# Data Analysis and Visualisation in Python

In this project, we used Python to extract and analyse specific data from the 'GDP (Nominal) per Capita.csv' dataset. Our first step was to load the data from the CSV file into a DataFrame named "df" using Colab notebook. To quickly explore the dataset, we printed the first 10 rows and the last 5 rows, providing a snapshot of the data. Additionally, we identified the country with the highest UN_Estimate value and examined essential columns, including 'Country/Territory' and 'UN_Region', which helped us better understand the data's structure and attributes.

Throughout the project, we made use of various Python functionalities. For instance, quotes in Python (single, double, or triple quotes) were used to define string values, while escape characters were used to handle special characters within strings. We used print() commands to display output, with display() being useful for a more formatted output in Colab, while print() and show() provided simpler forms of output to the console.

We also explored important positional arguments to define the order of parameters in functions, which helped streamline our functions. Furthermore, we utilised algorithms and arithmetic operators for data manipulation and analysis, along with shortcut operators for efficient coding. Conditional statements allowed us to perform operations based on specific conditions, such as filtering countries with MF_Estimate values below the average.

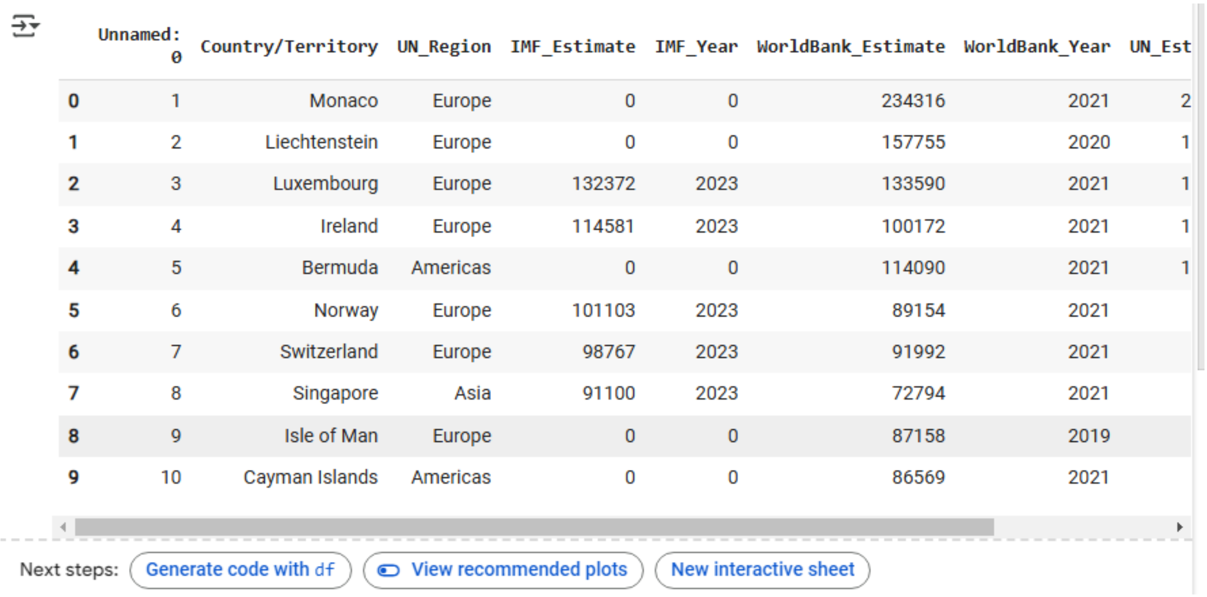
To assist with the visual analysis of the data, we employed several visualisation techniques, including bar plots, scatter plots, and box plots. These visualisations were instrumental in uncovering patterns, distributions, and relationships within the data, allowing us to gain deeper insights.

Additionally, we used Python's correlation function to explore the relationships between the estimates provided by the World Bank, UN, and IMF. By analysing these correlations, we were able to understand how the estimates from these organisations align or differ, providing valuable context for the dataset.

**Please download the 'GDP (nominal) per Capita.csv' dataset here.**

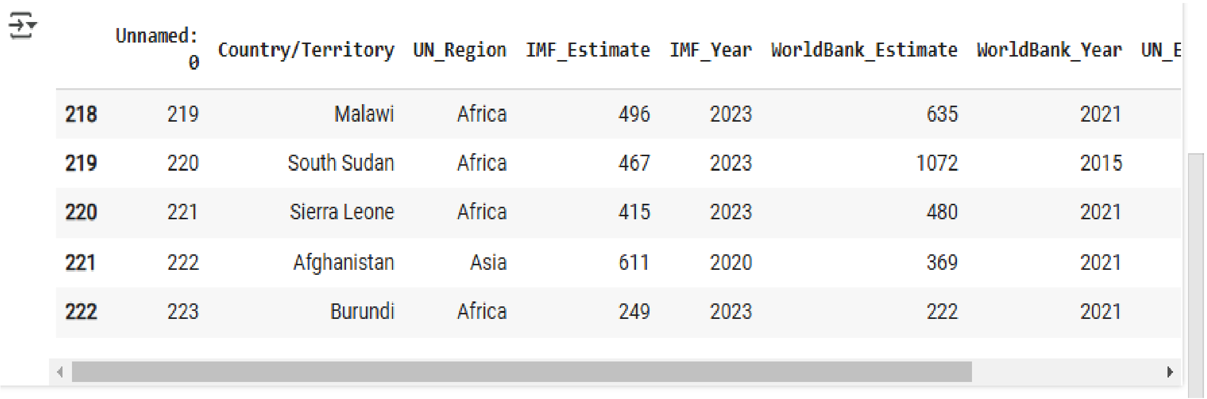
```
# Read and save the ‘GDP (nominal) per Capita’ data to a data frame called “df” in Colab notebook.
df = pd.read_csv('GDP (nominal) per Capita.csv')
df.to_csv('GDP (nominal) per Capita.csv', index=False)
```

```
#Print the first 10 rows
df.head(10)
```



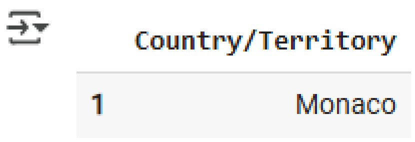
| | Unnamed: 0 | Country/Territory | UN_Region | IMF_Estimate | IMF_Year | WorldBank_Estimate | WorldBank_Year | UN_Est |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | Monaco | Europe | 0 | 0 | 234316 | 2021 | 2 |
| 1 | 2 | Liechtenstein | Europe | 0 | 0 | 157755 | 2020 | 1 |
| 2 | 3 | Luxembourg | Europe | 132372 | 2023 | 133590 | 2021 | 1 |
| 3 | 4 | Ireland | Europe | 114581 | 2023 | 100172 | 2021 | 1 |
| 4 | 5 | Bermuda | Americas | 0 | 0 | 114090 | 2021 | 1 |
| 5 | 6 | Norway | Europe | 101103 | 2023 | 89154 | 2021 | |
| 6 | 7 | Switzerland | Europe | 98767 | 2023 | 91992 | 2021 | |
| 7 | 8 | Singapore | Asia | 91100 | 2023 | 72794 | 2021 | |
| 8 | 9 | Isle of Man | Europe | 0 | 0 | 87158 | 2019 | |
| 9 | 10 | Cayman Islands | Americas | 0 | 0 | 86569 | 2021 | |

```
#Print the last 5 rows
df.tail()
```



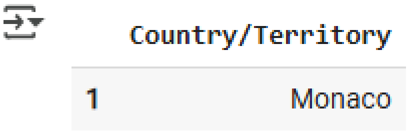
| | Unnamed: 0 | Country/Territory | UN_Region | IMF_Estimate | IMF_Year | WorldBank_Estimate | WorldBank_Year | UN_E |
|---|---|---|---|---|---|---|---|---|
| 218 | 219 | Malawi | Africa | 496 | 2023 | 635 | 2021 | |
| 219 | 220 | South Sudan | Africa | 467 | 2023 | 1072 | 2015 | |
| 220 | 221 | Sierra Leone | Africa | 415 | 2023 | 480 | 2021 | |
| 221 | 222 | Afghanistan | Asia | 611 | 2020 | 369 | 2021 | |
| 222 | 223 | Burundi | Africa | 249 | 2023 | 222 | 2021 | |

```
#Country has highest UN_Estimate
gdp=df[df["UN_Estimate"]==df["UN_Estimate"].max()]
gdp['Country/Territory']
```



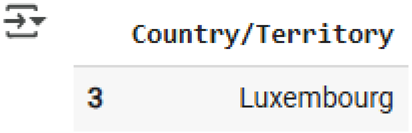
| | Country/Territory |
|---|---|
| 1 | Monaco |

```
#Country has highest Worlbank Estimate
gdp=df[df["WorldBank_Estimate"]==df["WorldBank_Estimate"].max()]
gdp['Country/Territory']
```



