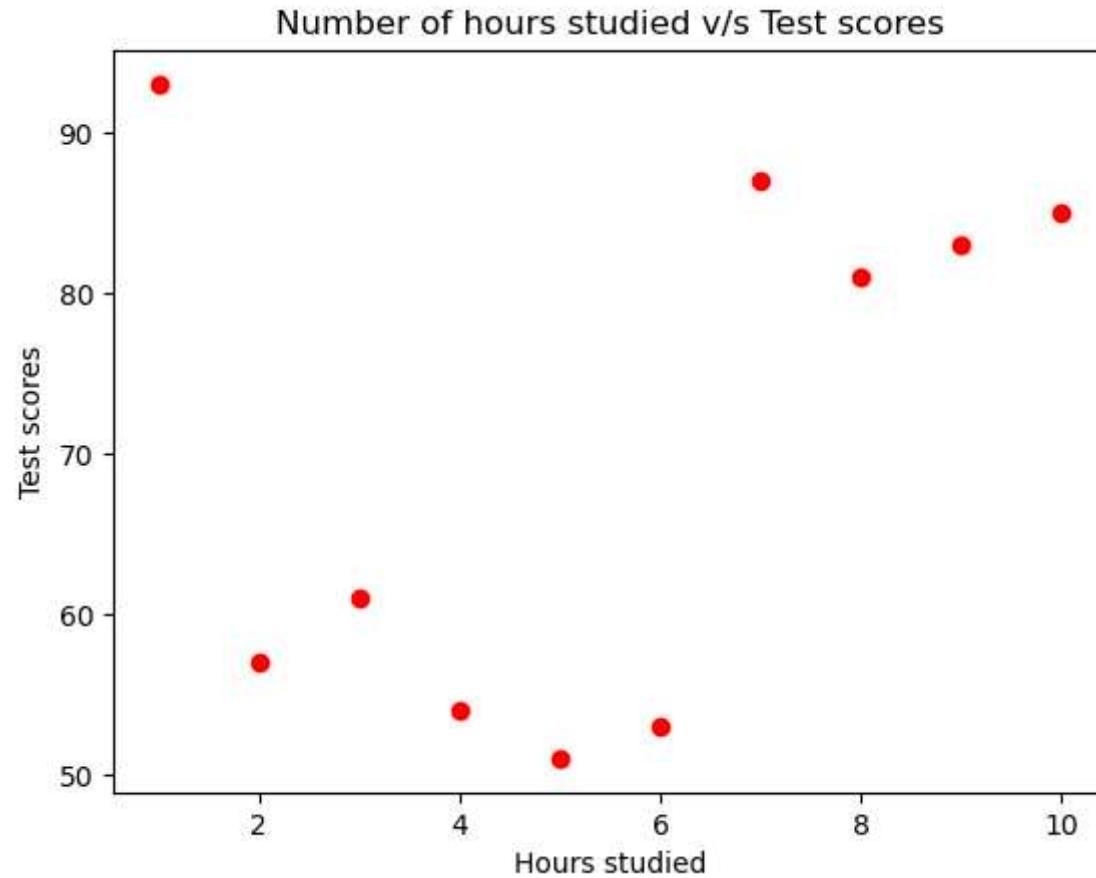
```
import matplotlib.pyplot as plt

hours=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
scores=[93, 57, 61, 54, 51, 53, 87, 81, 83, 85]

plt.scatter(hours,scores,color='r')
```

```
plt.xlabel('Hours studied')
plt.ylabel('Test scores')
plt.title('Number of hours studied v/s Test scores')

plt.show()
```



Exercise 3:

Create a bar chart using matplotlib pyplot that shows the total sales for each month of the year. Use the following data:

Month: ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"] Sales: [11860, 10480, 4997, 5523, 13965, 6011, 13158, 9533, 5158, 9058, 11346, 6675]

In [27]:

