In [15]:

```
#import the libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
#import the dataset
data= pd.read_csv('euro2024_players.csv')
data.head()
```

Out[2]:

	Name	Position	Age	Club	Height	Foot	Caps	Goals	MarketValue	Country
0	Marc-André ter Stegen	Goalkeeper	32	FC Barcelona	187	right	40	0	28000000	Germany
1	Manuel Neuer	Goalkeeper	38	Bayern Munich	193	right	119	0	4000000	Germany
2	Oliver Baumann	Goalkeeper	34	TSG 1899 Hoffenheim	187	right	0	0	3000000	Germany
3	Nico Schlotterbeck	Centre-Back	24	Borussia Dortmund	191	left	12	0	40000000	Germany
4	Jonathan Tah	Centre-Back	28	Bayer 04 Leverkusen	195	right	25	0	30000000	Germany

In [3]:

```
#Checking the rows and columns in the dataset
data.shape
```

Out[3]:

```
(623, 10)
```

In [4]:

```
#Checking if there are any null values in the dataset
data.isnull().sum()
```

Out[4]:

```
Name           0
Position       0
Age            0
Club           0
Height         0
Foot           3
Caps           0
Goals          0
MarketValue    0
Country        0
dtype: int64
```

In [5]:

```
#info of the dataset to check the column names, data-types
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 623 entries, 0 to 622
Data columns (total 10 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   Name         623 non-null    object
 1   Position     623 non-null    object
 2   Age          623 non-null    int64 
 3   Club         623 non-null    object
 4   Height       623 non-null    int64 
 5   Foot         620 non-null    object
 6   Caps         623 non-null    int64 
 7   Goals        623 non-null    int64 
 8   MarketValue  623 non-null    int64 
 9   Country      623 non-null    object
dtypes: int64(5), object(5)
memory usage: 48.8+ KB
```

Exploratory Data Analysis:-

In [6]:

```
#Player with most goals
most_goals_index = data['Goals'].idxmax()
player_most_goals = data.loc[most_goals_index]

print("Player with the most goals:")
print(player_most_goals)
```

```
Player with the most goals:
Name           Cristiano Ronaldo
Position          Centre-Forward
Age                           39
Club                 Al-Nassr FC
Height                       188
Foot                       right
Caps                         206
Goals                        128
MarketValue             15000000
Country                 Portugal
Name: 596, dtype: object
```

In [7]:

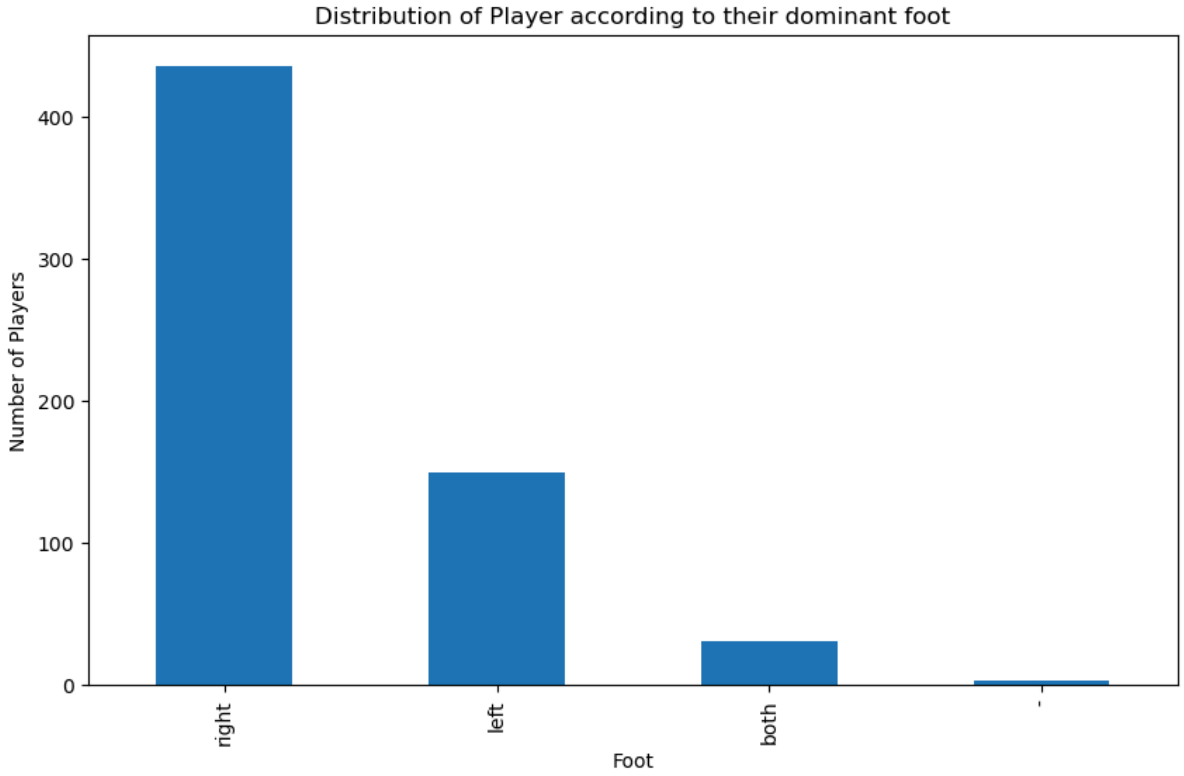
```
#Player with most caps
most_caps_index = data['Caps'].idxmax()
player_most_caps = data.loc[most_caps_index,['Name','Caps']]

print("Player with the most caps:")
print(player_most_caps)
```

```
Player with the most caps:
Name    Cristiano Ronaldo
Caps                  206
Name: 596, dtype: object
```

So, No surprises there, Cristiano Ronaldo is the most capped player in the Euros 2024 and also the top position in the number of goals scored.

In [8]:

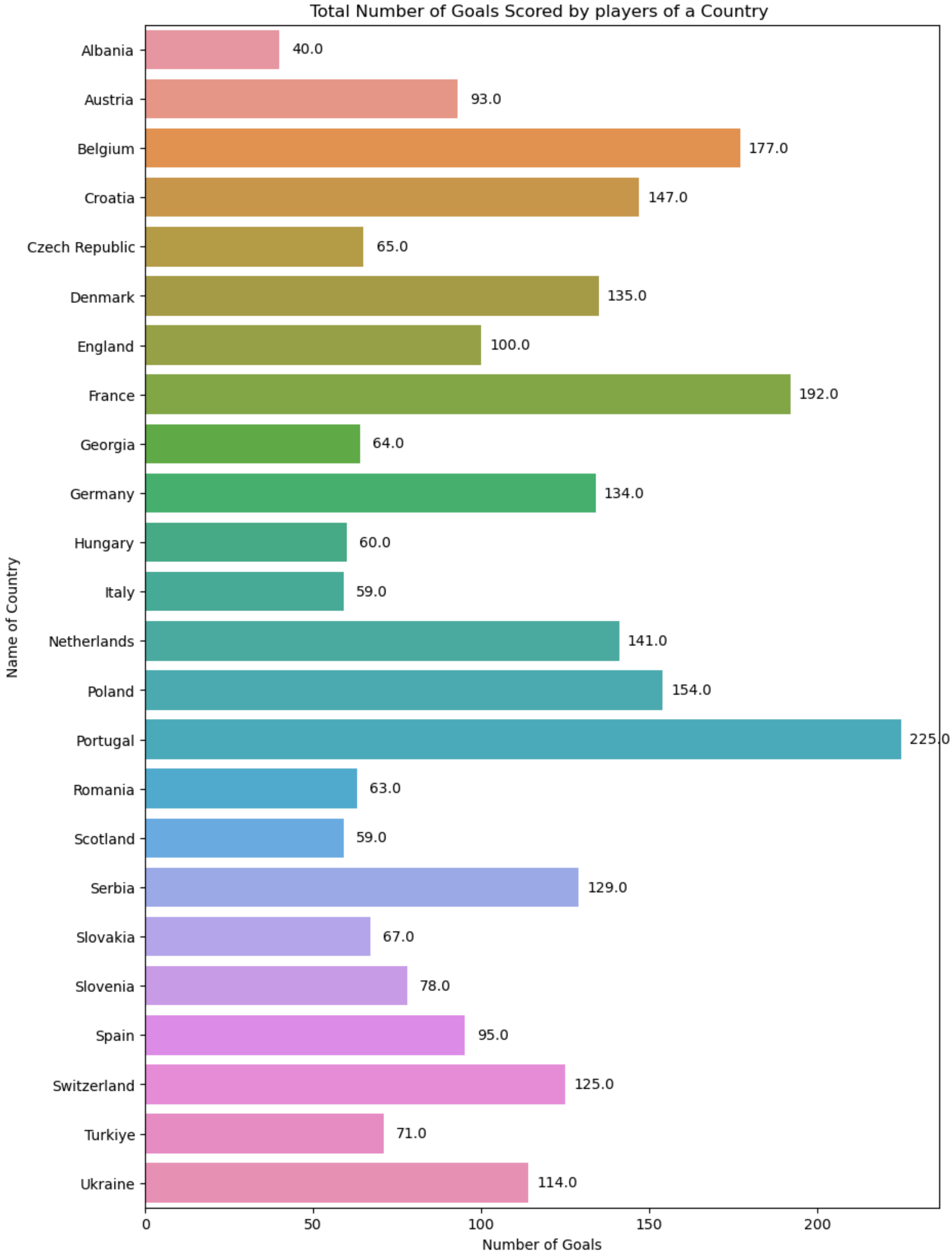
```
# Define vertical Bar Graph using matplotlib
foot = data['Foot'].value_counts()
plt.figure(figsize=(10, 6))
foot.plot(kind='bar')
plt.title('Distribution of Player according to their dominant foot')
plt.xlabel('Foot')
plt.ylabel('Number of Players')
plt.show()
```



Majority of the players are right-footed.

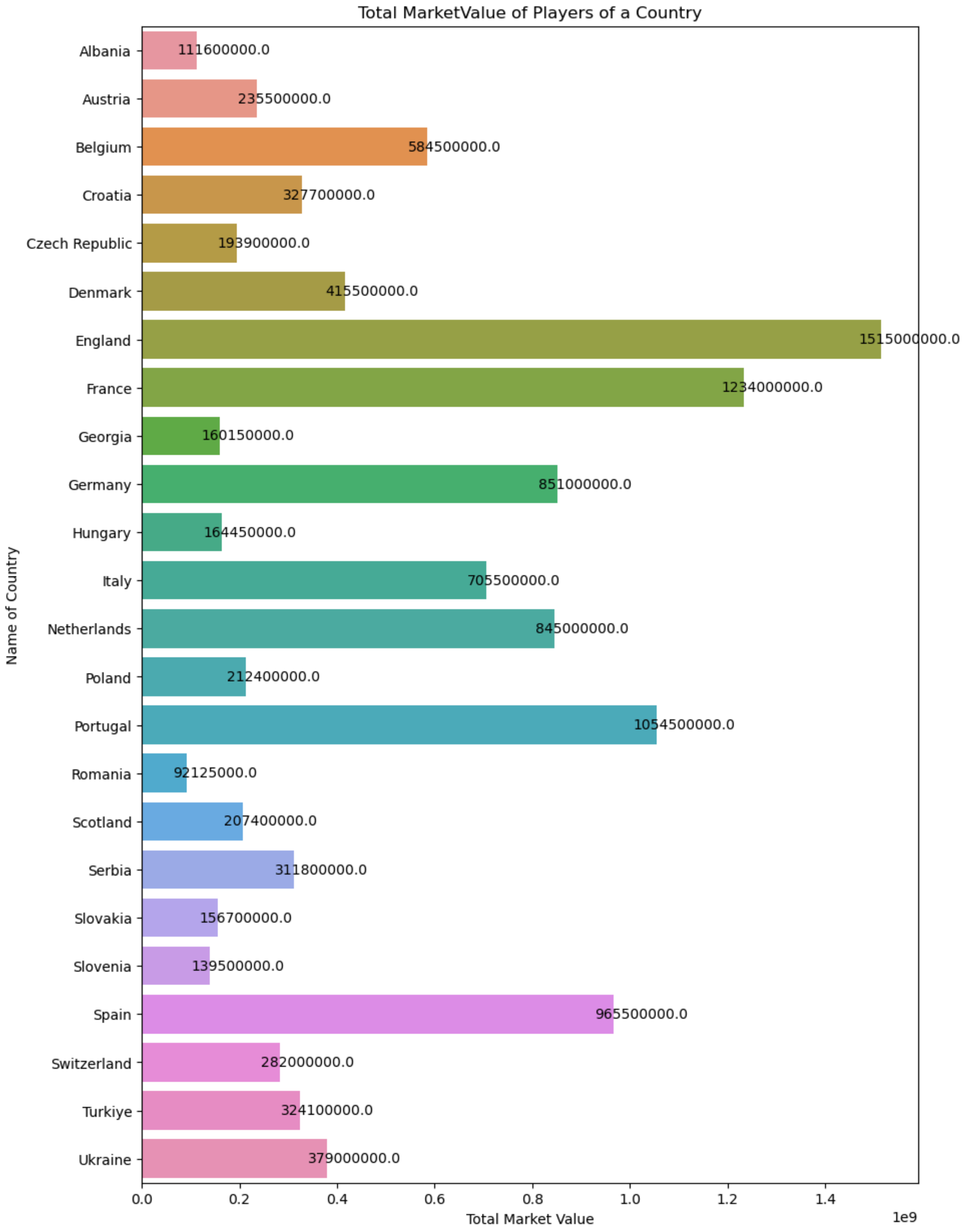
In [9]:

```
# Define horizontal Bar Graph using matplotlib
data_grouped = data.groupby('Country').sum().reset_index()
plt.figure(figsize=(10, 15))
goalplot=sns.barplot(x='Goals',y='Country',data = data_grouped)
plt.title('Total Number of Goals Scored by players of a Country ')
plt.xlabel('Number of Goals')
plt.ylabel('Name of Country')
for p in goalplot.patches:
    goalplot.annotate(format(p.get_width(), '.1f'),
                     (p.get_width(), p.get_y() + p.get_height() / 2.),
                     ha = 'center', va = 'center',
                     xytext = (20, 0),
                     textcoords = 'offset points')

plt.show()
```



Players of Portugal as a collective have scored the highest number of goals for their country. Albania ranks last with 40.