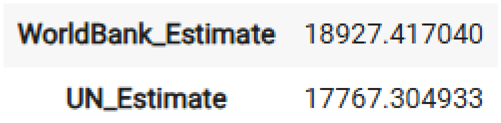
```
   Country/Territory
1             Monaco
```

```
#Country has highest IMF Estimate
gdp=df[df["IMF_Estimate"]==df["IMF_Estimate"].max()]
gdp['Country/Territory']
```



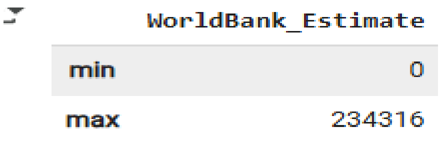
```
   Country/Territory
3         Luxembourg
```

```
# Calculate the average of 'Worldbank_Estimate' and 'UN_Estimate' columns
avg_worldbank_UN = df[['WorldBank_Estimate', 'UN_Estimate']].mean()
avg_worldbank_UN
```



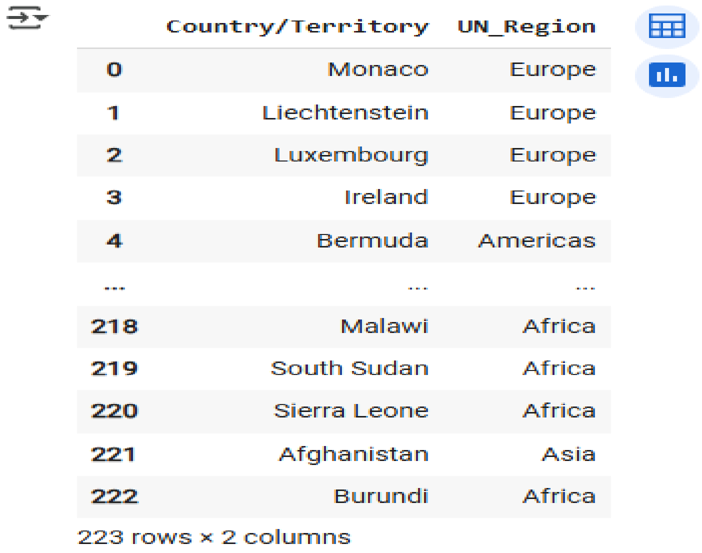
```
WorldBank_Estimate   18927.417040
       UN_Estimate   17767.304933
```

```
# Minimum and maximum of 'Worldbank_Estimate'
df["WorldBank_Estimate"].agg(["min","max"])
```



```
         WorldBank_Estimate
min                       0
max                  234316
```

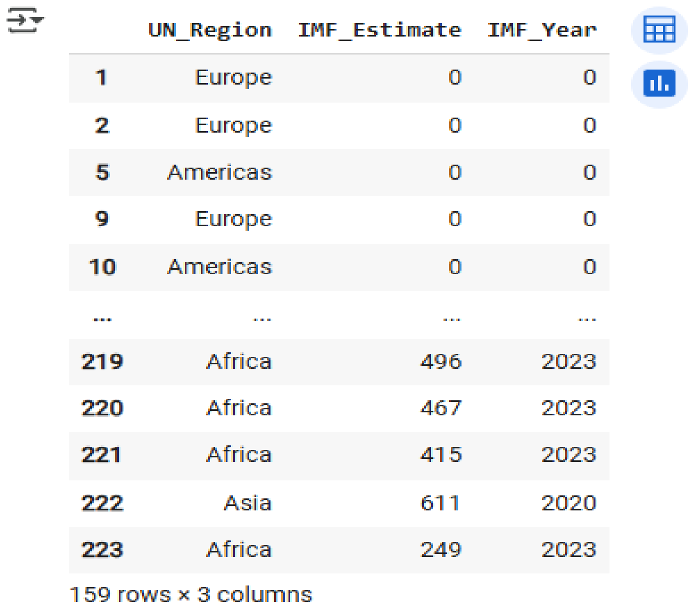
```
# Print ‘Country/Territory’ and ‘UN_Region’ columns
df[['Country/Territory', 'UN_Region']]
```



|  | Country/Territory | UN_Region |
|---|---|---|
| 0 | Monaco | Europe |
| 1 | Liechtenstein | Europe |
| 2 | Luxembourg | Europe |
| 3 | Ireland | Europe |
| 4 | Bermuda | Americas |
| ... | ... | ... |
| 218 | Malawi | Africa |
| 219 | South Sudan | Africa |
| 220 | Sierra Leone | Africa |
| 221 | Afghanistan | Asia |
| 222 | Burundi | Africa |

223 rows × 2 columns

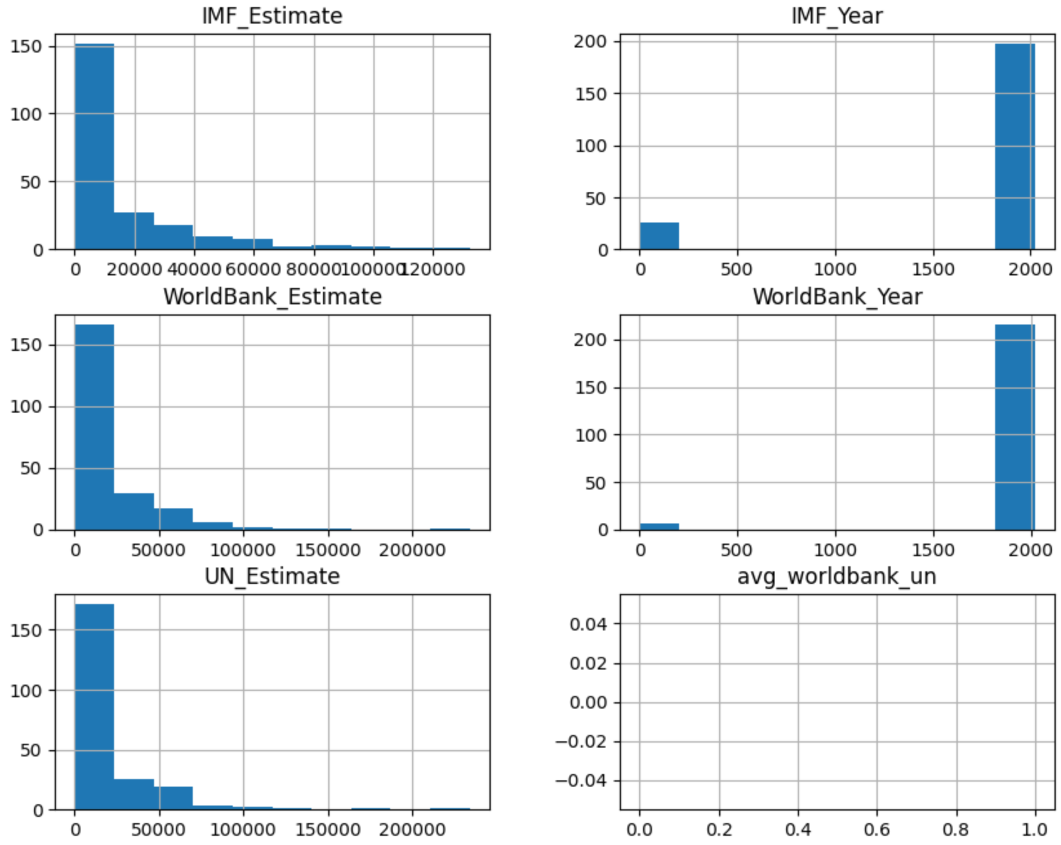
```
# Countries below average by IMF_Estimate
gdp=df[df["IMF_Estimate"]<df["IMF_Estimate"].mean()]
gdp.iloc[:,1:4]
```