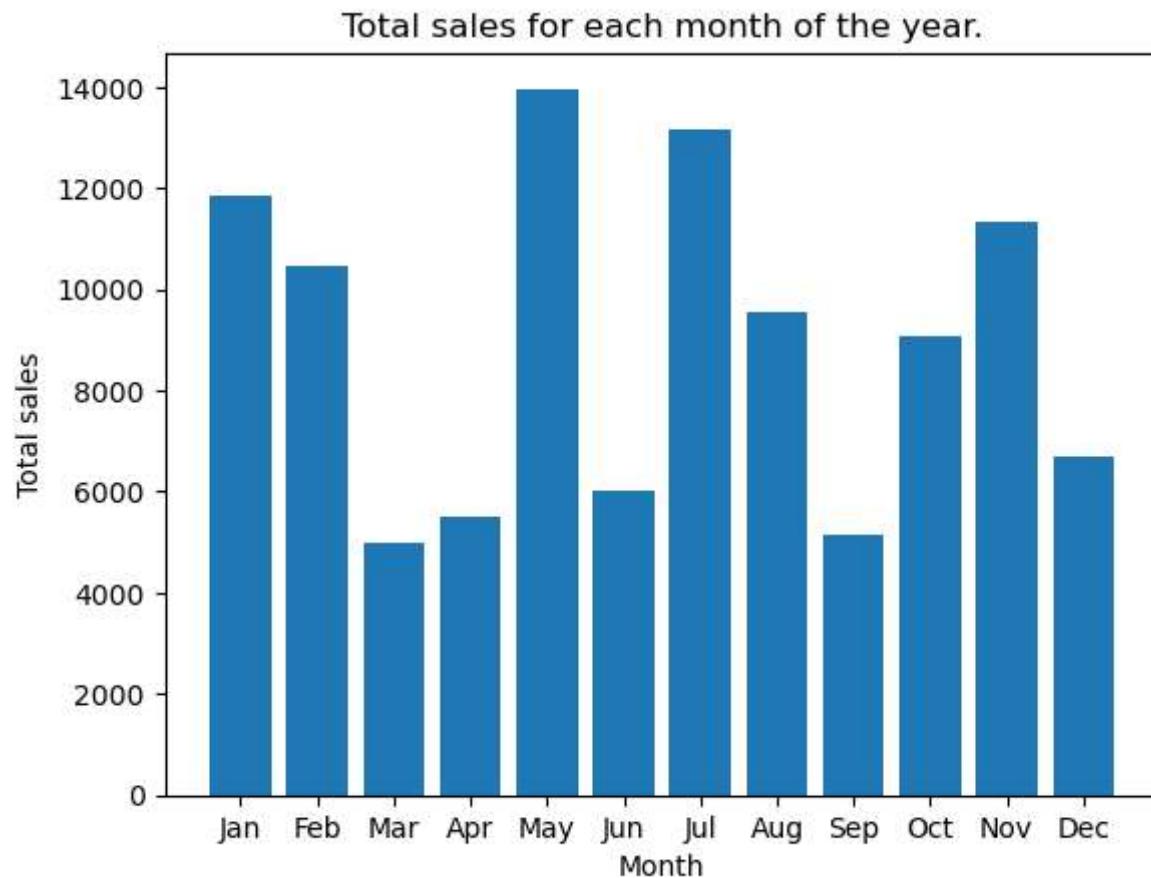
```
import matplotlib.pyplot as plt

month=["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"]
sales=[11860, 10480, 4997, 5523, 13965, 6011, 13158, 9533, 5158, 9058, 11346, 6675]

plt.bar(month,sales)

plt.xlabel('Month')
plt.ylabel('Total sales')
plt.title('Total sales for each month of the year.')

plt.show()
```



Exercise 4:

Using the below dataset of fifa players database, Perform Exploratory data analysis and find the following insights:

- 1.Which country has the most number of players
- 2.Plot a bar chart of 5 top countries with most number of players
- 3.Which player has the highest salary?
- 4.Plot a histogram to get the salary range of the players.
- 5.Who is the tallest player in the fifa players?
- 6.Which club has most number of players?
- 7.Which foot is most preferred by the players?Draw a bar chart for preferred foot

```
In [61]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df=pd.read_csv("C://Users//ACER//Documents//fifa_data.csv")
df.head()
```