In [10]:
```
# Define horizontal Bar Graph using matplotlib
plt.figure(figsize=(10, 15))
goalplot2=sns.barplot(x='MarketValue',y='Country',data = data_grouped)
plt.title('Total MarketValue of Players of a Country')
plt.xlabel('Total Market Value')
plt.ylabel('Name of Country')
for p in goalplot2.patches:
    goalplot2.annotate(format(p.get_width(), '.1f'),
                       (p.get_width(), p.get_y() + p.get_height() / 2.),
                       ha = 'center', va = 'center',
                       xytext = (20, 0),
                    textcoords = 'offset points')

plt.show()
```



The Dataset unfortunately doesnot specify any particular currency so we will just call it units. English players have the highest combined Market Value with total exceeding 1.5 billion units.

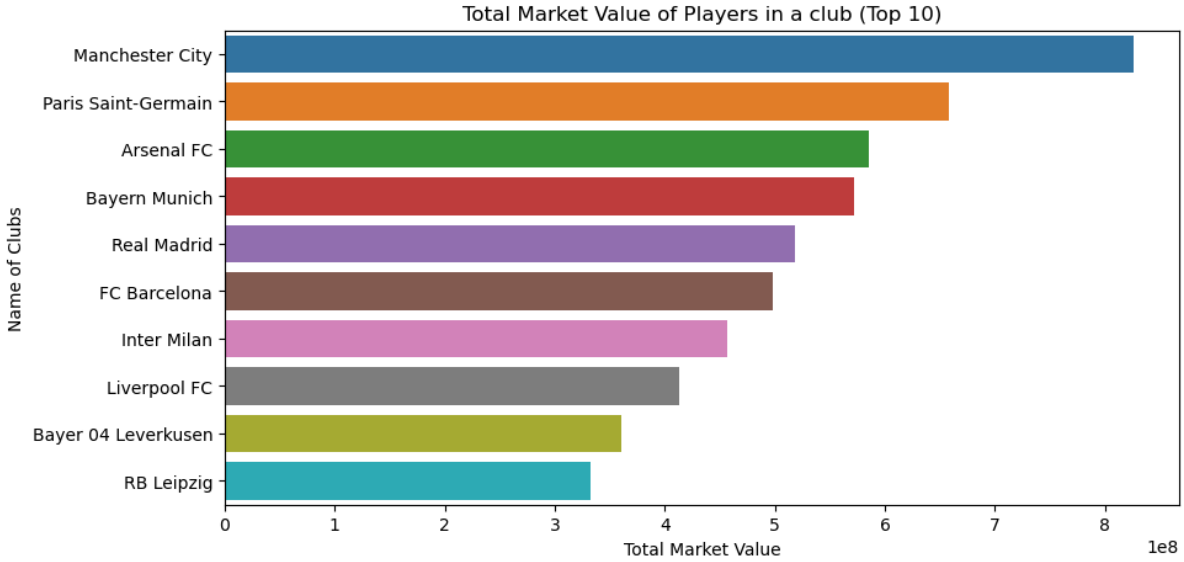
In [11]:

```
#Player with highest Market Value
most_mv_index = data['MarketValue'].idxmax()
player_max_value = data.loc[most_caps_index,['Name','MarketValue']]

print("Player with the highest Market Value:")
print(player_max_value)
```

```
Player with the highest Market Value:
Name           Cristiano Ronaldo
MarketValue             15000000
Name: 596, dtype: object
```

In [12]:
```
data_group = data.groupby('Club').sum().reset_index()
top_10_clubs = data_group.sort_values(by='MarketValue', ascending=False).head(10)
```

In [14]:
```
# Define horizontal Bar Graph using matplotlib
plt.figure(figsize=(10, 5))
sns.barplot(x='MarketValue',y='Club',data = top_10_clubs)
plt.title('Total Market Value of Players in a club (Top 10)')
plt.xlabel('Total Market Value')
plt.ylabel('Name of Clubs')
plt.show()
```



This notebook is made to help people get started, quite a few more comparisons and plots can be done. Hope this helps!!!

In [18]:
```
#checking duplicate values
data= pd.read_csv('euro2024_players.csv')
data.nunique()
```

Out[18]:
```
Name           623
Position        13
Age             24
Club           219
Height          34
Foot             4
Caps           113
Goals           39
MarketValue     83
Country         24
dtype: int64
```

In [19]:
```
# describing the data
data.describe()
```

Out[19]:

	Age	Height	Caps	Goals	MarketValue
count	623.000000	623.000000	623.000000	623.000000	6.230000e+02
mean	27.040128	184.181380	30.338684	4.152488	1.840903e+07
std	4.124275	6.569258	30.987902	10.086803	2.426195e+07
min	16.000000	167.000000	0.000000	0.000000	5.000000e+04
25%	24.000000	180.000000	7.000000	0.000000	2.900000e+06
50%	27.000000	185.000000	21.000000	1.000000	9.000000e+06
75%	30.000000	189.000000	42.000000	4.000000	2.500000e+07
max	41.000000	202.000000	206.000000	128.000000	1.800000e+08

In [20]:

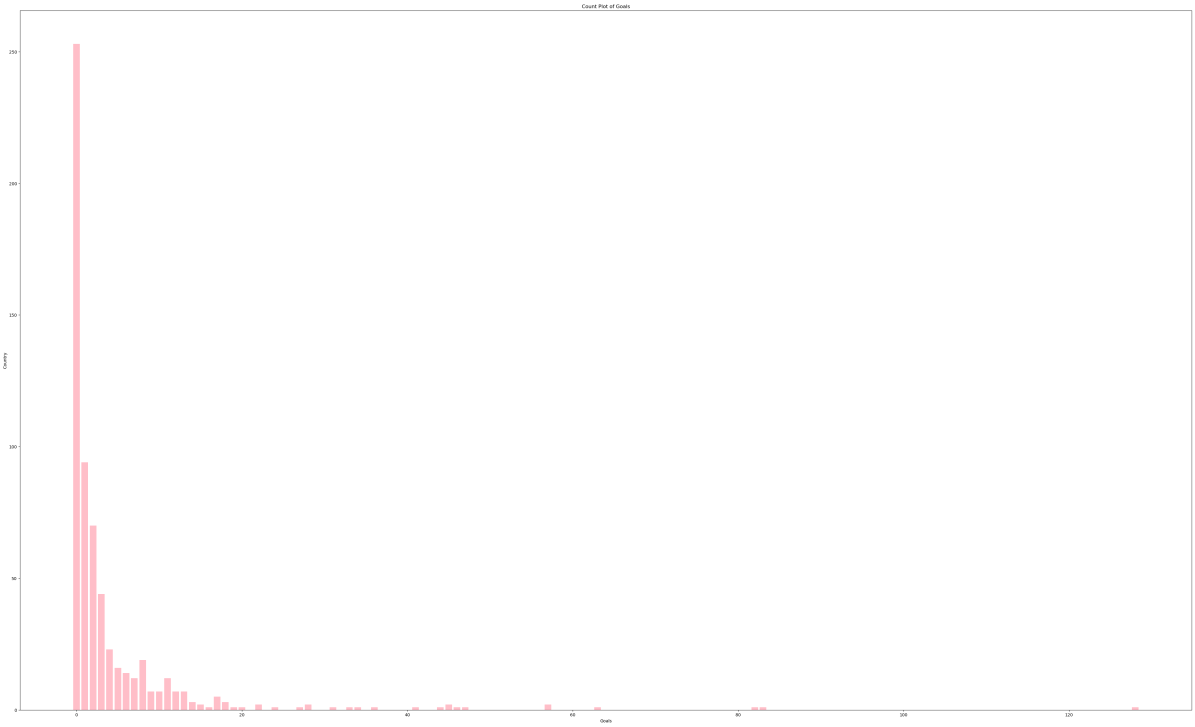
```
#column to list
data.columns.tolist()
```

Out[20]:

```
['Name',
 'Position',
 'Age',
 'Club',
 'Height',
 'Foot',
 'Caps',
 'Goals',
 'MarketValue',
 'Country']
```

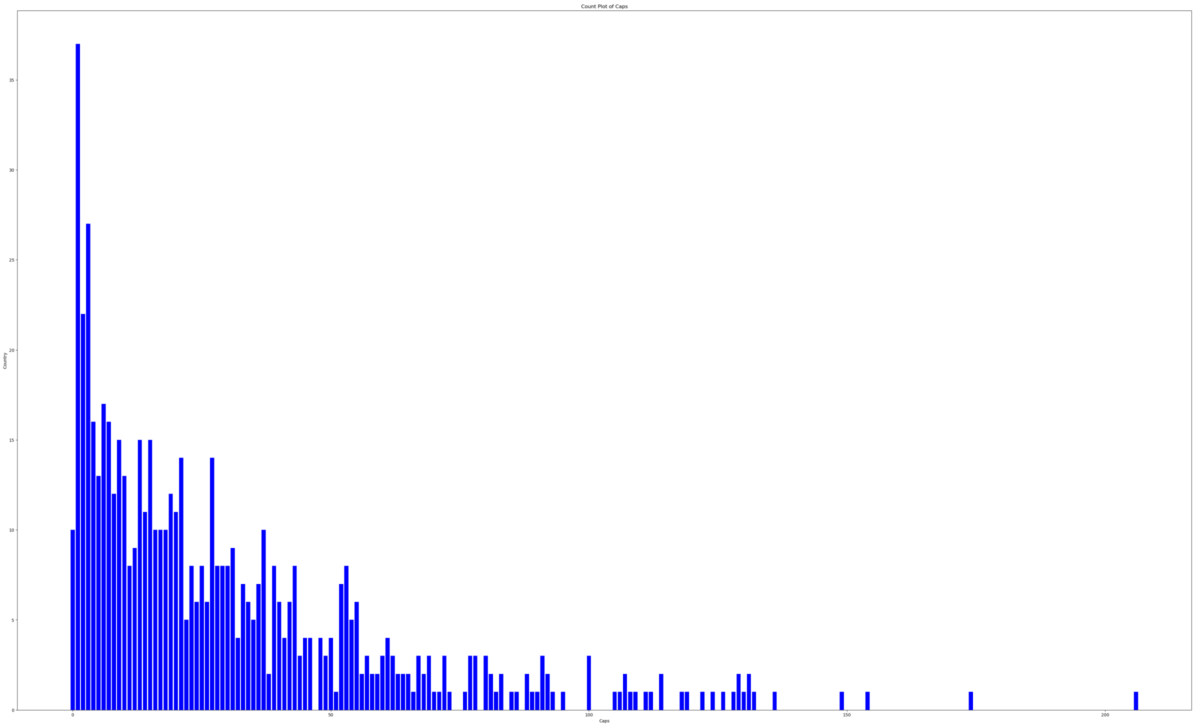
In [33]:

```
# Assuming 'data' is your DataFrame
Goals_counts = data['Goals'].value_counts()

# Using Matplotlib to create a count plot
plt.figure(figsize=(50,30))
plt.bar(Goals_counts.index, Goals_counts, color='pink')
plt.title('Count Plot of Goals')
plt.xlabel('Goals')
plt.ylabel('Country')
plt.show()
```



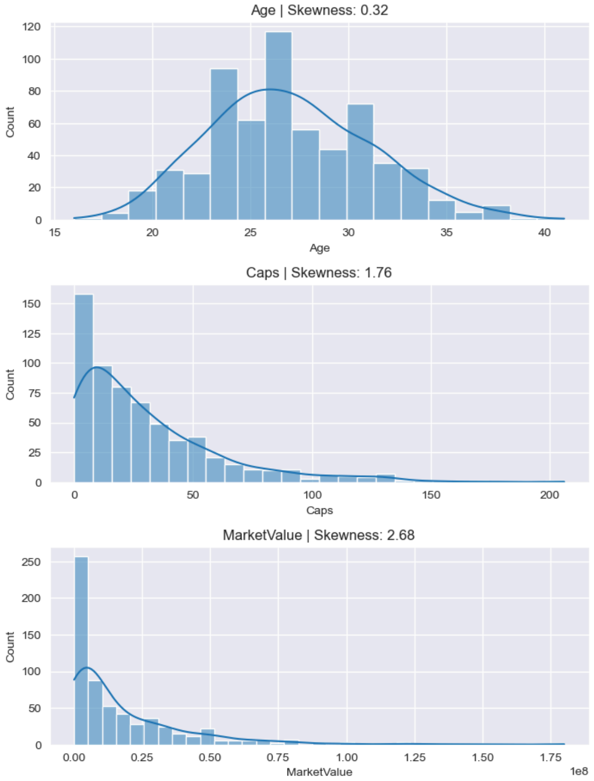
In [34]:

```
# Assuming 'data' is your DataFrame
Caps_counts = data['Caps'].value_counts()

# Using Matplotlib to create a count plot
plt.figure(figsize=(50,30))
plt.bar(Caps_counts.index, Caps_counts, color='Blue')
plt.title('Count Plot of Caps')
plt.xlabel('Caps')
plt.ylabel('Country')
plt.show()
```