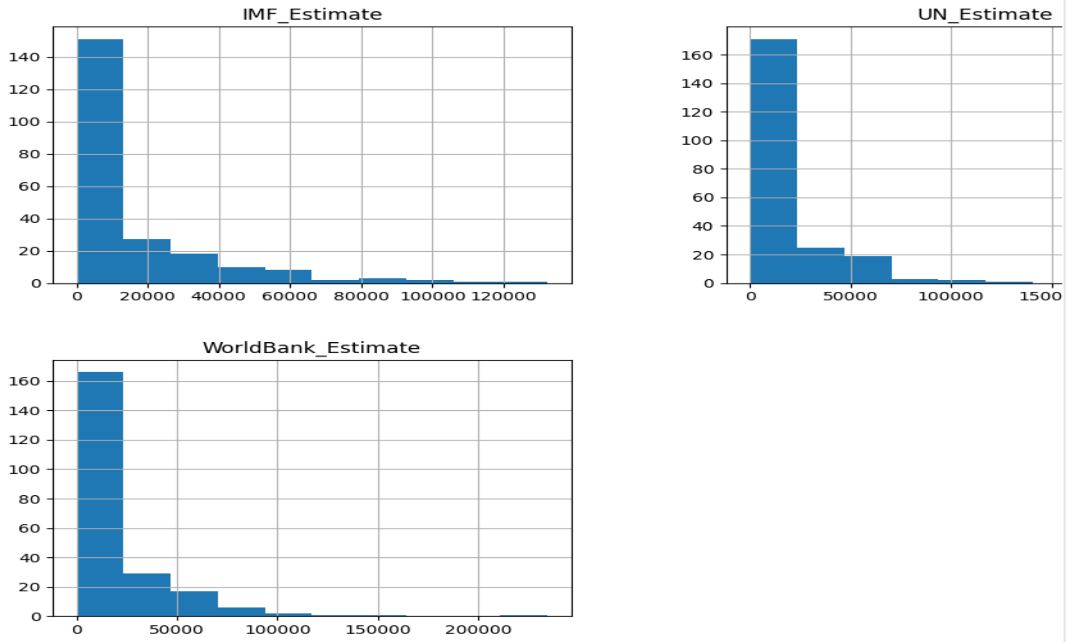
|  | UN_Region | IMF_Estimate | IMF_Year |
|---|---|---|---|
| 1 | Europe | 0 | 0 |
| 2 | Europe | 0 | 0 |
| 5 | Americas | 0 | 0 |
| 9 | Europe | 0 | 0 |
| 10 | Americas | 0 | 0 |
| ... | ... | ... | ... |
| 219 | Africa | 496 | 2023 |
| 220 | Africa | 467 | 2023 |
| 221 | Africa | 415 | 2023 |
| 222 | Asia | 611 | 2020 |
| 223 | Africa | 249 | 2023 |

159 rows × 3 columns

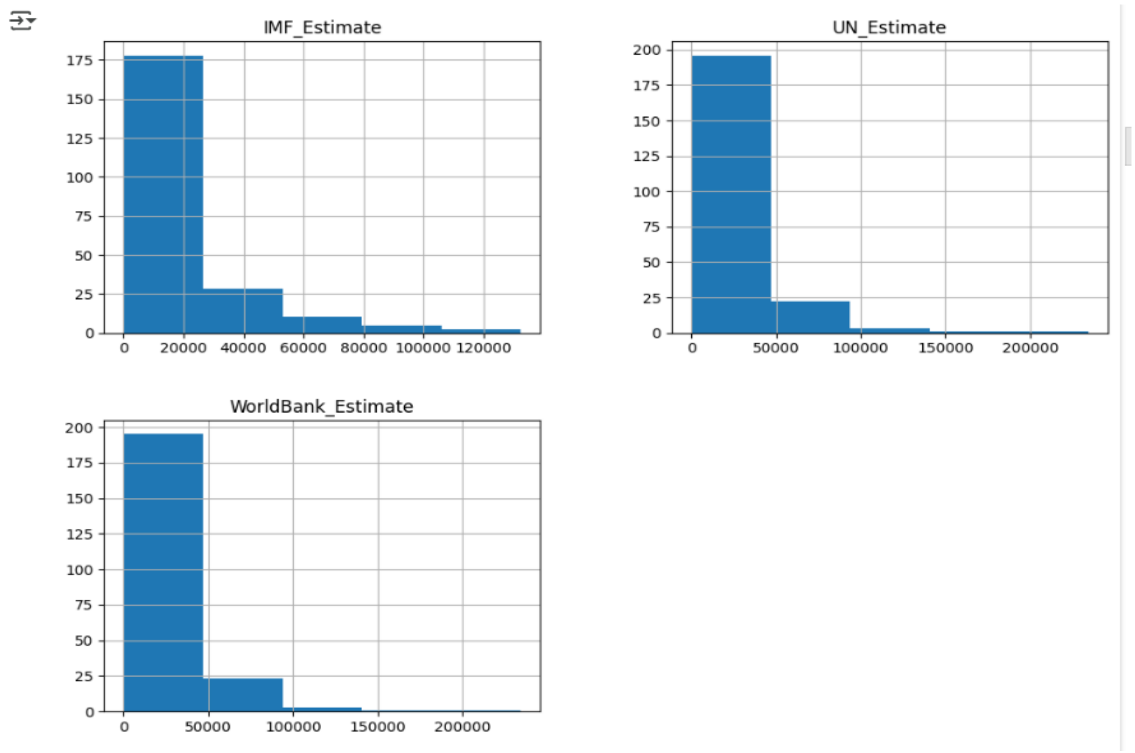
```
#Histogram
df.hist(figsize=(10,8))
plt.show()
```



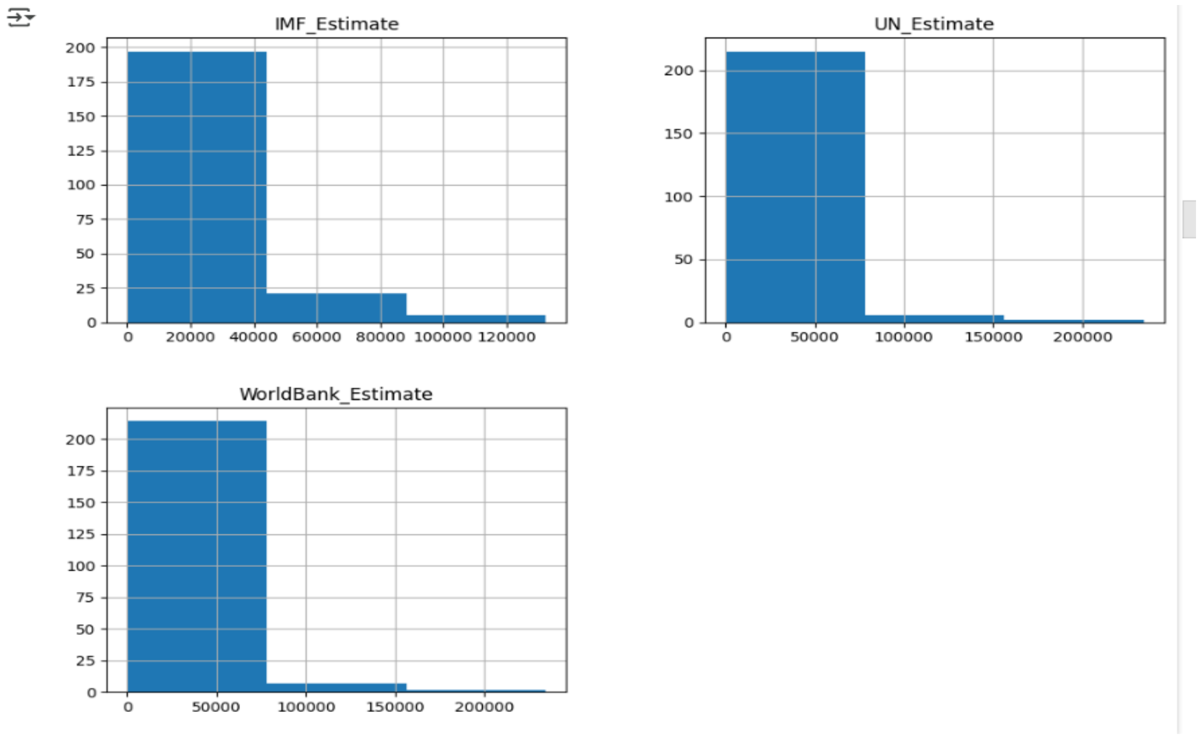
```
df[["IMF_Estimate", "UN_Estimate", "WorldBank_Estimate"]].hist(figsize=(12,9))
plt.show()
```



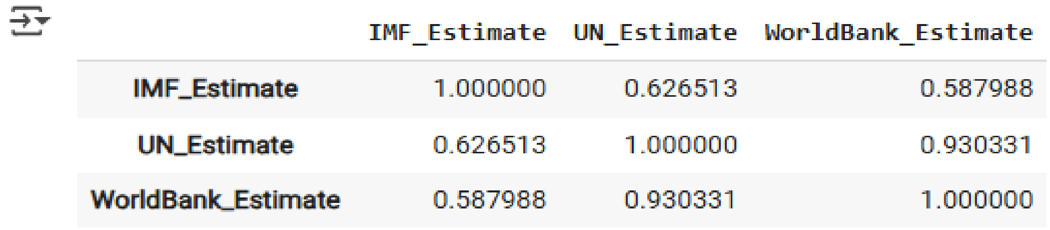
```
df[["IMF_Estimate", "UN_Estimate", "WorldBank_Estimate"]].hist(bins=5,
figsize=(12,9))
plt.show()
```



```
df[["IMF_Estimate", "UN_Estimate", "WorldBank_Estimate"]].hist(bins=3,
figsize=(12,9))
plt.show()
```