Out[61]:

	Unnamed: 0	ID	Name	Age	Photo	Nationality	Flag	Overall	Potential
0	0	158023	L. Messi	31	https://cdn.sofifa.org/players/4/19/158023.png	Argentina	https://cdn.sofifa.org/flags/52.png	94	95
1	1	20801	Cristiano Ronaldo	33	https://cdn.sofifa.org/players/4/19/20801.png	Portugal	https://cdn.sofifa.org/flags/38.png	94	95
2	2	190871	Neymar Jr	26	https://cdn.sofifa.org/players/4/19/190871.png	Brazil	https://cdn.sofifa.org/flags/54.png	92	95
3	3	193080	De Gea	27	https://cdn.sofifa.org/players/4/19/193080.png	Spain	https://cdn.sofifa.org/flags/45.png	91	95
4	4	192985	K. De Bruyne	27	https://cdn.sofifa.org/players/4/19/192985.png	Belgium	https://cdn.sofifa.org/flags/7.png	91	95

5 rows × 89 columns

```
In [80]: 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	GKDiving	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
```

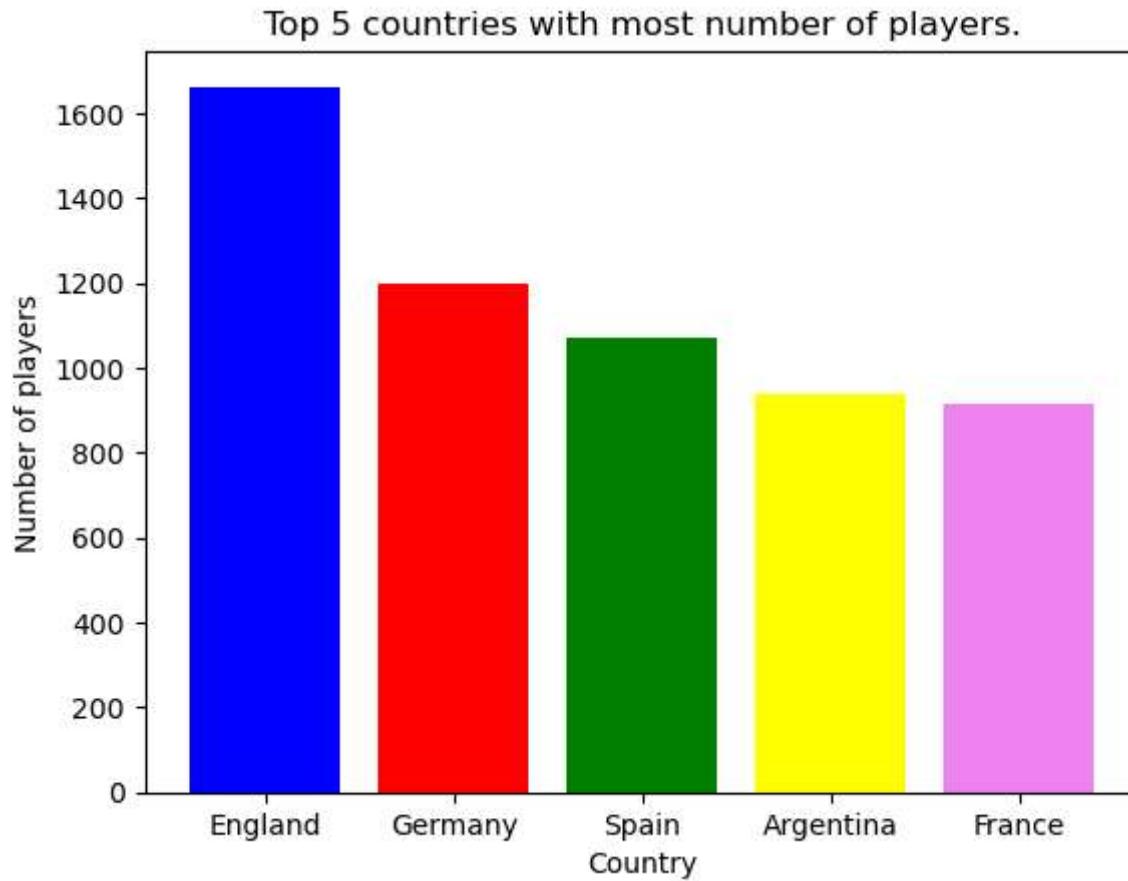
1.Which country has the most number of players?

```
In [68]: country=df[ 'Nationality' ].value_counts().head(1)
print(f'The country with maximum number of players is,\n {country}')
```

```
The country with maximum number of players is,
England ... 1662
Name: Nationality, dtype: int64
```