In [41]:

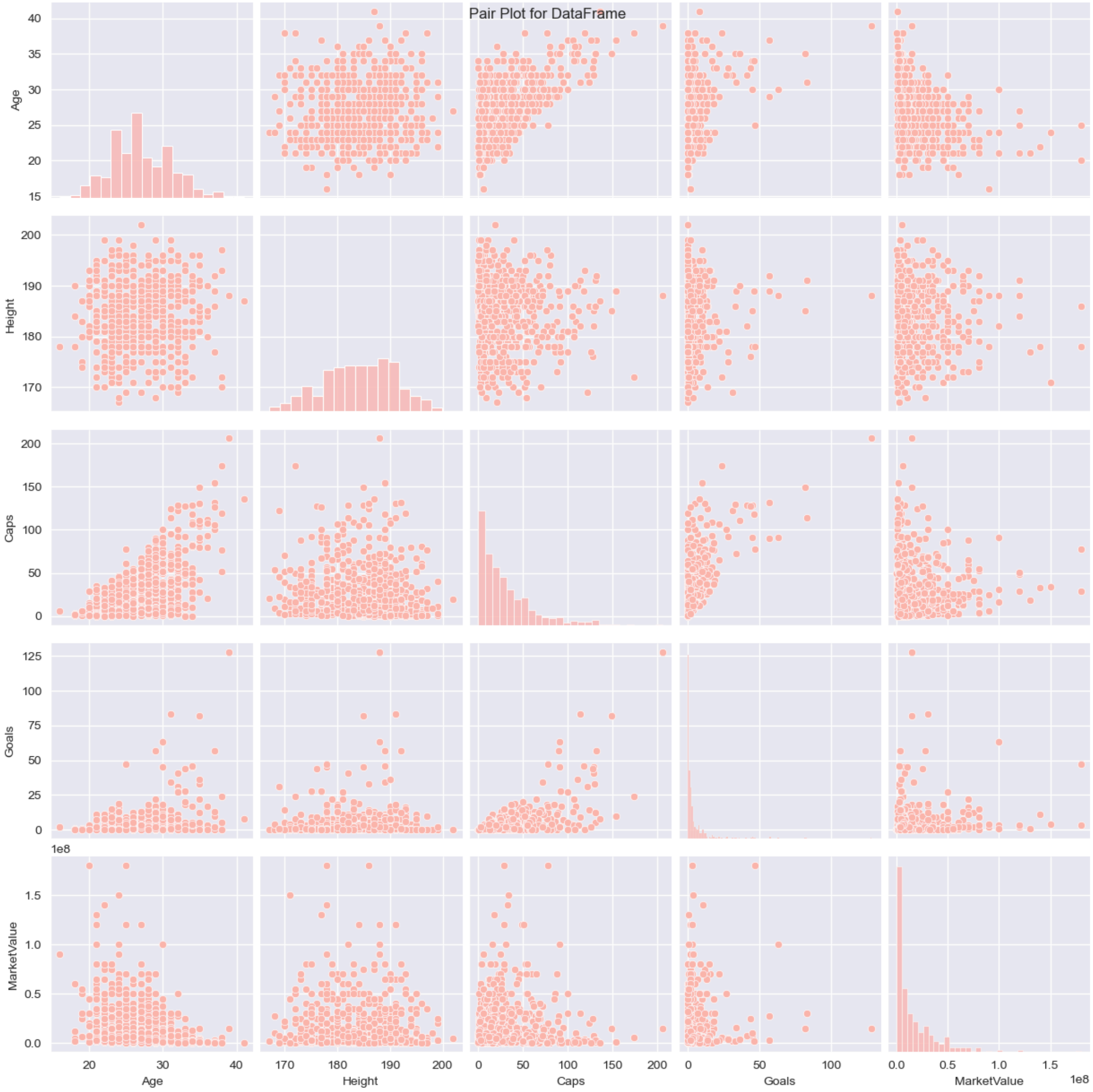
```
 # Set Seaborn style
sns.set_style("darkgrid")

# Identify numerical columns
numerical_columns = data.select_dtypes(include=["int64", "float64"]).columns

# Plot distribution of each numerical feature
plt.figure(figsize=(14, len(numerical_columns) * 3))
for idx, feature in enumerate(numerical_columns, 1):
    plt.subplot(len(numerical_columns), 2, idx)
    sns.histplot(data[feature], kde=True)
    plt.title(f"{feature} | Skewness: {round(data[feature].skew(), 2)}")

# Adjust layout and show plots
plt.tight_layout()
plt.show()
```



In [51]:

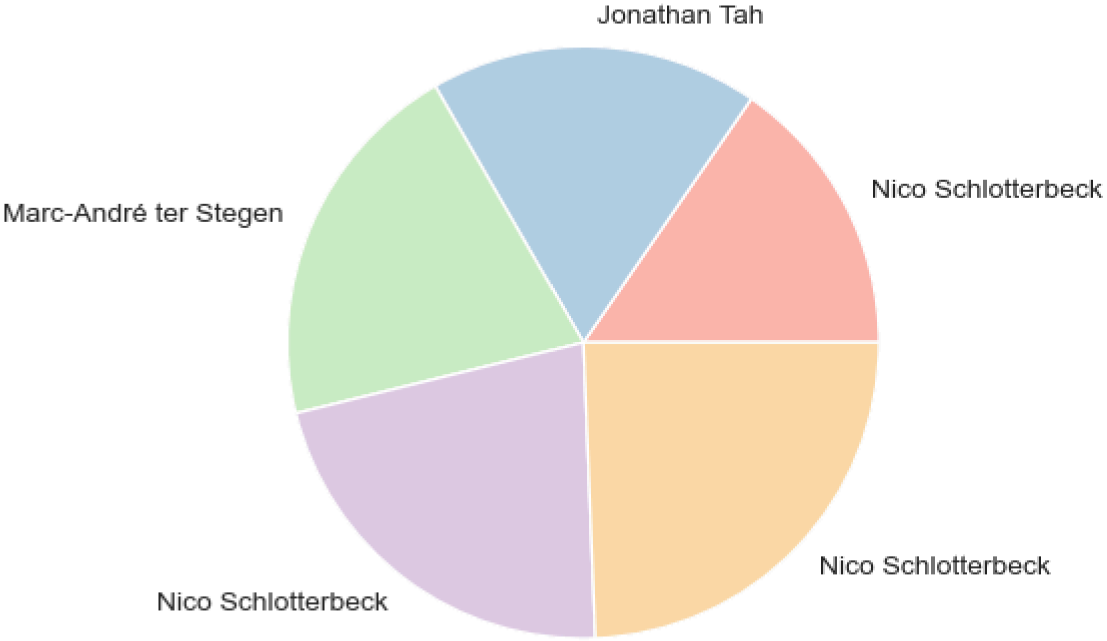
```
# Set the color palette
sns.set_palette("Pastel1")

# Assuming 'data' is your DataFrame
plt.figure(figsize=(10, 6))

# Using Seaborn to create a pair plot with the specified color palette
sns.pairplot(data)

plt.suptitle('Pair Plot for DataFrame')
plt.show()
```

```
C:\Users\INDRAJIT\OneDrive\New folder\Lib\site-packages\seaborn\axisgrid.py:118: U
serWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
<Figure size 1000x600 with 0 Axes>
```



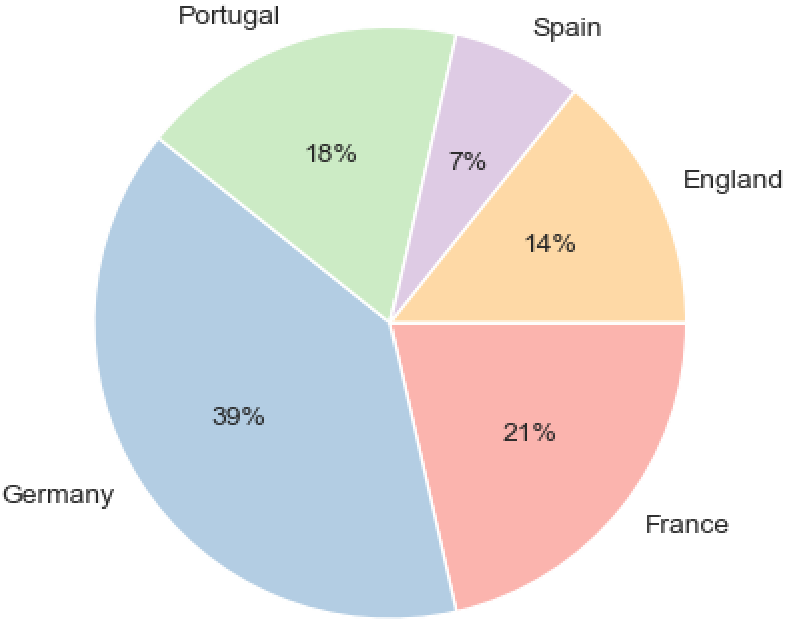
In [68]:

```
# Pie Charts using Matplotlib in Python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
# Mesurment of Age of the following Players.
x = np.array([24,28,32,34,38])
mylabels = ["Nico Schlotterbeck","Jonathan Tah","Marc-André ter Stegen","Nico Schlo
plt.pie(x, labels = mylabels)
plt.show()
```



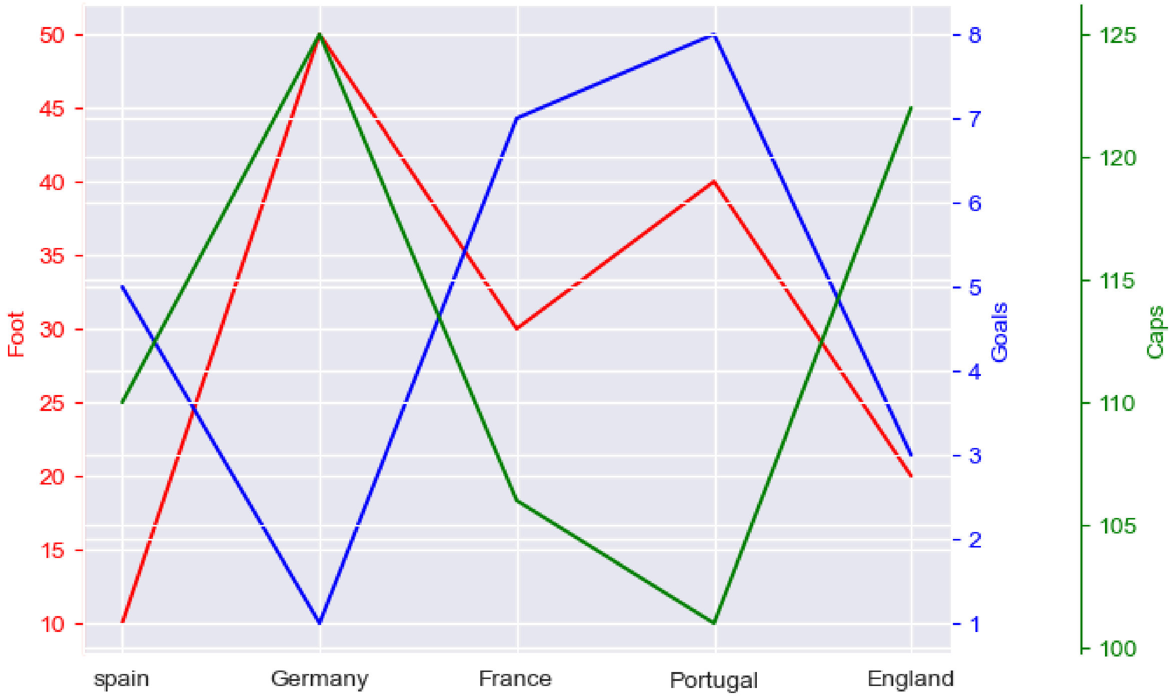
In [72]:

```
 # Pie Charts using Matplotlib in Python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
# Define win rates of following countries.
labels = ('France','Germany','Portugal','Spain','England')
sizes = ([30,55,25,10,20])
plt.pie(sizes, labels = labels, autopct = '%1.f%%', counterclock = False)
#Display th figure
plt.show()
```



In [101…

```
#Line Charts using Matplotlib in Python
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

fig, ax1 = plt.subplots()
Countries = ['spain','Germany','France','Portugal','England']
Foot = [10,50,30,40,20]
Goals = [5,1,7,8,3]
Caps = [110,125,106,101,122]


ax1.plot(Countries,Foot,color = "red")
ax2 = ax1.twinx()
ax2.plot(Countries,Goals,color = "blue")
ax3 = ax1.twinx()
ax3.plot(Countries,Caps,color = "green")
#ax3.spines['right'].set_position(('outward',60))
ax3.spines['right'].set_position(('axes',1.15))

ax1.set_ylabel("Foot",color="red")
ax2.set_ylabel("Goals",color="blue")
ax3.set_ylabel("Caps",color="green")

ax1.tick_params(axis='y',colors = "red")
ax2.tick_params(axis='y',colors = "blue")
ax3.tick_params(axis='y',colors = "green")

ax2.spines['left'].set_color("red")
ax3.spines['right'].set_color("blue")
ax3.spines['right'].set_color("green")

plt.show()

#fig.tight_layout()
fig.savefig("3-axis-v2.png",bbox_inches='tight')
```