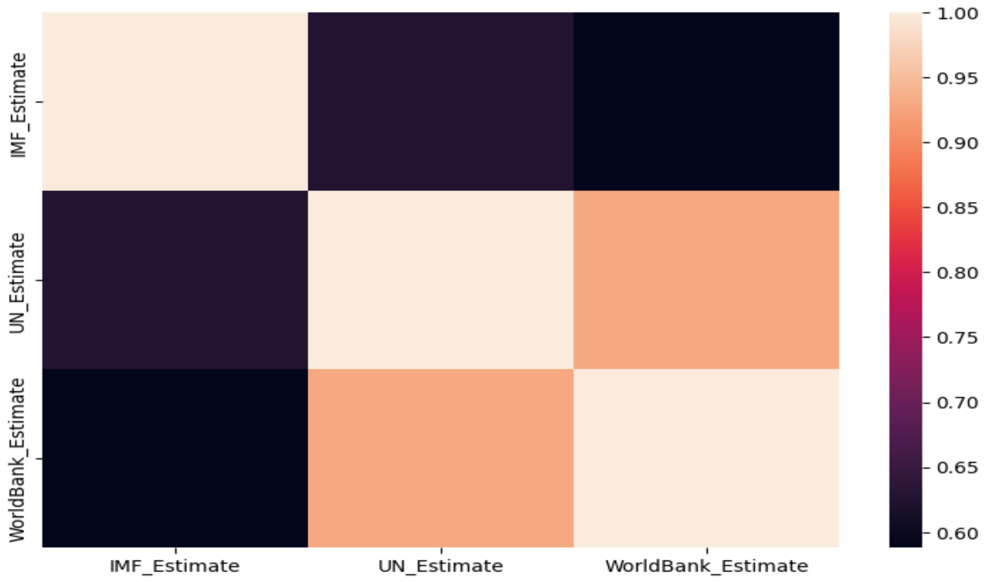


```
# Correlation
df[["IMF_Estimate", "UN_Estimate", "WorldBank_Estimate"]].corr()
```

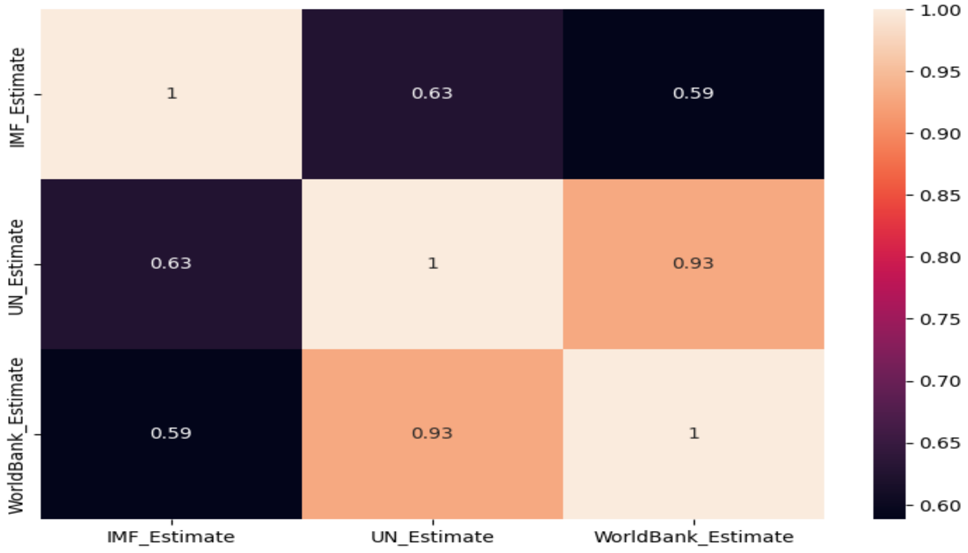


| | IMF_Estimate | UN_Estimate | WorldBank_Estimate |
|---|---|---|---|
| **IMF_Estimate** | 1.000000 | 0.626513 | 0.587988 |
| **UN_Estimate** | 0.626513 | 1.000000 | 0.930331 |
| **WorldBank_Estimate** | 0.587988 | 0.930331 | 1.000000 |

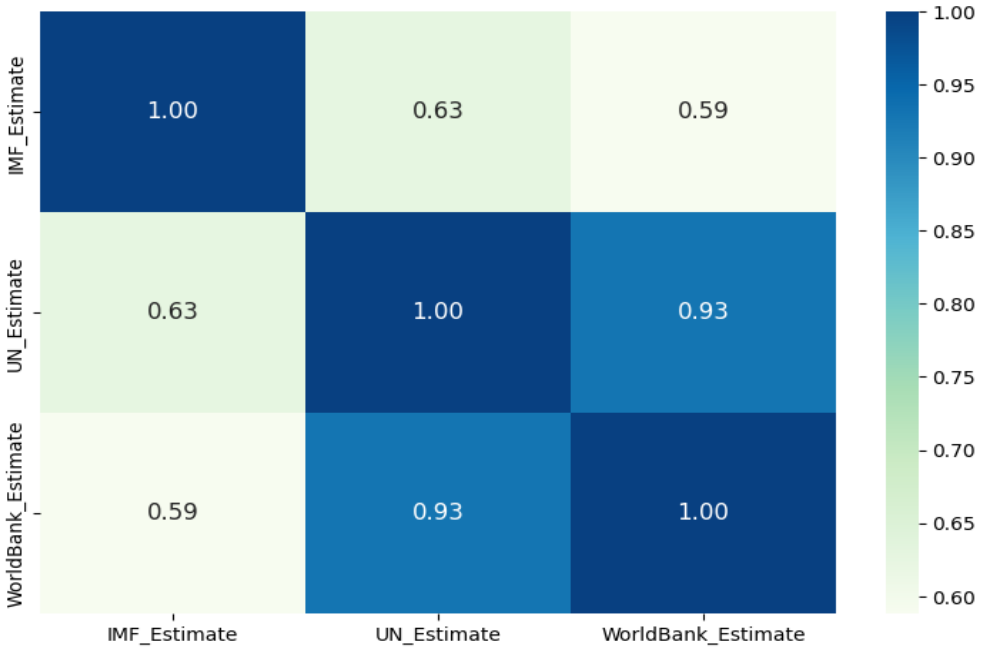
```
corr = df[["IMF_Estimate", "UN_Estimate", "WorldBank_Estimate"]].corr()
plt.figure(figsize=(9,6))
sns.heatmap(corr)
plt.show()
```



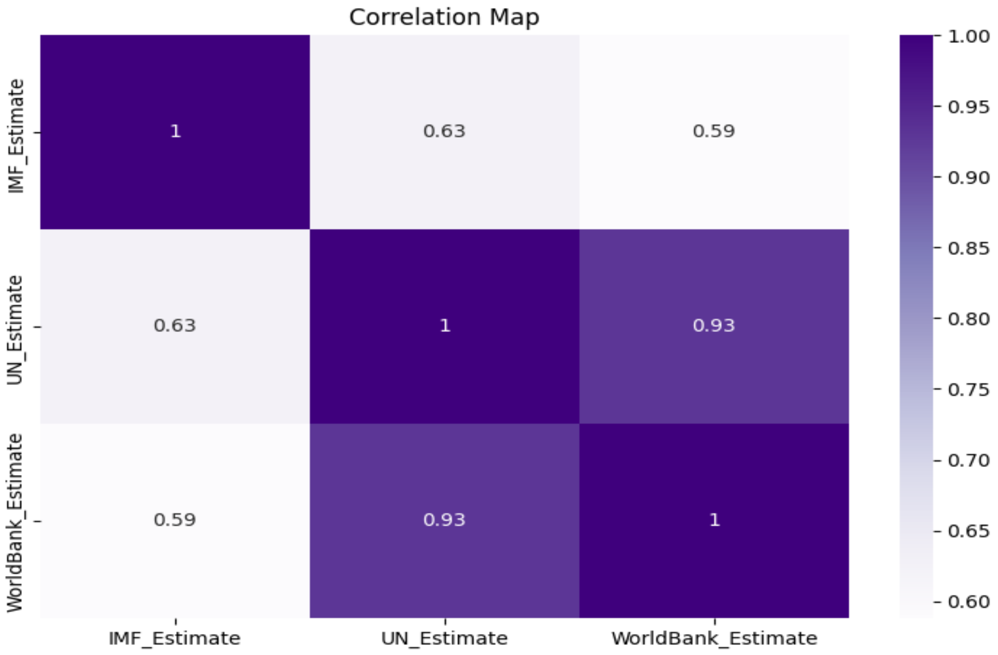
```
corr = df[["IMF_Estimate", "UN_Estimate", "WorldBank_Estimate"]].corr()
plt.figure(figsize=(9,6))
sns.heatmap(corr, annot=True)
plt.show()
```