2.Plot a bar chart of 5 top countries with most number of players?

```
In [251]: top_countries=df[ 'Nationality' ].value_counts().head(5)
x=top_countries.index
y=top_countries.values
plt.bar(x,y,color=['blue','red','green','yellow','violet'])
plt.xlabel('Country')
plt.ylabel('Number of players')
plt.title('Top 5 countries with most number of players.')
plt.show()
```



3.Which player has the highest salary?

```
In [ ]: # Here the Wage column is in 'string' format, so we need to convert it into 'int' format,
def convert(wage):
    wage_int=wage.replace('€','').replace('K','000')
    return int(wage_int)
df['Wage']=df['Wage'].apply(convert)
```

```
In [129]: player=df.sort_values(by=[ 'Wage'],ascending=False).iloc[0]['Name']
salary=df.sort_values(by=[ 'Wage'],ascending=False).iloc[0]['Wage']

print(f"The player with highest salary is,\n Name: {player}\n Wage: € {salary}")
```

The player with highest salary is,

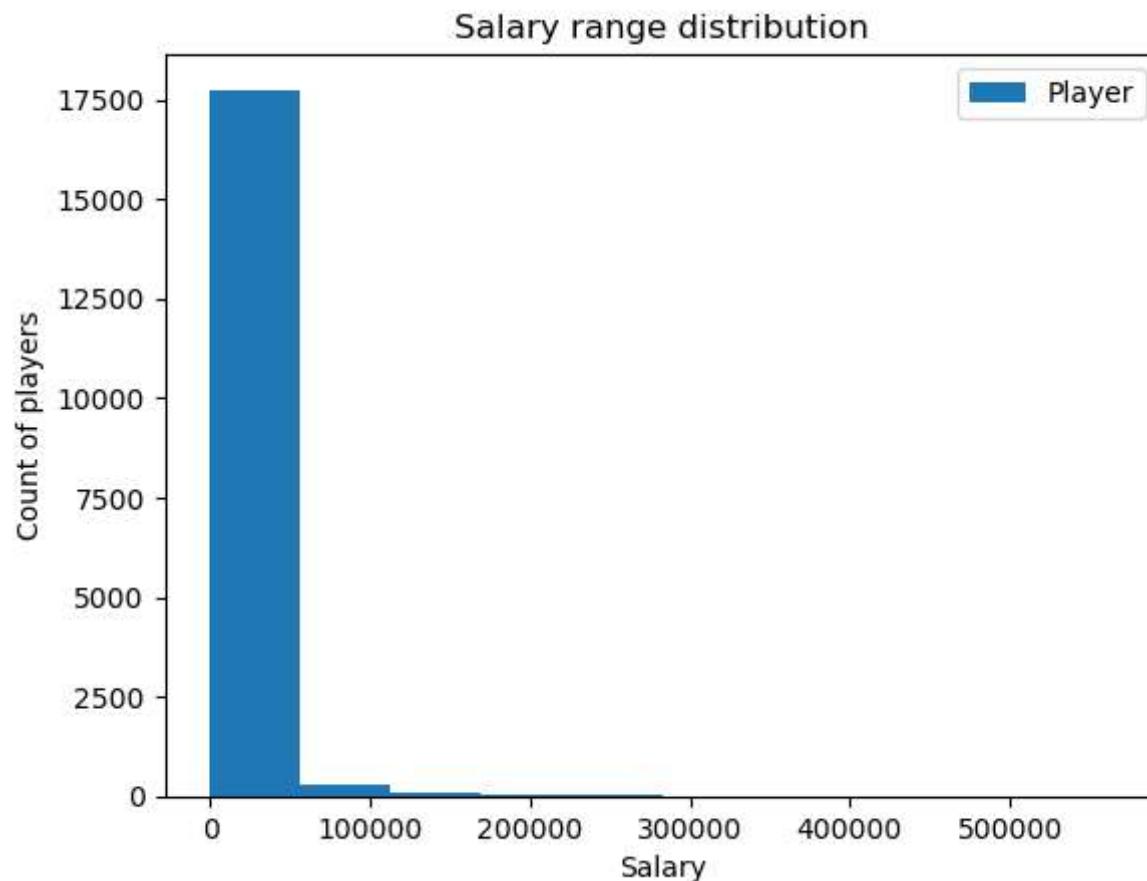
Name: L. Messi

Wage: € 565000

4. Plot a histogram to get the salary range of the players?

```
In [138]: plt.hist(df['Wage'],label='Player')
plt.xlabel('Salary')
plt.ylabel('Count of players')
plt.title('Salary range distribution')
plt.legend()

plt.show()
```



5. Who is the tallest player in the fifa players?

```
In [ ]: #here the height is in string format so convert into int also convert the height into centimeters for ease of calculation,
def height_convert(height):
    if isinstance(height, str):
        feet,inch= height.split('\'')
        total_inches=int(feet) * 12 + int(inch)
        cm=total_inches * 2.54
        return cm
    else:
        return height

df['Height']=df['Height'].apply(height_convert)
df['Height']
```

```
In [189]: name_ht=df.sort_values(by=[ 'Height'],ascending=False).iloc[0]['Name']