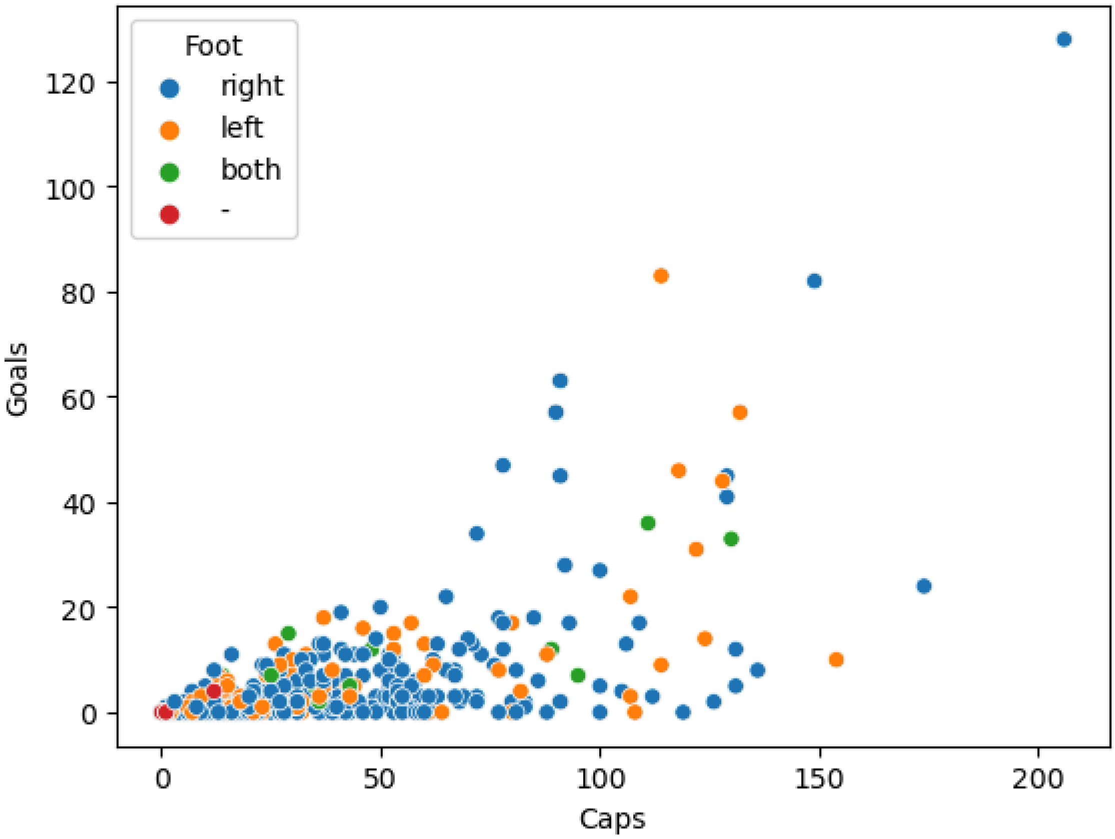
In [1]:

```
# Scatterplot using Seaborn in Python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
Countries_data = pd.read_csv('euro2024_players.csv')
```

```
sns.scatterplot(x = Countries_data['Caps'], y = Countries_data['Goals'], hue = Cour
```

Out[1]:

```
<Axes: xlabel='Caps', ylabel='Goals'>
```

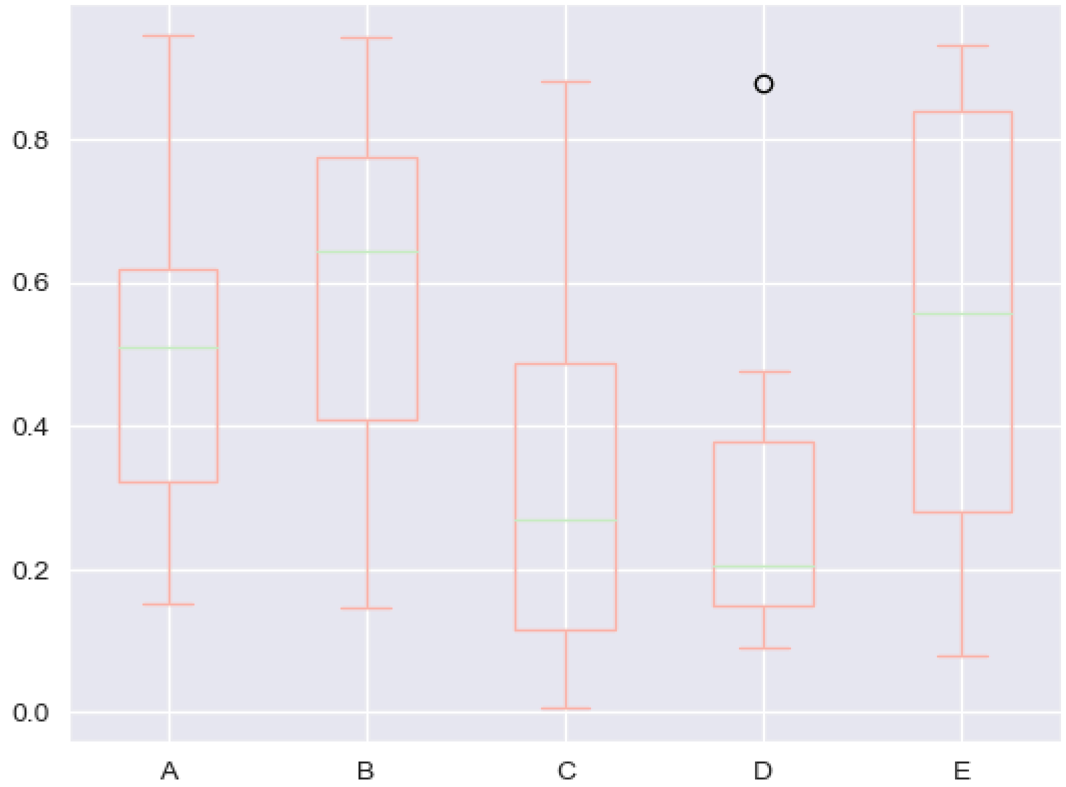


In [110…

```
# Boxplot using seaborn in Python
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
data = pd.read_csv('euro2024_players.csv')
data = pd.DataFrame(np.random.rand(10,5), columns =['A','B','C','D','E'])
data.plot.box(grid='True')
```

Out[110]:

```
<Axes: >
```



In [113…

```
# Boxplot using seaborn in Python
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns

data = pd.read_csv('euro2024_players.csv')
data.head()
```

Out[113]:

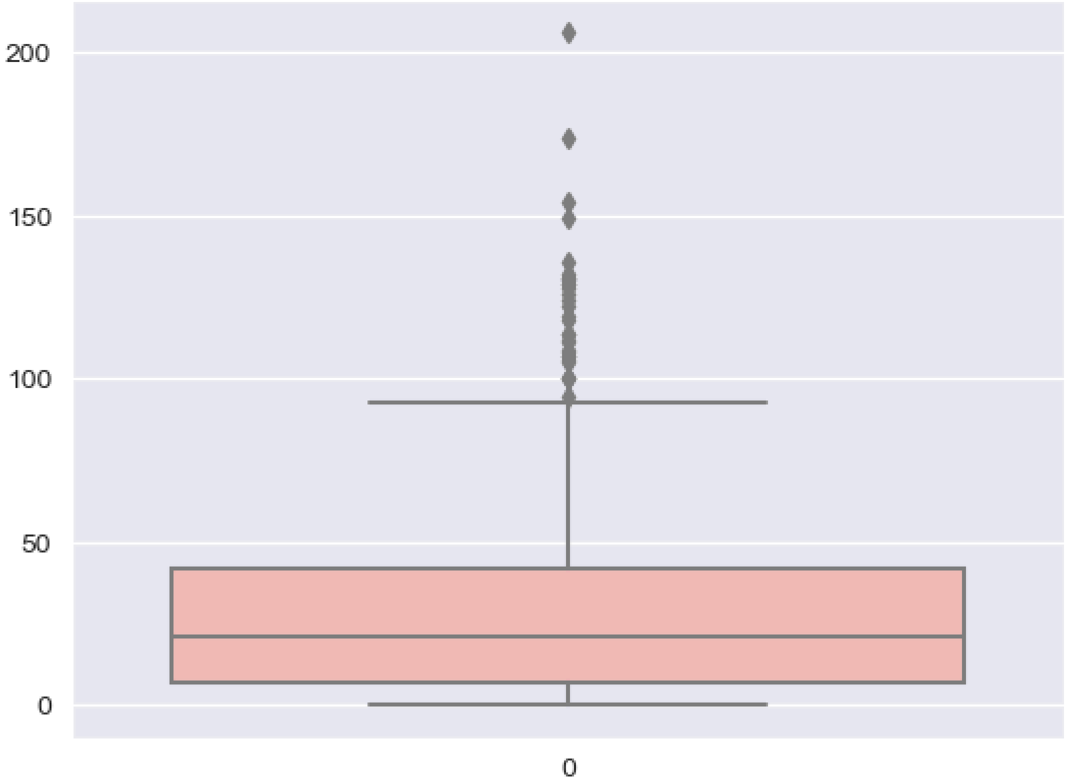
	Name	Position	Age	Club	Height	Foot	Caps	Goals	MarketValue	Country
0	Marc-André ter Stegen	Goalkeeper	32	FC Barcelona	187	right	40	0	28000000	Germany
1	Manuel Neuer	Goalkeeper	38	Bayern Munich	193	right	119	0	4000000	Germany
2	Oliver Baumann	Goalkeeper	34	TSG 1899 Hoffenheim	187	right	0	0	3000000	Germany
3	Nico Schlotterbeck	Centre-Back	24	Borussia Dortmund	191	left	12	0	40000000	Germany
4	Jonathan Tah	Centre-Back	28	Bayer 04 Leverkusen	195	right	25	0	30000000	Germany

In [118…

```
# Boxplot using seaborn in Python
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns

data = pd.read_csv('euro2024_players.csv')
sns.boxplot(data["Caps"])
```

Out[118]:

```
<Axes: >
```



In [119…

```
data.describe()
```

Out[119]:

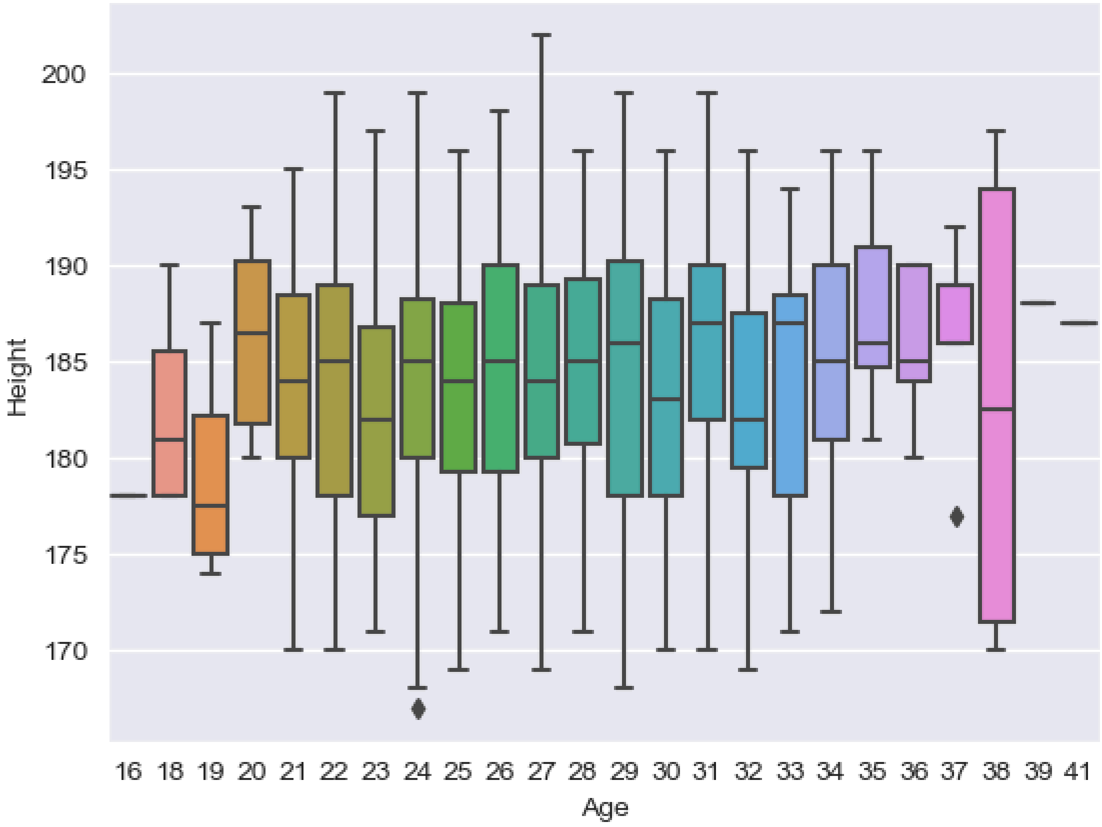
	Age	Height	Caps	Goals	MarketValue
count	623.000000	623.000000	623.000000	623.000000	6.230000e+02
mean	27.040128	184.181380	30.338684	4.152488	1.840903e+07
std	4.124275	6.569258	30.987902	10.086803	2.426195e+07
min	16.000000	167.000000	0.000000	0.000000	5.000000e+04
25%	24.000000	180.000000	7.000000	0.000000	2.900000e+06
50%	27.000000	185.000000	21.000000	1.000000	9.000000e+06
75%	30.000000	189.000000	42.000000	4.000000	2.500000e+07
max	41.000000	202.000000	206.000000	128.000000	1.800000e+08