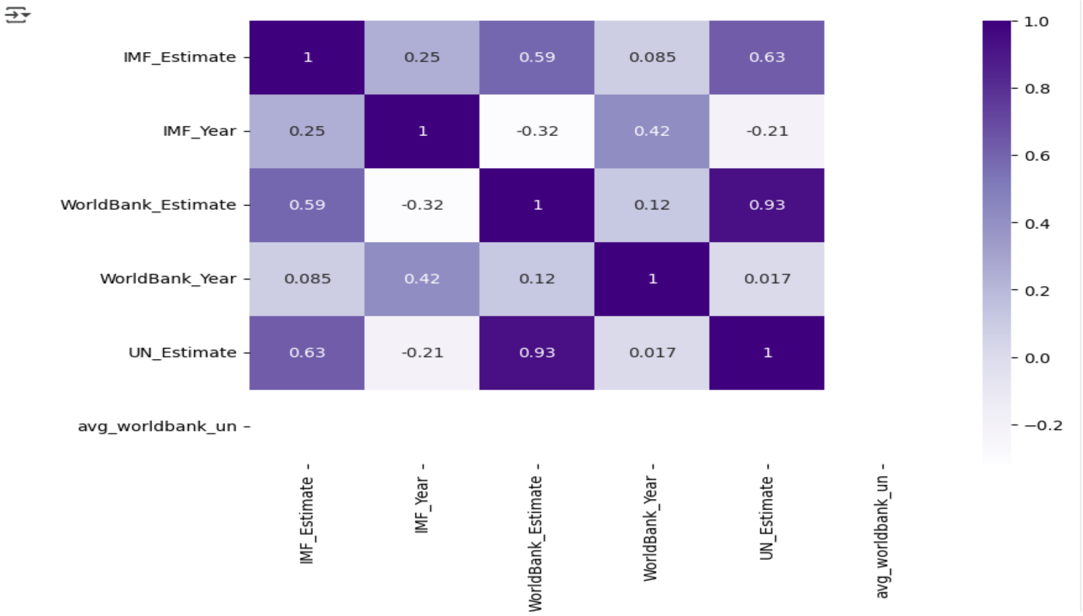
```
corr = df[["IMF_Estimate", "UN_Estimate", "WorldBank_Estimate"]].corr()
plt.figure(figsize=(9,6))
sns.heatmap(corr, annot=True, fmt=".2f", cmap = 'GnBu', annot_kws={"size": 12})
plt.show()
```



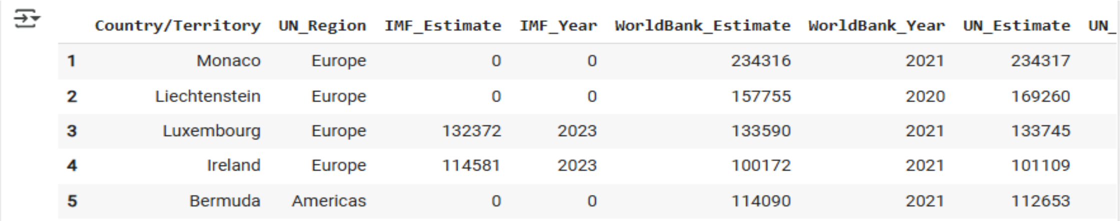
```
corr = df[["IMF_Estimate", "UN_Estimate", "WorldBank_Estimate"]].corr()
plt.figure(figsize=(9,6))
sns.heatmap(corr, annot=True, cmap = 'Purples')
plt.title("Correlation Map")
plt.show()
```



```
corr = df.select_dtypes(include=[int, float]).corr()
plt.figure(figsize=(9,6))
sns.heatmap(corr, annot=True, cmap = 'Purples')
plt.show()
```

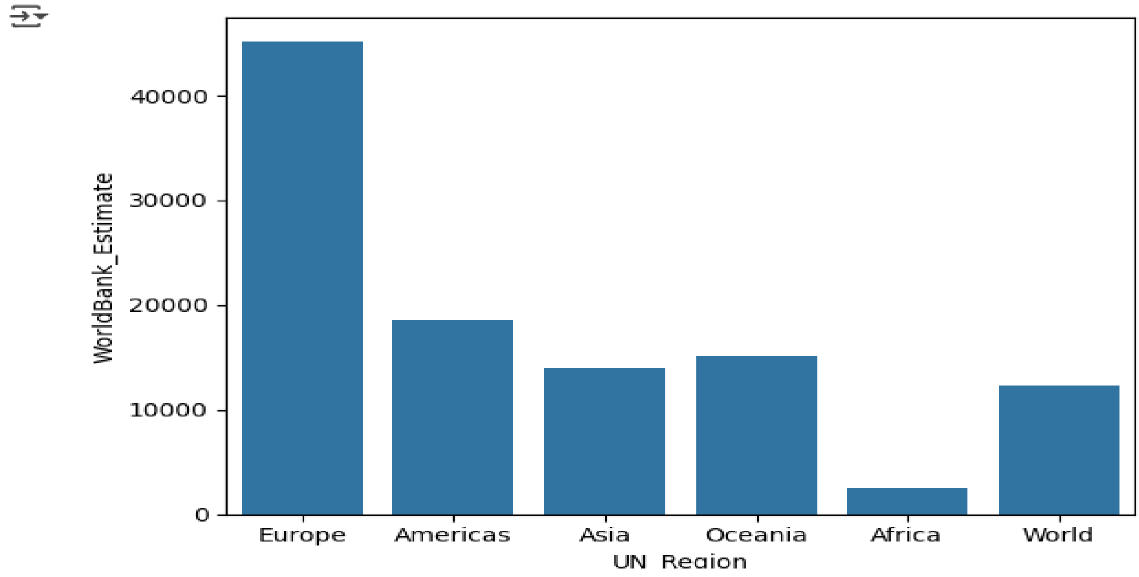


```
#Print the first 5 rows
df.head()
```

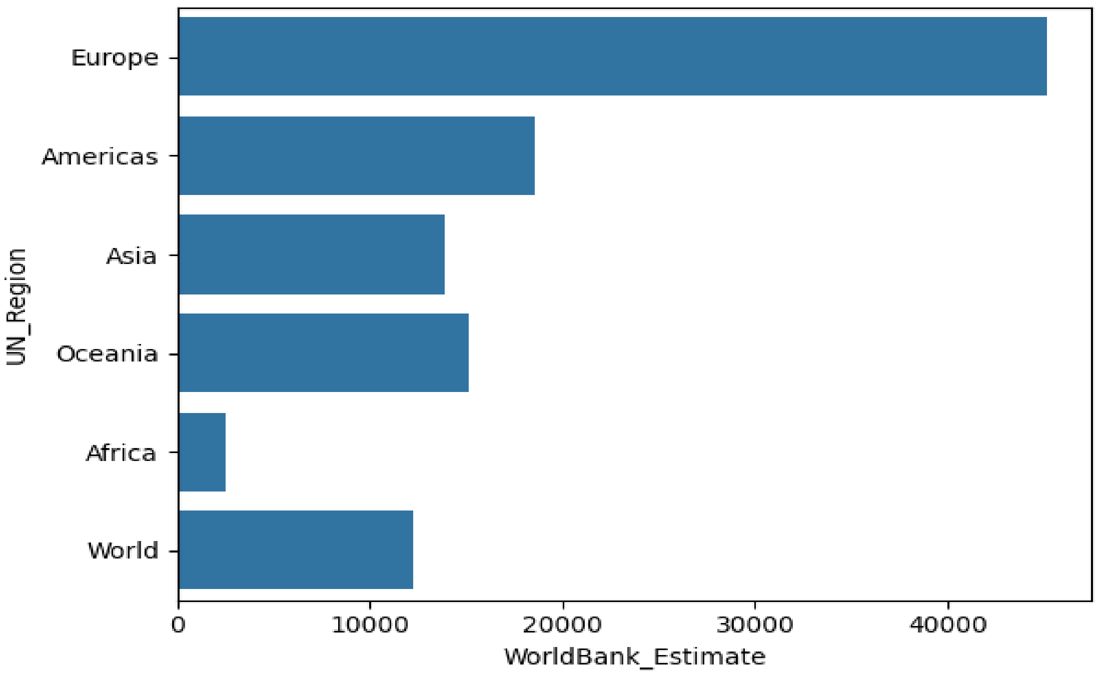


| | Country/Territory | UN_Region | IMF_Estimate | IMF_Year | WorldBank_Estimate | WorldBank_Year | UN_Estimate | UN_ |
|---|---|---|---|---|---|---|---|---|
| 1 | Monaco | Europe | 0 | 0 | 234316 | 2021 | 234317 | |
| 2 | Liechtenstein | Europe | 0 | 0 | 157755 | 2020 | 169260 | |
| 3 | Luxembourg | Europe | 132372 | 2023 | 133590 | 2021 | 133745 | |
| 4 | Ireland | Europe | 114581 | 2023 | 100172 | 2021 | 101109 | |
| 5 | Bermuda | Americas | 0 | 0 | 114090 | 2021 | 112653 | |