height_player=df.sort_values(by=[ 'Height'],ascending=False).iloc[0]['Height']
print(f"The tallest player in the fifa players is, \n Name: {name_ht} \n Height in cm: {height_player}")
```

The tallest player in the fifa players is,
Name: D. Hodzic
Height in cm: 205.74

6.Which club has most number of players?

```
In [218]: df['Club'].value_counts()
```



```
Out[218]: FC Barcelona      33
Valencia CF      33
Fortuna Düsseldorf 33
Cardiff City      33
Rayo Vallecano    33
...
América FC (Minas Gerais) 20
Fluminense       20
Sligo Rovers     19
Limerick FC       19
Derry City        18
Name: Club, Length: 651, dtype: int64
```

```
In [220]: club=df['Club'].value_counts()
club33=club[club==33]
num=len(club33)
num
```

```
Out[220]: 26
```

```
In [223]: l=df['Club'].value_counts().head(26)
club_list=list(l.index)
print('The following list of clubs having the maximum no: of players, \n',club_list)
```

The following list of clubs having the maximum no: of players,
['FC Barcelona', 'Valencia CF', 'Fortuna Düsseldorf', 'Cardiff City', 'Rayo Vallecano', 'CD Leganés', 'Frosinone', 'Newcastle United', 'Southampton', 'Burnley', 'Eintracht Frankfurt', 'Wolverhampton Wanderers', 'TSG 1899 Hoffenheim', 'Everton', 'AS Monaco', 'RC Celta', 'Empoli', 'Manchester City', 'Manchester United', 'Borussia Dortmund', 'Real Madrid', 'Atlético Madrid', 'Tottenham Hotspur', 'Chelsea', 'Liverpool', 'Arsenal']

7.Which foot is most preferred by the players?Draw a bar chart for preferred foot

```
In [231]: foot=df['Preferred Foot'].value_counts()
foot
```

```
Out[231]: Right .... 13948
Left ..... 4211
Name: Preferred Foot, dtype: int64
```

```
In [250]: plt.figure(figsize=(5,5),dpi=100)
x=foot.index
y=foot.values
plt.bar(x,y,color=['green','yellow'])
plt.xlabel('Preferred foot')
plt.ylabel('Number of players')
plt.title('Preferred foot of players')

plt.show()
```