In [127…

```
# Boxplot using seaborn in Python
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
Countries_data = pd.read_csv('euro2024_players.csv')
sns.boxplot(x = "Age", y = "Height", data= Countries_data)
```

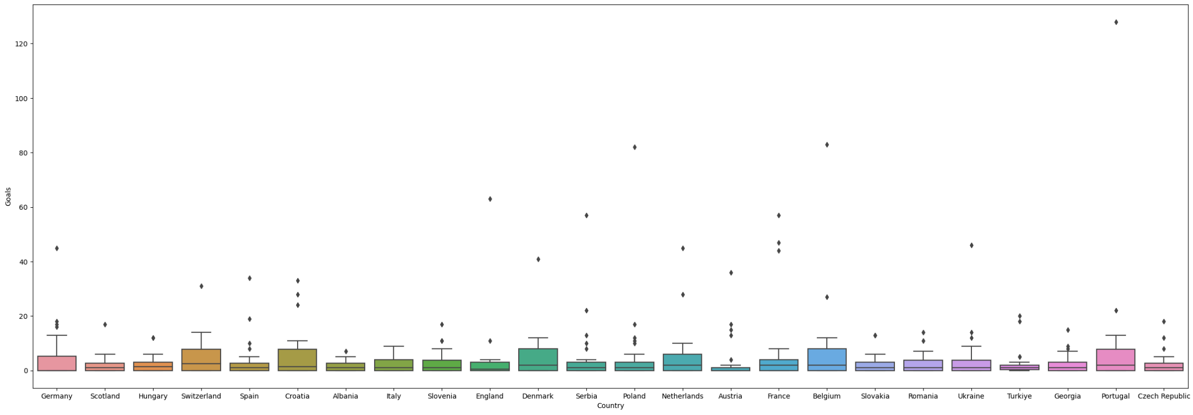
Out[127]:

```
<Axes: xlabel='Age', ylabel='Height'>
```



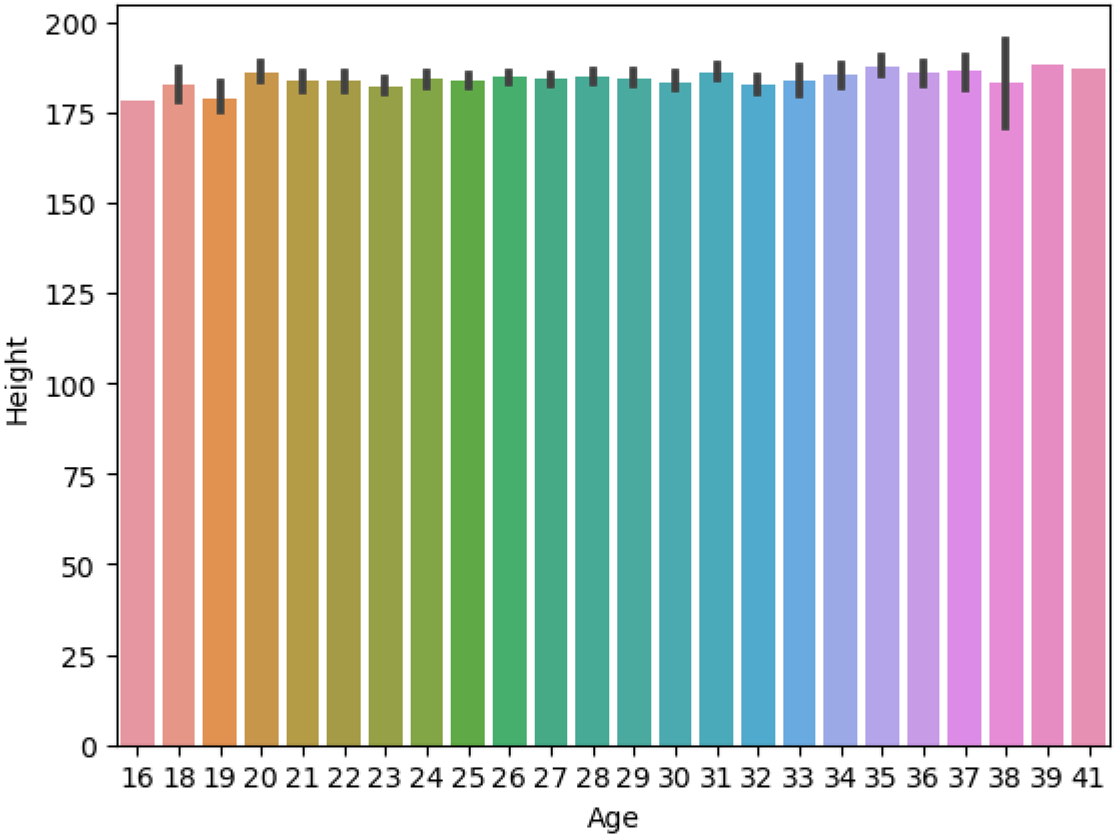
```
In [30]: # Define Comperison Box plot using seaborn in Python
         plt.figure(figsize=(30,10))
         sns.boxplot(x = "Country", y = "Goals",data = Countries_data)
```

```
Out[30]: <Axes: xlabel='Country', ylabel='Goals'>
```



```
In [31]: # Barplot using seaborn in Python
         import matplotlib.pyplot as plt
         import pandas as pd
         import numpy as np
         import seaborn as sns

         Countries_data = pd.read_csv('euro2024_players.csv')
         sns.barplot(x = "Age", y = "Height", data= Countries_data)
```

```
Out[31]: <Axes: xlabel='Age', ylabel='Height'>
```

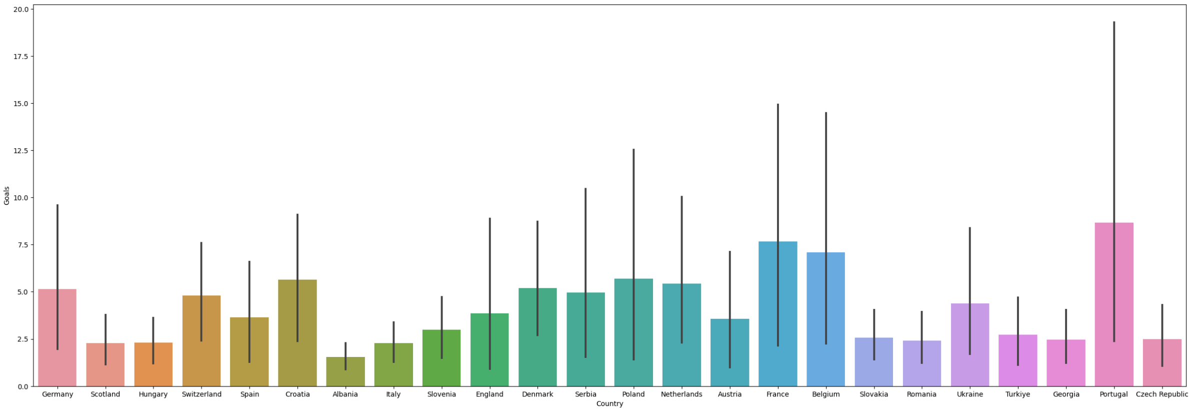


In [8]:

```
# Barplot using seaborn in Python
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns

plt.figure(figsize=(30,10))
Countries_data = pd.read_csv('euro2024_players.csv')
sns.barplot(x = "Country", y = "Goals", data= Countries_data)
```

Out[8]:

```
<Axes: xlabel='Country', ylabel='Goals'>
```

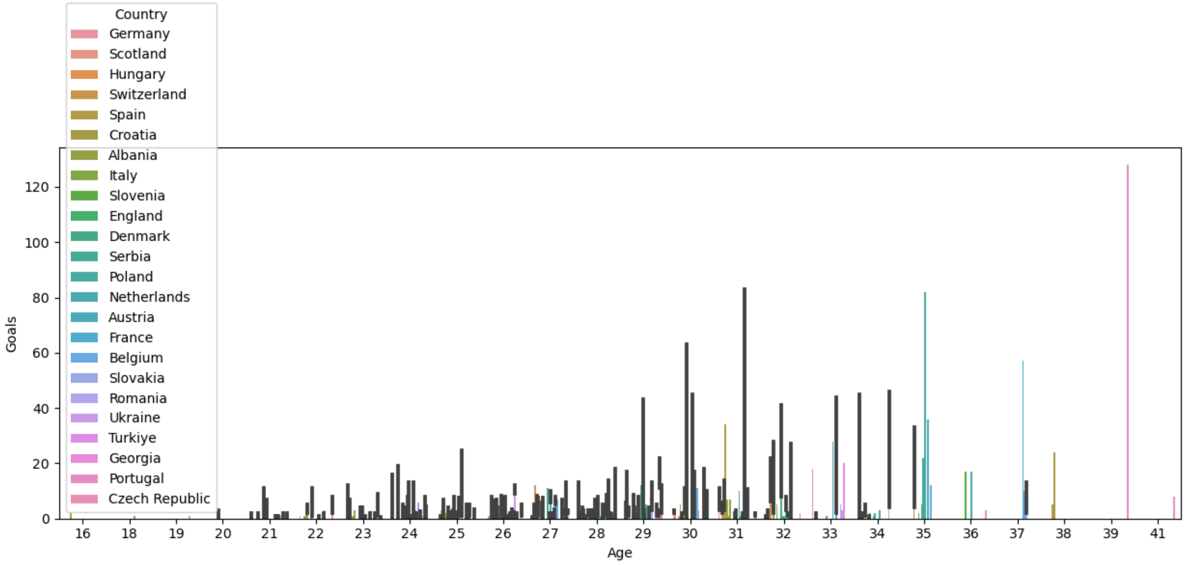


In [12]:

```
# Define Comperison Bar graph using seaborn in Python
plt.figure(figsize = (15,5))

sns.barplot( x = 'Age',y = 'Goals', hue = "Country", data= Countries_data )
```

Out[12]:

```
<Axes: xlabel='Age', ylabel='Goals'>
```



In [13]:

```
# Comparison Pie Charts using in python
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

Countries_data = pd.read_csv('euro2024_players.csv')
Countries_data = {"Countries":['France','Germany','Portugal','Spain','England'],
                  "Goals":[15,8,18,10,5]}
data = pd.DataFrame(Countries_data)
data
```

Out[13]:

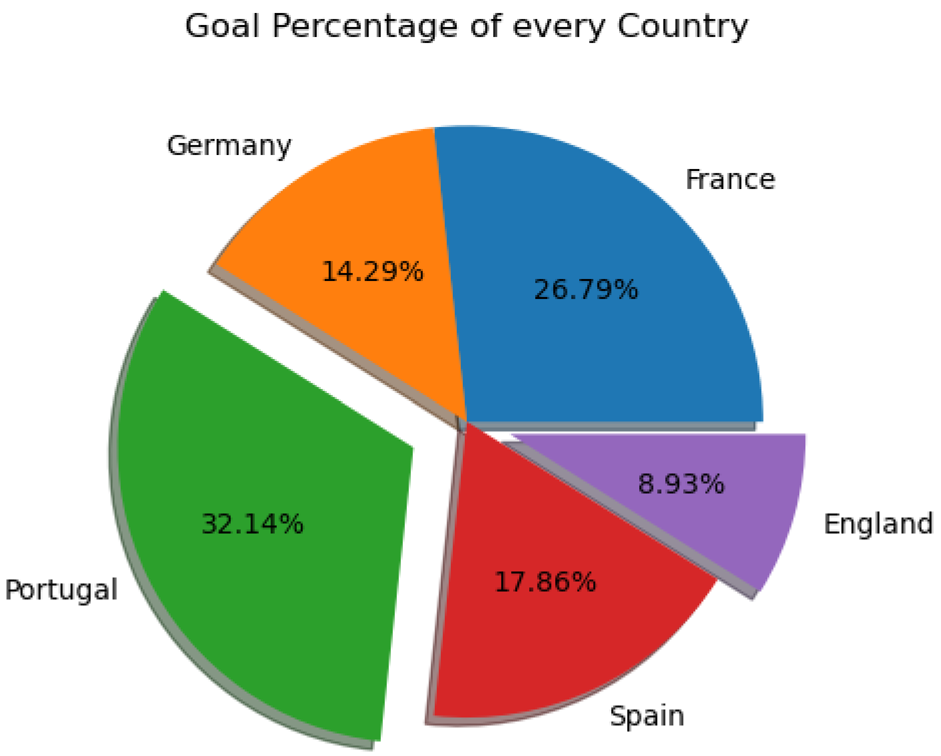
	Countries	Goals
0	France	15
1	Germany	8
2	Portugal	18
3	Spain	10
4	England	5

In [28]:

```
# Define Comperison Pie chart using matplotlib in Python
plt.pie(data['Goals'],labels=data['Countries'] ,autopct='%1.2f%%',explode=(0,0,0.2,
plt.title('Goal Percentage of every Country')
```

Out[28]:

```
Text(0.5, 1.0, 'Goal Percentage of every Country')
```



In [28]:

```
#import the liberies
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
```

In [6]:

```
# import the dataset
data= pd.read_csv('euro2024_players.csv')
data
```

Out[6]:

	Name	Position	Age	Club	Height	Foot	Caps	Goals	MarketValue	Countr
0	Marc-André ter Stegen	Goalkeeper	32	FC Barcelona	187	right	40	0	28000000	German
1	Manuel Neuer	Goalkeeper	38	Bayern Munich	193	right	119	0	4000000	German
2	Oliver Baumann	Goalkeeper	34	TSG 1899 Hoffenheim	187	right	0	0	3000000	German
3	Nico Schlotterbeck	Centre-Back	24	Borussia Dortmund	191	left	12	0	40000000	German
4	Jonathan Tah	Centre-Back	28	Bayer 04 Leverkusen	195	right	25	0	30000000	German
...	...	...	...	...	...	...	...	...	...	
618	Adam Hlozek	Second Striker	21	Bayer 04 Leverkusen	188	right	31	2	12000000	Czec Republi
619	Patrik Schick	Centre-Forward	28	Bayer 04 Leverkusen	191	left	37	18	22000000	Czec Republi
620	Mojmír Chytil	Centre-Forward	25	SK Slavia Prague	187	-	12	4	6500000	Czec Republi
621	Jan Kuchta	Centre-Forward	27	AC Sparta Prague	185	right	20	3	5000000	Czec Republi
622	Tomas Chory	Centre-Forward	29	FC Viktoria Plzen	199	right	3	2	3200000	Czec Republi

623 rows × 10 columns

In [8]:

```
# import the first five data from the entire dataset
data.head(5)
```

Out[8]:

	Name	Position	Age	Club	Height	Foot	Caps	Goals	MarketValue	Country
0	Marc-André ter Stegen	Goalkeeper	32	FC Barcelona	187	right	40	0	28000000	Germany
1	Manuel Neuer	Goalkeeper	38	Bayern Munich	193	right	119	0	4000000	Germany
2	Oliver Baumann	Goalkeeper	34	TSG 1899 Hoffenheim	187	right	0	0	3000000	Germany
3	Nico Schlotterbeck	Centre-Back	24	Borussia Dortmund	191	left	12	0	40000000	Germany
4	Jonathan Tah	Centre-Back	28	Bayer 04 Leverkusen	195	right	25	0	30000000	Germany

In [9]:

```
# import the last fine data from the entire dataset
data.tail(5)
```

Out[9]:

	Name	Position	Age	Club	Height	Foot	Caps	Goals	MarketValue	Country
618	Adam Hlozek	Second Striker	21	Bayer 04 Leverkusen	188	right	31	2	12000000	Czech Republic
619	Patrik Schick	Centre-Forward	28	Bayer 04 Leverkusen	191	left	37	18	22000000	Czech Republic
620	Mojmír Chytil	Centre-Forward	25	SK Slavia Prague	187	-	12	4	6500000	Czech Republic
621	Jan Kuchta	Centre-Forward	27	AC Sparta Prague	185	right	20	3	5000000	Czech Republic
622	Tomas Chory	Centre-Forward	29	FC Viktoria Plzen	199	right	3	2	3200000	Czech Republic

In [10]:

```
# Define information about the entire dataset
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 623 entries, 0 to 622
Data columns (total 10 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   Name         623 non-null    object
 1   Position     623 non-null    object
 2   Age          623 non-null    int64 
 3   Club         623 non-null    object
 4   Height       623 non-null    int64 
 5   Foot         620 non-null    object
 6   Caps         623 non-null    int64 
 7   Goals        623 non-null    int64 
 8   MarketValue  623 non-null    int64 
 9   Country      623 non-null    object
dtypes: int64(5), object(5)
memory usage: 48.8+ KB
```

In [11]:

```
# Define Description of entire dataset
data.describe().T
```

Out[11]:

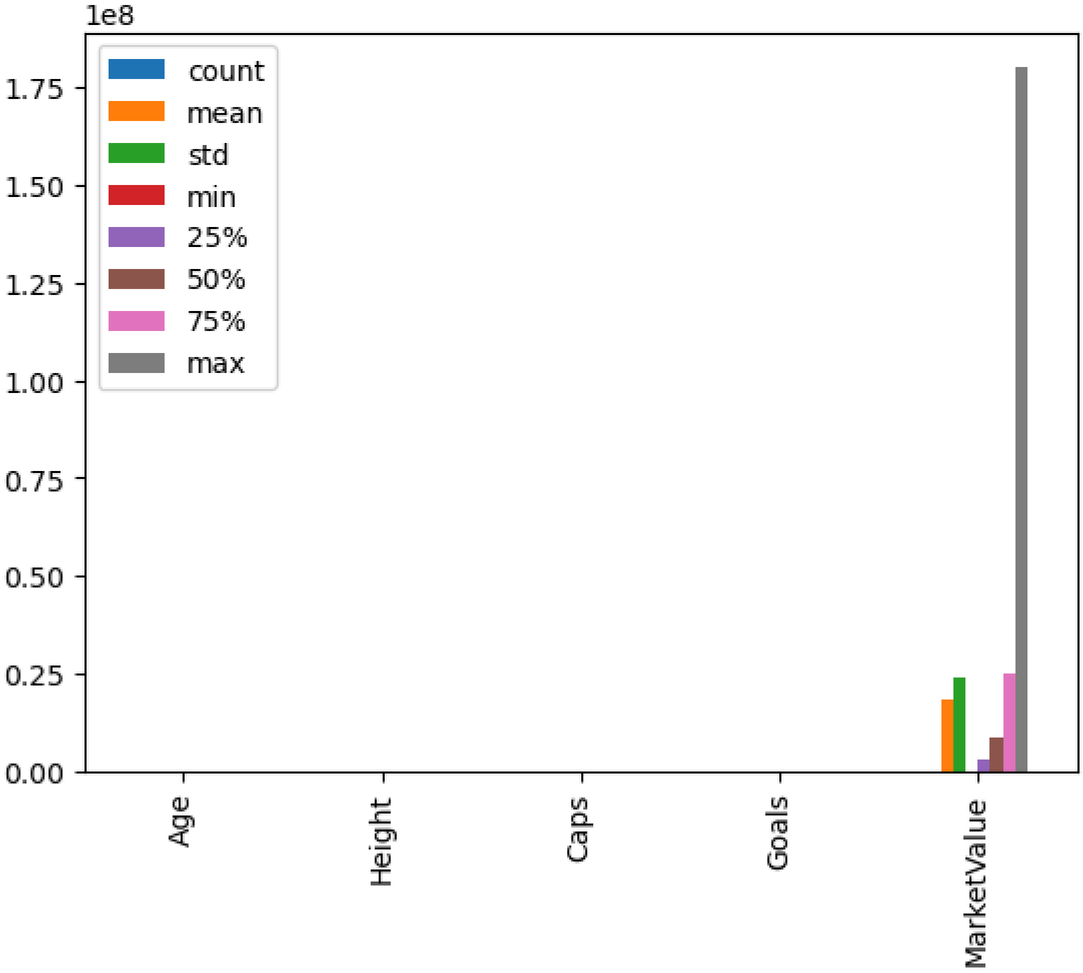
	count	mean	std	min	25%	50%	75%	
Age	623.0	2.704013e+01	4.124275e+00	16.0	24.0	27.0	30.0	
Height	623.0	1.841814e+02	6.569258e+00	167.0	180.0	185.0	189.0	
Caps	623.0	3.033868e+01	3.098790e+01	0.0	7.0	21.0	42.0	
Goals	623.0	4.152488e+00	1.008680e+01	0.0	0.0	1.0	4.0	
MarketValue	623.0	1.840903e+07	2.426195e+07	50000.0	2900000.0	9000000.0	25000000.0	18000

In [13]:

```
# Define T-plot using Bar Graph
data.describe().T.plot(kind='bar')
```

Out[13]:

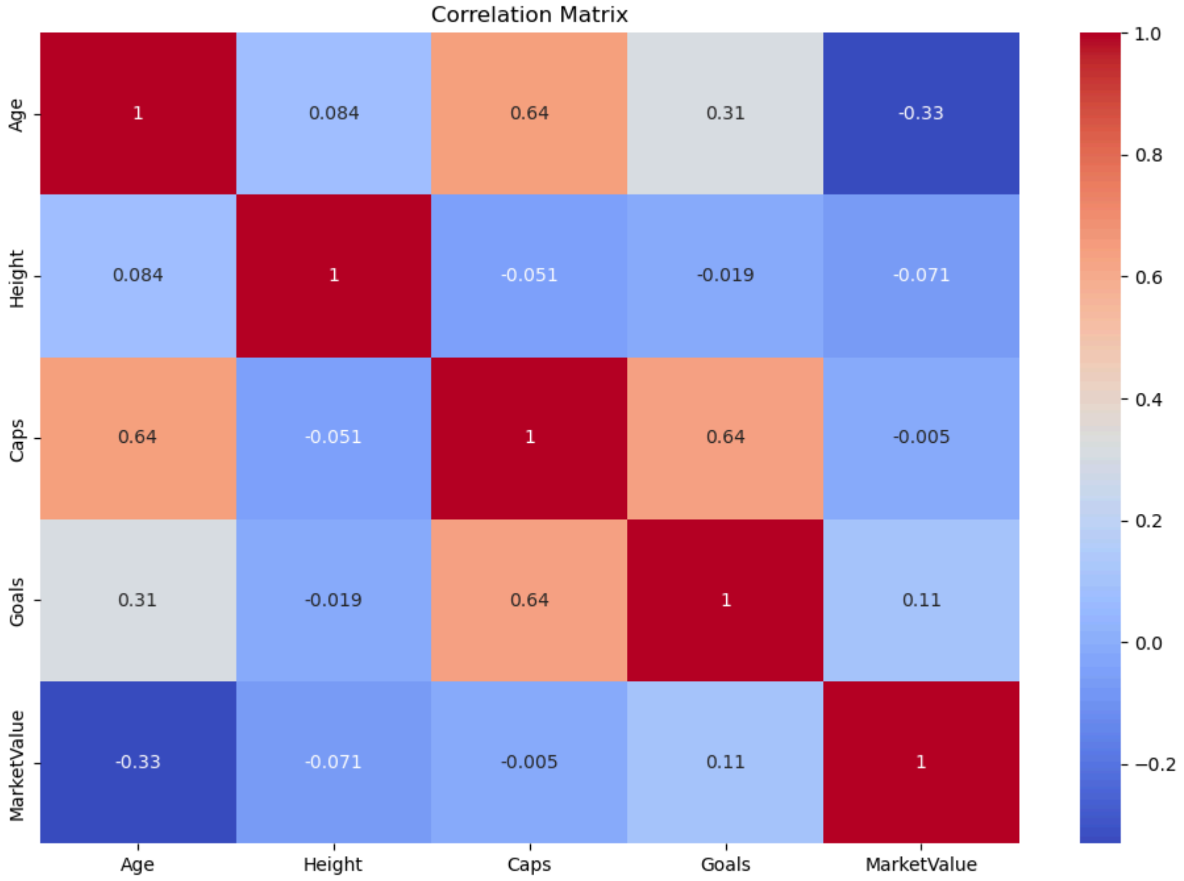
```
<Axes: >
```



In [15]:
```
# Checking the duplicate value into dataset
data.duplicated().sum()
```

Out[15]:
```
0
```

In [19]:
```
# Define Correlation Matrix using Seaborn in Python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

numeric_cols = data.select_dtypes(include=np.number).columns
plt.figure(figsize=(12, 8))
sns.heatmap(data[numeric_cols].corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```



In [20]:
```
# import column by using Python
data.columns.to_list()
```

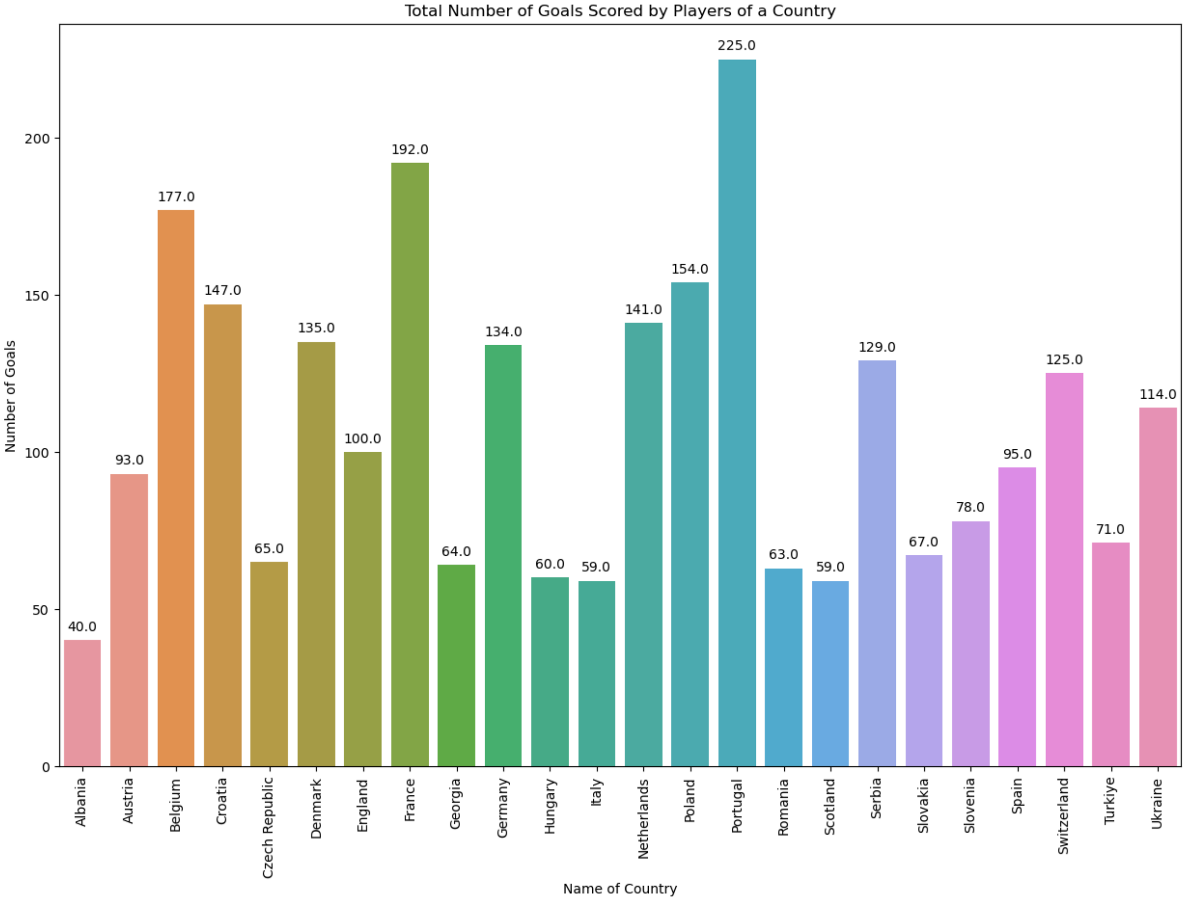
Out[20]:
```
['Name',
 'Position',
 'Age',
 'Club',
 'Height',
 'Foot',
 'Caps',
 'Goals',
 'MarketValue',
 'Country']
```

In [22]:
```
# Most Successful Goal Keeper
least_goals_index = data['Goals'].idxmin()