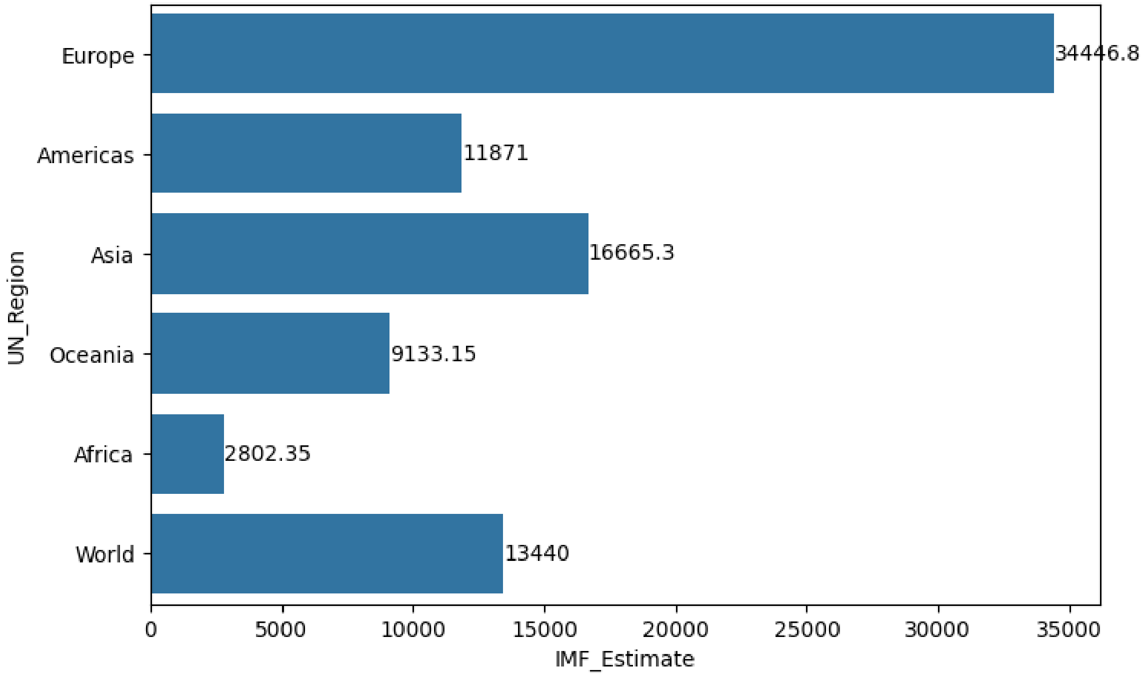
```
sns.barplot(x="UN_Region", y="WorldBank_Estimate", data=df, errorbar=None)
plt.show()
```



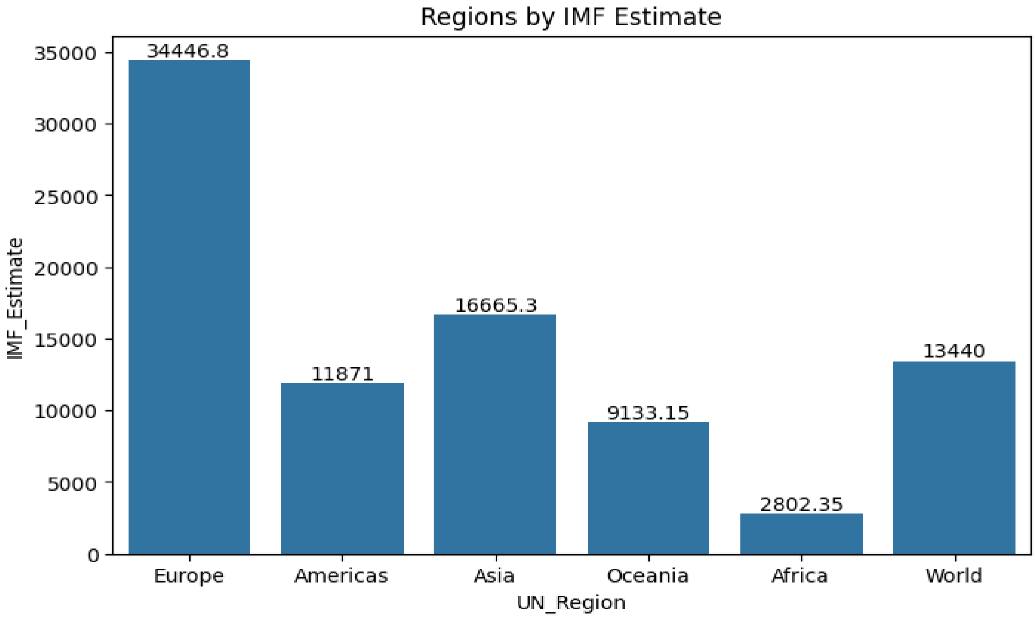
```
sns.barplot(x="WorldBank_Estimate", y="UN_Region", data=df, errorbar=None)
plt.show()
```



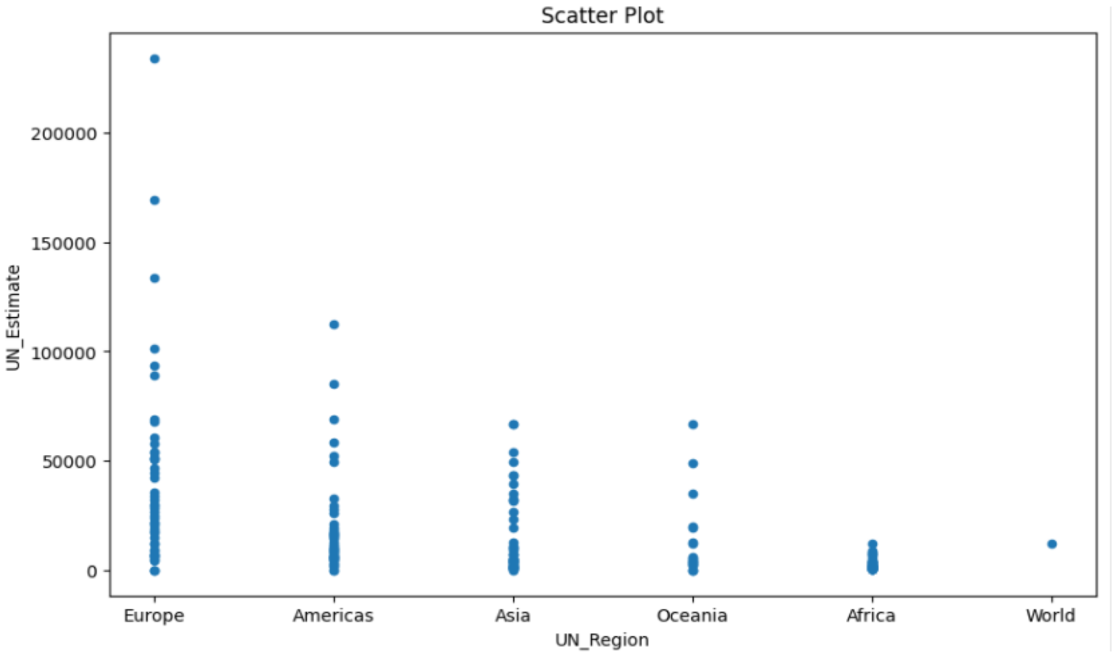
```
fig = plt.figure(figsize = (8,5))
ax = sns.barplot(x = "IMF_Estimate",  y = "UN_Region",
data = df, errorbar = None)
ax.bar_label(ax.containers[0])
plt.show()
```



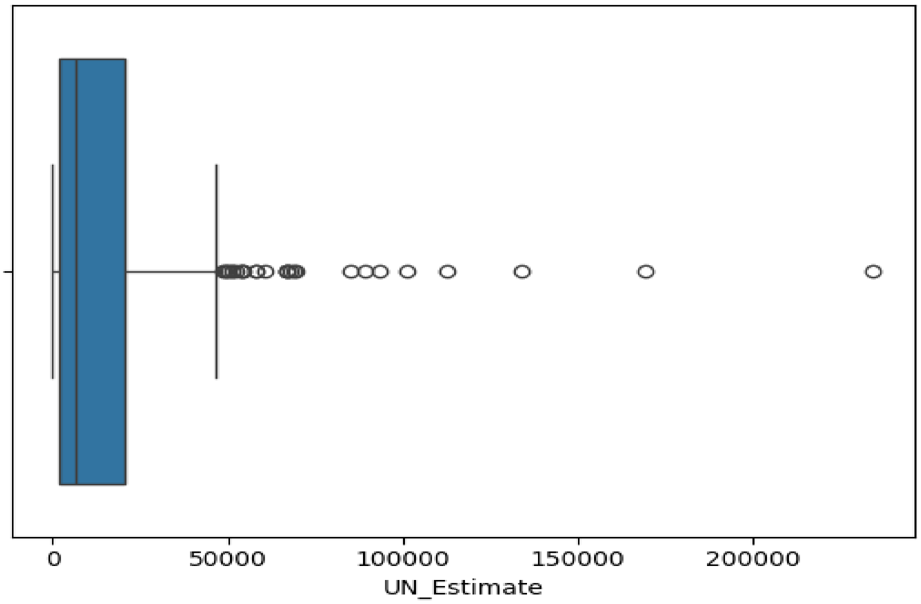
```
fig = plt.figure(figsize = (8,5))
ax = sns.barplot(x = "UN_Region",  y = "IMF_Estimate",
              data = df, errorbar = None)
ax.bar_label(ax.containers[0])
ax.set_title("Regions by IMF Estimate")
plt.show()
```



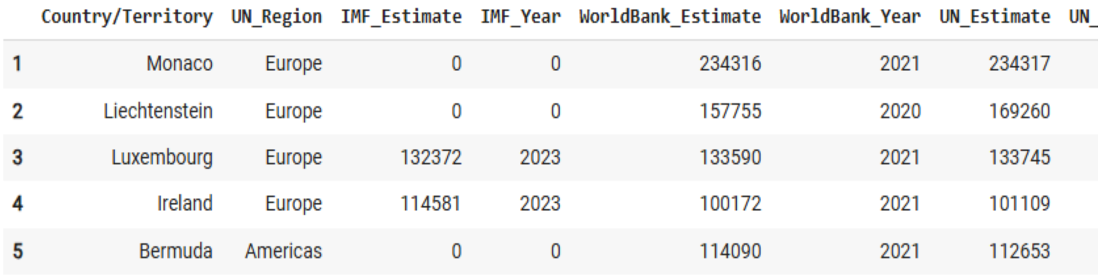
```
#Scatter Plot
df.plot(x='UN_Region', y='UN_Estimate', kind='scatter',
       figsize=(10,6),
       title="Scatter Plot")
plt.show()
```



```
#Boxplot
sns.boxplot(x=df["UN_Estimate"])
plt.show()
```

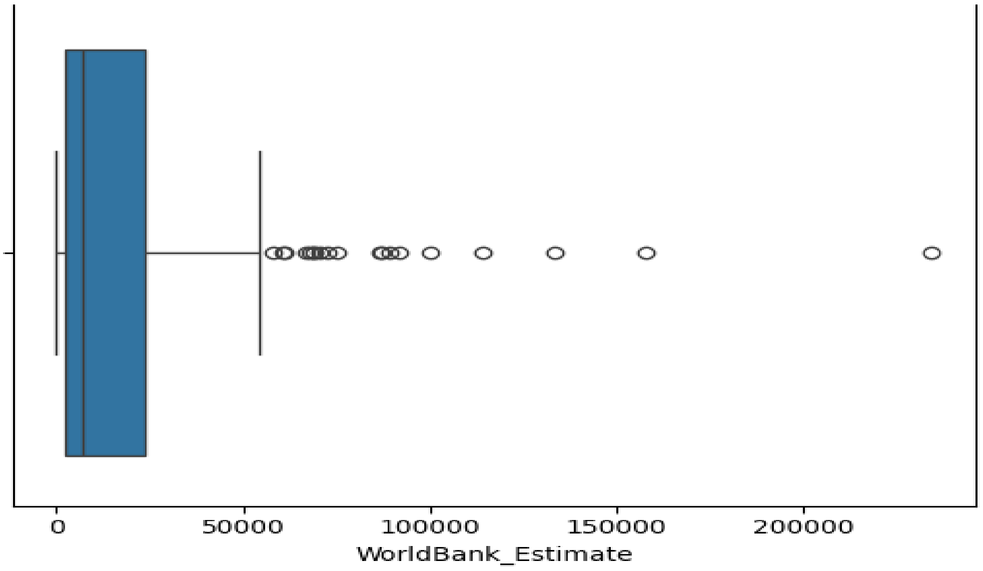


```
df[df["UN_Estimate"]>50000].head()
```

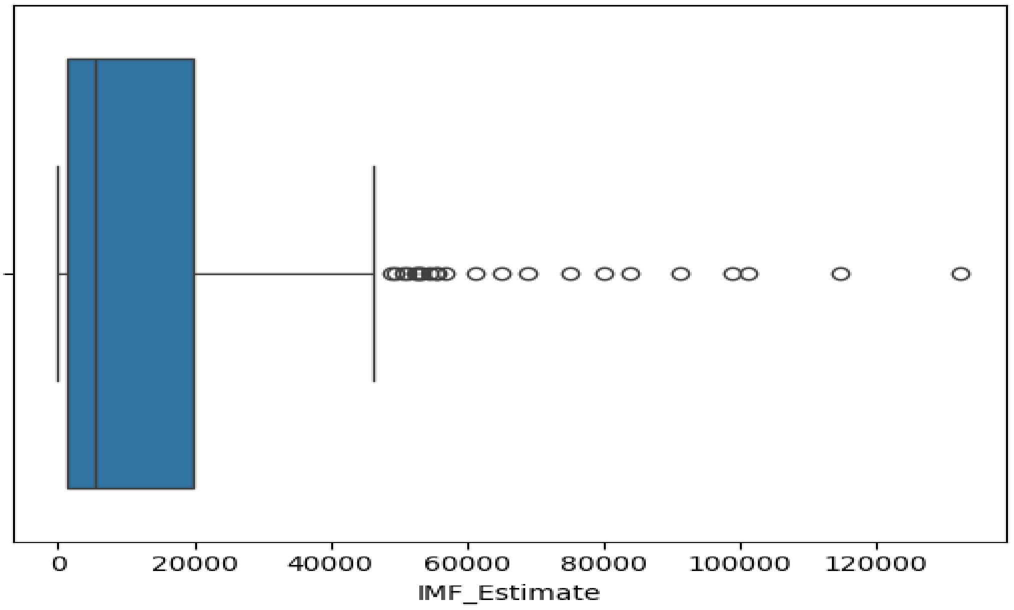


| | Country/Territory | UN_Region | IMF_Estimate | IMF_Year | WorldBank_Estimate | WorldBank_Year | UN_Estimate | UN_ |
|---|---|---|---|---|---|---|---|---|
| 1 | Monaco | Europe | 0 | 0 | 234316 | 2021 | 234317 | |
| 2 | Liechtenstein | Europe | 0 | 0 | 157755 | 2020 | 169260 | |
| 3 | Luxembourg | Europe | 132372 | 2023 | 133590 | 2021 | 133745 | |
| 4 | Ireland | Europe | 114581 | 2023 | 100172 | 2021 | 101109 | |
| 5 | Bermuda | Americas | 0 | 0 | 114090 | 2021 | 112653 | |

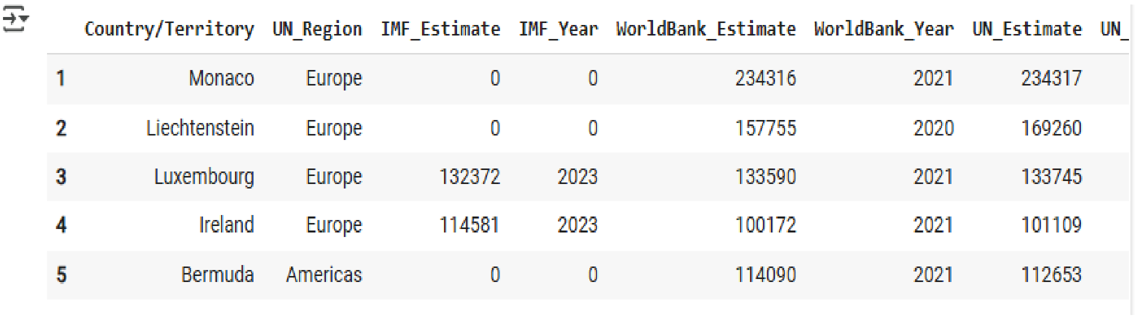
```
sns.boxplot(x=df["WorldBank_Estimate"])
plt.show()
```



```
sns.boxplot(x=df["IMF_Estimate"])
plt.show()
```