player_least_goals = data.loc[least_goals_index]

print("Player with the least goals:")
print(player_least_goals)
```

```
Player with the least goals:
Name           Marc-André ter Stegen
Position                  Goalkeeper
Age                               32
Club                    FC Barcelona
Height                           187
Foot                           right
Caps                              40
Goals                              0
MarketValue                 28000000
Country                      Germany
Name: 0, dtype: object
```

In [23]:

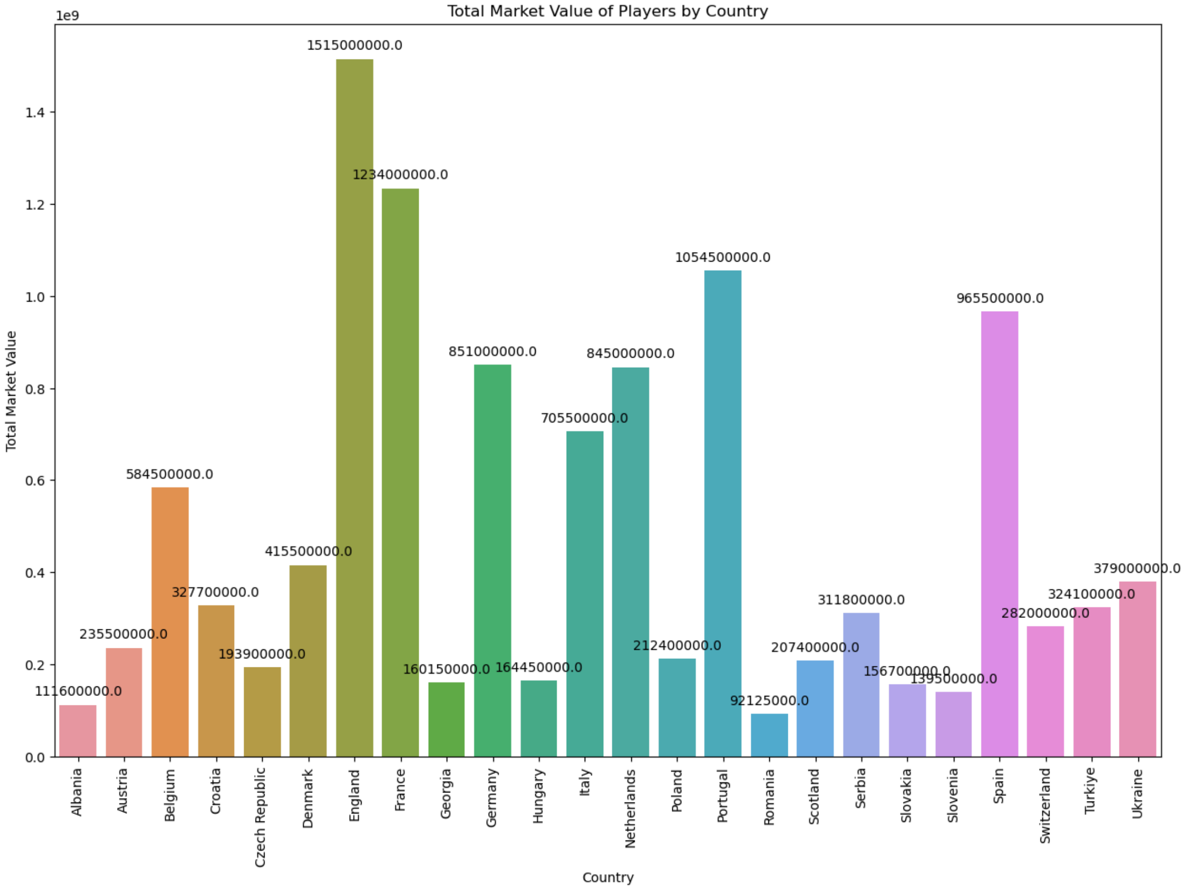
```
# Define vertical Bar Graph using matplotlib
data_grouped = data.groupby('Country').sum().reset_index()
plt.figure(figsize=(15, 10))
goalplot = sns.barplot(x='Country', y='Goals', data=data_grouped)
plt.title('Total Number of Goals Scored by Players of a Country')
plt.xlabel('Name of Country')
plt.ylabel('Number of Goals')

for p in goalplot.patches:
    goalplot.annotate(format(p.get_height(), '.1f'),
                      (p.get_x() + p.get_width() / 2., p.get_height()),
                      ha='center', va='center',
                      xytext=(0, 10),
                      textcoords='offset points')

plt.xticks(rotation=90)
plt.show()
```



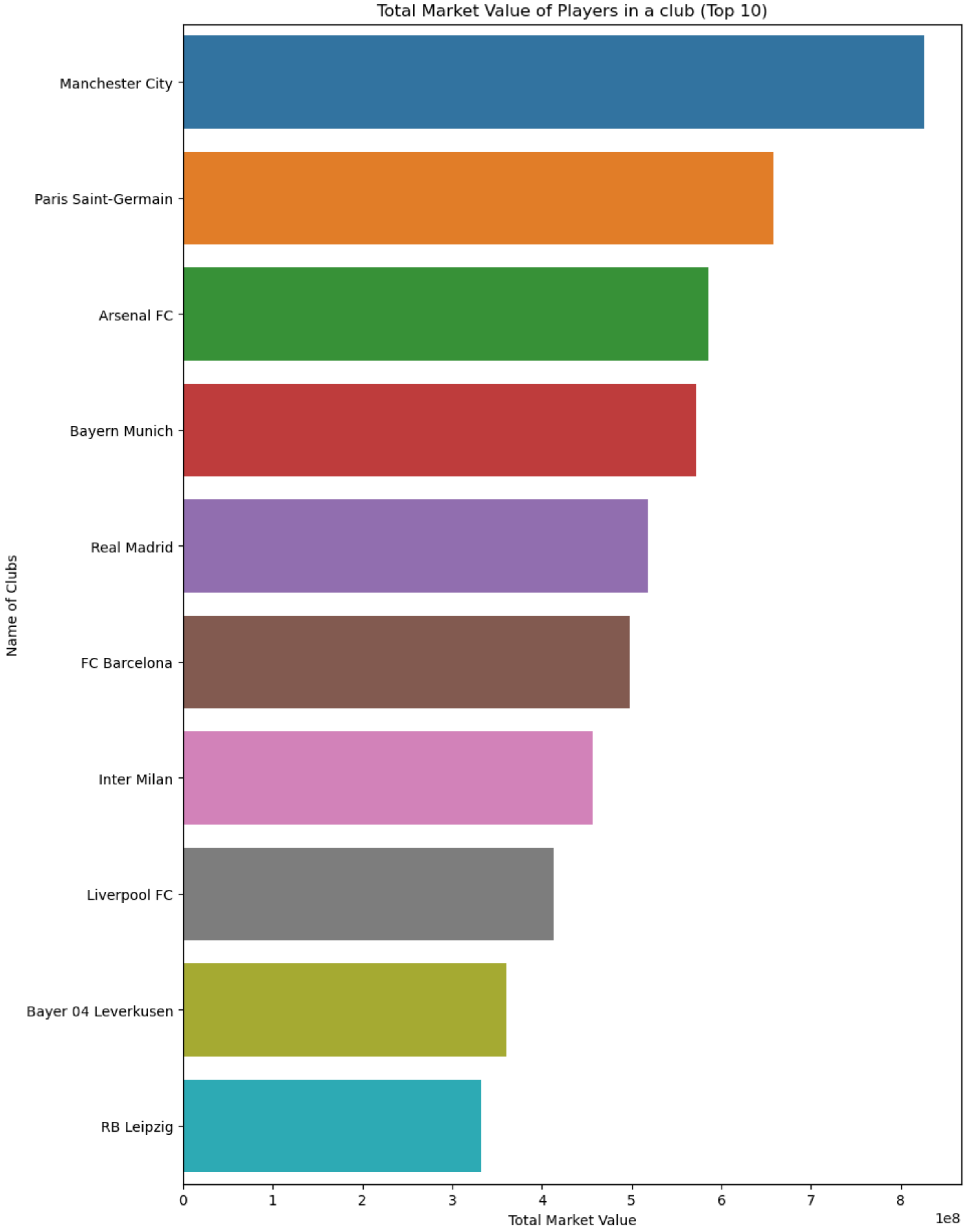
In [24]:

```
# Define vertical Bar Graph using matplotlib
data_grouped = data.groupby('Country').sum().reset_index()
plt.figure(figsize=(15, 10))


goalplot2 = sns.barplot(x='Country', y='MarketValue', data=data_grouped)
plt.title('Total Market Value of Players by Country')
plt.xlabel('Country')
plt.ylabel('Total Market Value')


for p in goalplot2.patches:
    goalplot2.annotate(format(p.get_height(), '.1f'),
                       (p.get_x() + p.get_width() / 2., p.get_height()),
                       ha='center', va='center',
                       xytext=(0, 10), textcoords='offset points')
```

```
plt.xticks(rotation=90)
plt.show()
```

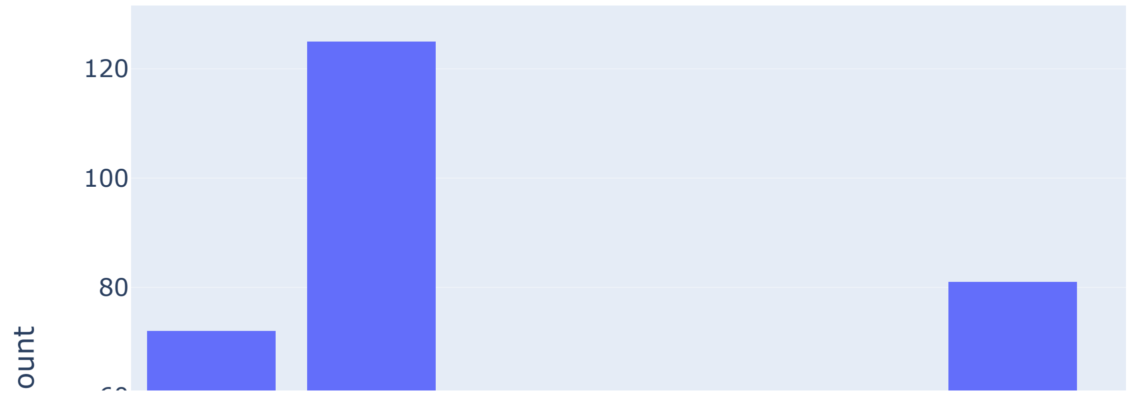


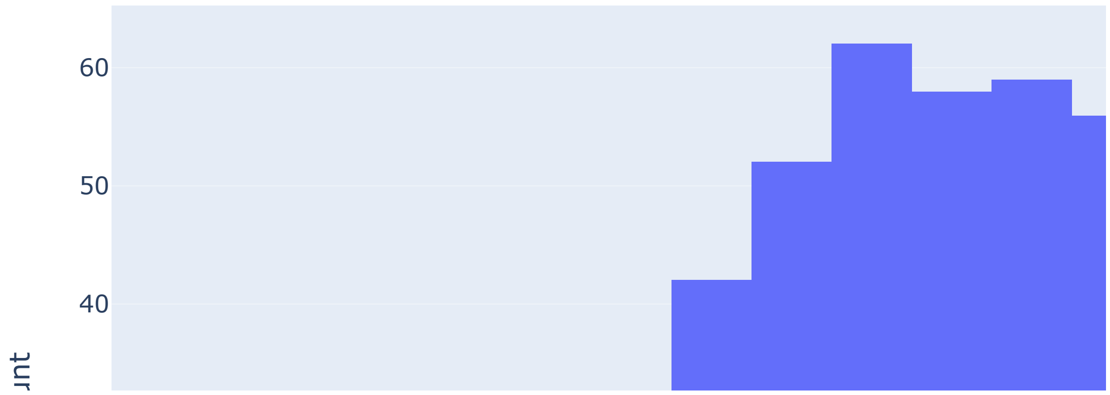
In [25]:
```
# Define horizontal Bar Graph using matplotlib
data_group = data.groupby('Club').sum().reset_index()
top_10_clubs = data_group.sort_values(by='MarketValue', ascending=False).head(10)
plt.figure(figsize=(10, 15))
sns.barplot(x='MarketValue',y='Club',data = top_10_clubs)
plt.title('Total Market Value of Players in a club (Top 10)')
plt.xlabel('Total Market Value')
plt.ylabel('Name of Clubs')
plt.show()
```

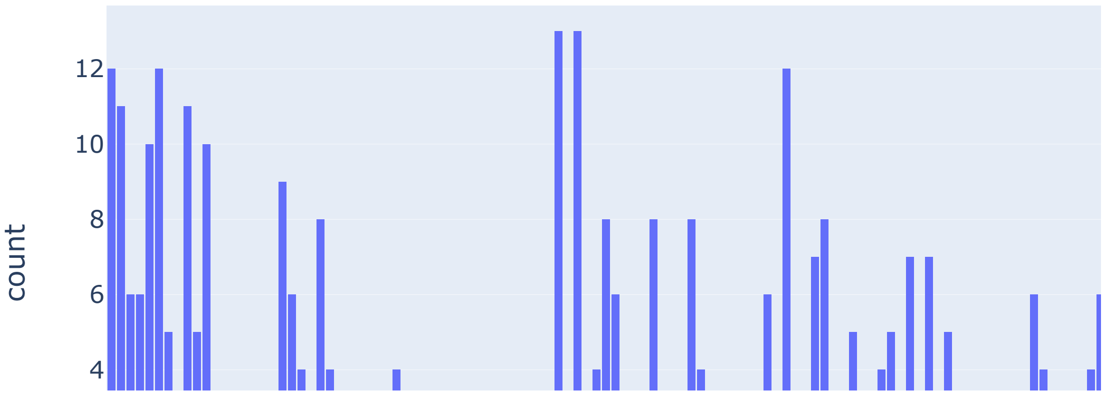


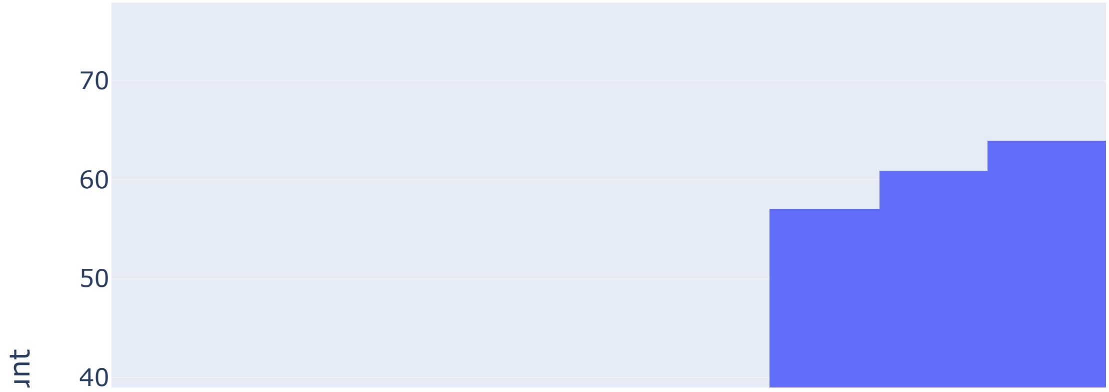
In [1]:

```
# import different types of Bar graphs using Python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from plotly import express

data= pd.read_csv('euro2024_players.csv')
for column in ['Name',
 'Position',
 'Age',
 'Club',
 'Height',
 'Foot',
 'Caps',
 'Goals',
 'MarketValue',
 'Country']:
    express.histogram(data_frame=data, x=column).show()
```

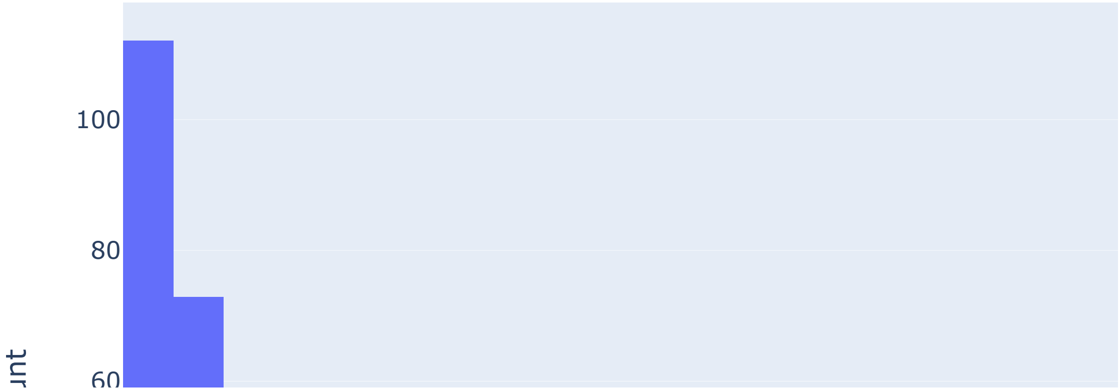

120
100
80
ount

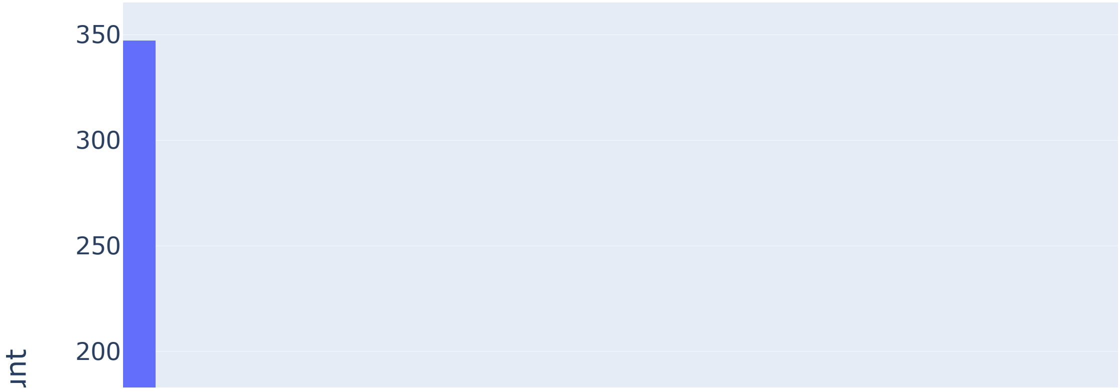

60
50
40
unt

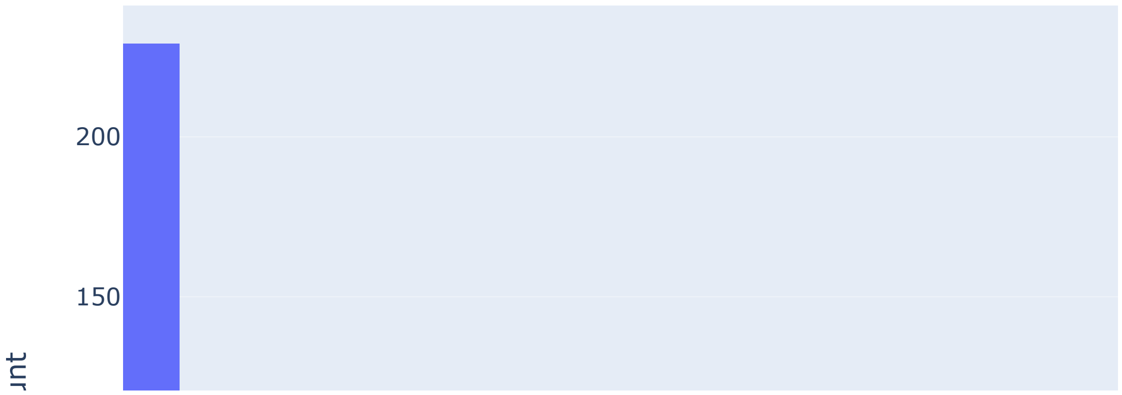

count
12
10
8
6
4

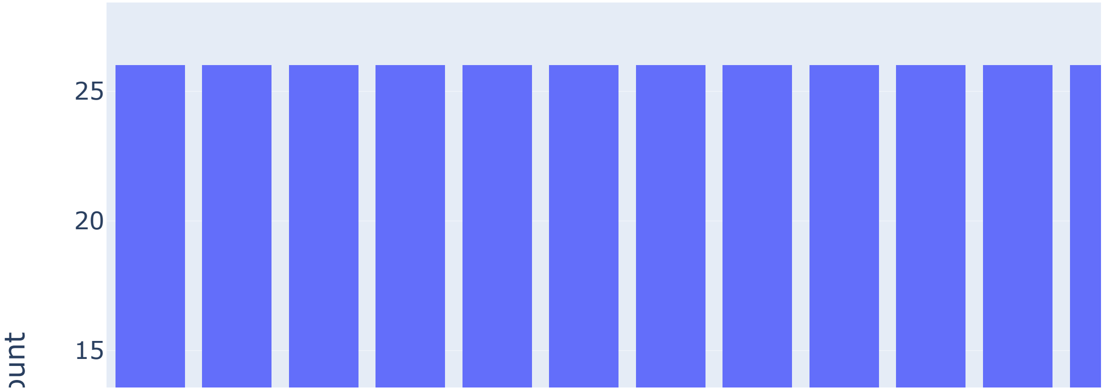

70
60
50
40
unt

450
400
350
300
250
unt


100
80
60
unt

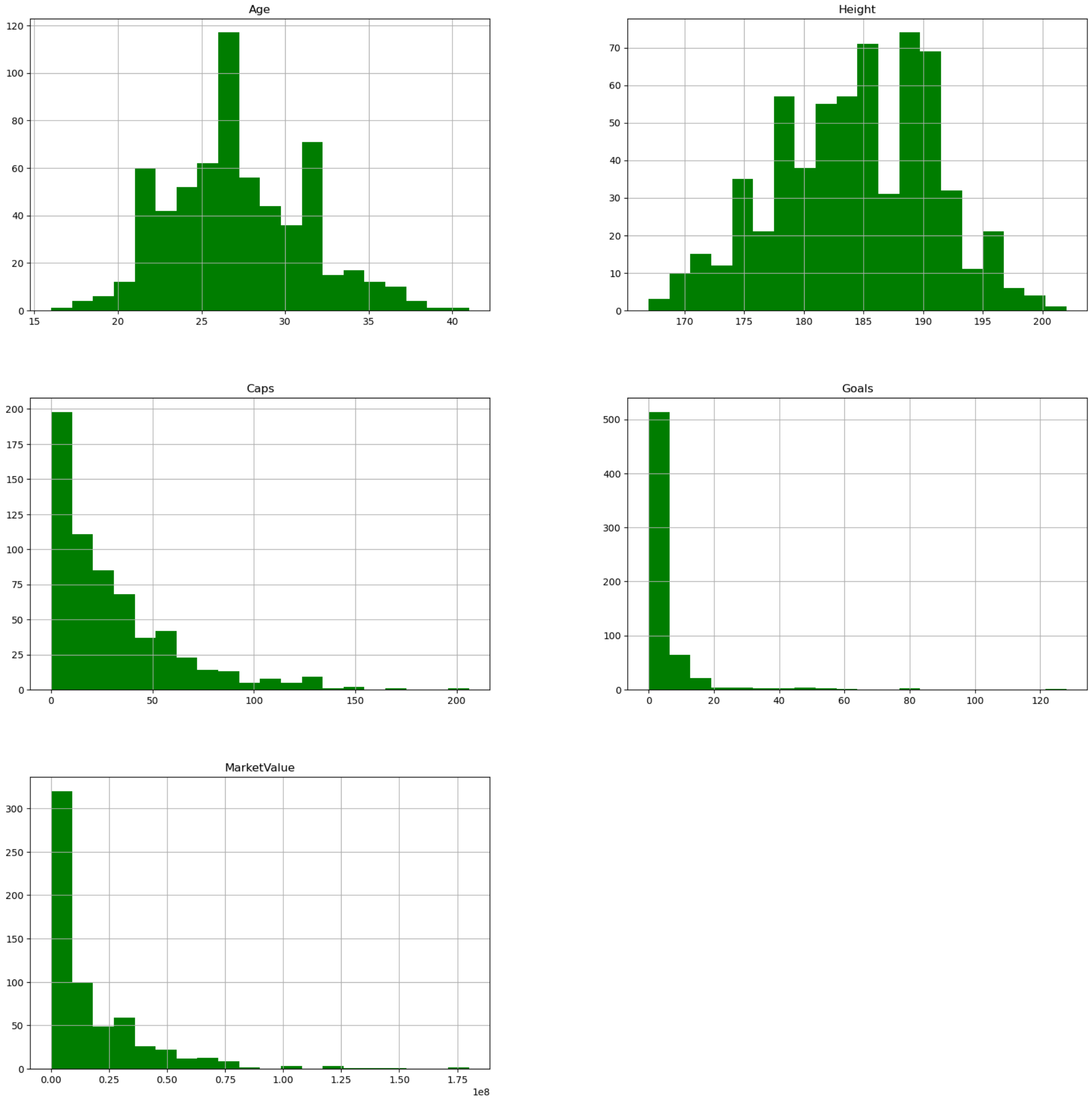

350
300
250
200
unt


200
150
unt



```
In [ ]:
```

```
In [27]: # import different types of Bar graphs using Python
         data.hist(bins = 20, figsize = (20,20), color = 'green')
         plt.show()
```



title

In [13]:

```
# from IPython.display import Image
Image(filename="Euro cup 2024.py.jpg",width=1000,height=400)
```

```
Out[13]:
```

THANK YOU SO MUCH !!!!!