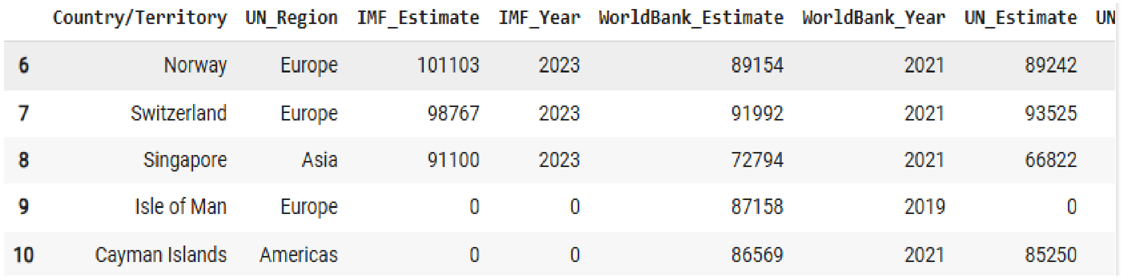


```
df[df["UN_Estimate"]>100000]
```

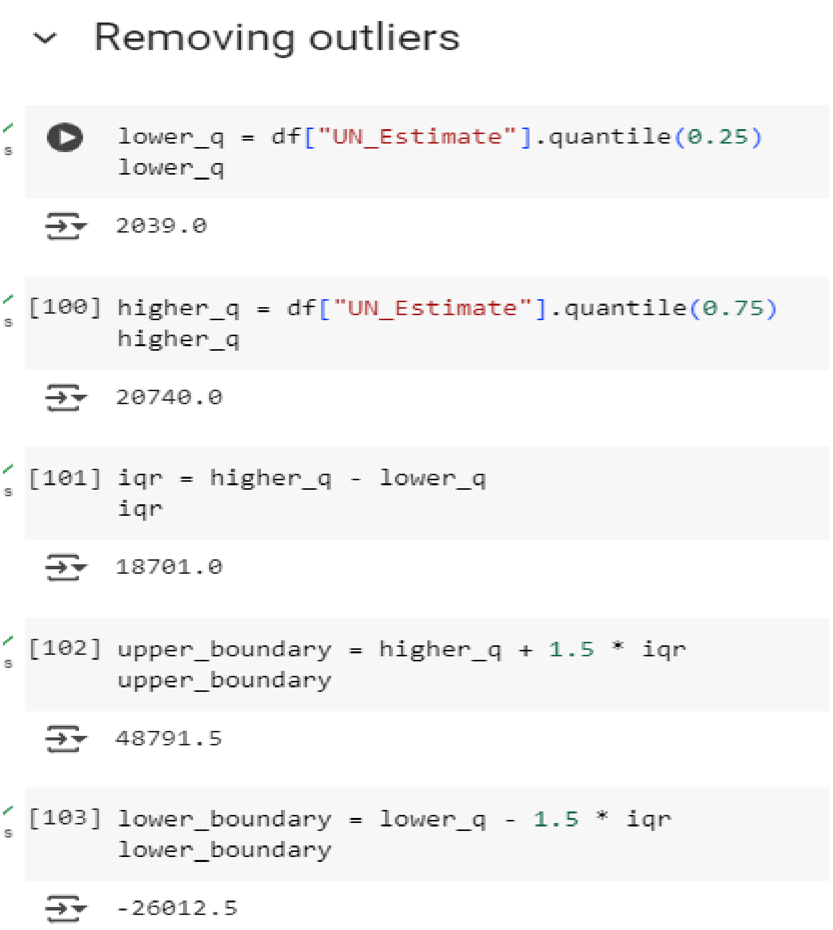


| | Country/Territory | UN_Region | IMF_Estimate | IMF_Year | WorldBank_Estimate | WorldBank_Year | UN_Estimate | UN_ |
|---|---|---|---|---|---|---|---|---|
| 1 | Monaco | Europe | 0 | 0 | 234316 | 2021 | 234317 | |
| 2 | Liechtenstein | Europe | 0 | 0 | 157755 | 2020 | 169260 | |
| 3 | Luxembourg | Europe | 132372 | 2023 | 133590 | 2021 | 133745 | |
| 4 | Ireland | Europe | 114581 | 2023 | 100172 | 2021 | 101109 | |
| 5 | Bermuda | Americas | 0 | 0 | 114090 | 2021 | 112653 | |

```
#Create another dataframe called data excluding 5 countries with highest UN estimate
data = df[-(df["UN_Estimate"]>100000)]
data.head()
```



| | Country/Territory | UN_Region | IMF_Estimate | IMF_Year | WorldBank_Estimate | WorldBank_Year | UN_Estimate | UN |
|---|---|---|---|---|---|---|---|---|
| 6 | Norway | Europe | 101103 | 2023 | 89154 | 2021 | 89242 | |
| 7 | Switzerland | Europe | 98767 | 2023 | 91992 | 2021 | 93525 | |
| 8 | Singapore | Asia | 91100 | 2023 | 72794 | 2021 | 66822 | |
| 9 | Isle of Man | Europe | 0 | 0 | 87158 | 2019 | 0 | |
| 10 | Cayman Islands | Americas | 0 | 0 | 86569 | 2021 | 85250 | |



## Removing outliers

```
lower_q = df["UN_Estimate"].quantile(0.25)
lower_q
```

```
2039.0
```

```
[100] higher_q = df["UN_Estimate"].quantile(0.75)
      higher_q
```

```
20740.0
```

```
[101] iqr = higher_q - lower_q
      iqr
```

```
18701.0
```

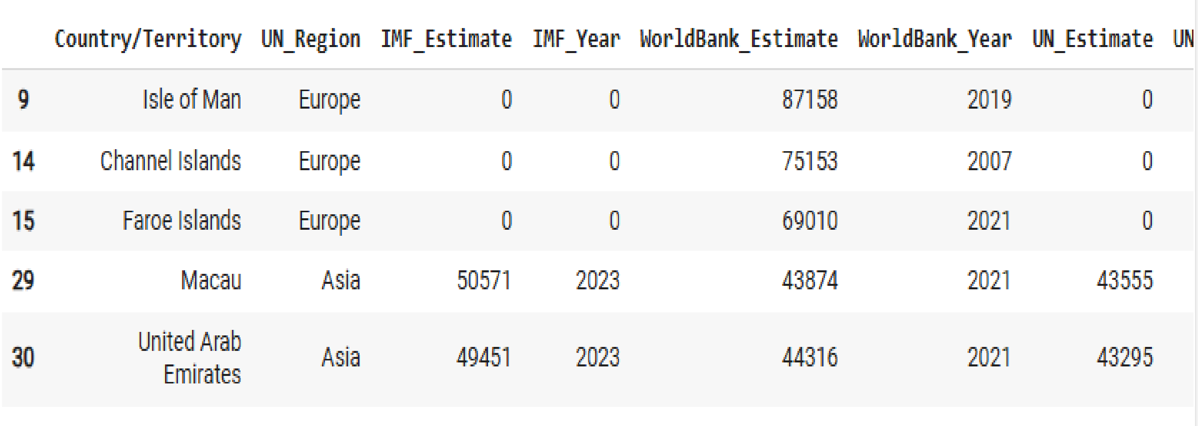
```
[102] upper_boundary = higher_q + 1.5 * iqr
      upper_boundary
```

```
48791.5
```

```
[103] lower_boundary = lower_q - 1.5 * iqr
      lower_boundary
```

```
-26012.5
```

df_filtered = df[(df["UN_Estimate"] < upper_boundary) & (df["UN_Estimate"] > lower_boundary)]
df_filtered.head()



| | Country/Territory | UN_Region | IMF_Estimate | IMF_Year | WorldBank_Estimate | WorldBank_Year | UN_Estimate | UN |
|---|---|---|---|---|---|---|---|---|
| **9** | Isle of Man | Europe | 0 | 0 | 87158 | 2019 | 0 | |
| **14** | Channel Islands | Europe | 0 | 0 | 75153 | 2007 | 0 | |
| **15** | Faroe Islands | Europe | 0 | 0 | 69010 | 2021 | 0 | |
| **29** | Macau | Asia | 50571 | 2023 | 43874 | 2021 | 43555 | |
| **30** | United Arab Emirates | Asia | 49451 | 2023 | 44316 | 2021 | 43295 | |