Pandas, Matplotlib and Seaborn

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import pandas as pd
```

Dataset Soruce

CONTENT

The figures presented here do not take into account differences in the cost of living in different countries, and the results vary greatly from one year to another based on fluctuations in the exchange rates of the country's currency. Such fluctuations change a country's ranking from one year to the next, even though they often make little or no difference to the standard of living of its population.

GDP per capita is often considered an indicator of a country's standard of living; however, this is inaccurate because GDP per capita is not a measure of personal income.

Comparisons of national income are also frequently made on the basis of purchasing power parity (PPP), to adjust for differences in the cost of living in different countries. (See List of countries by GDP (PPP) per capita.) PPP largely removes the exchange rate problem but not others; it does not reflect the value of economic output in international trade, and it also requires more estimation than GDP per capita. On the whole, PPP per capita figures are more narrowly spread than nominal GDP per capita figures.

Here are some resources to learn about GDP:

World Bank

Our World in Data

IMF

UN Data

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
df = pd.read_csv("/content/drive/MyDrive/JustIT_Python/pandas/Resources/Copy of GDP (nominal) per Capita.csv",encoding= 'unicode_esc
```

EDA (Exploratory Data Analysis)

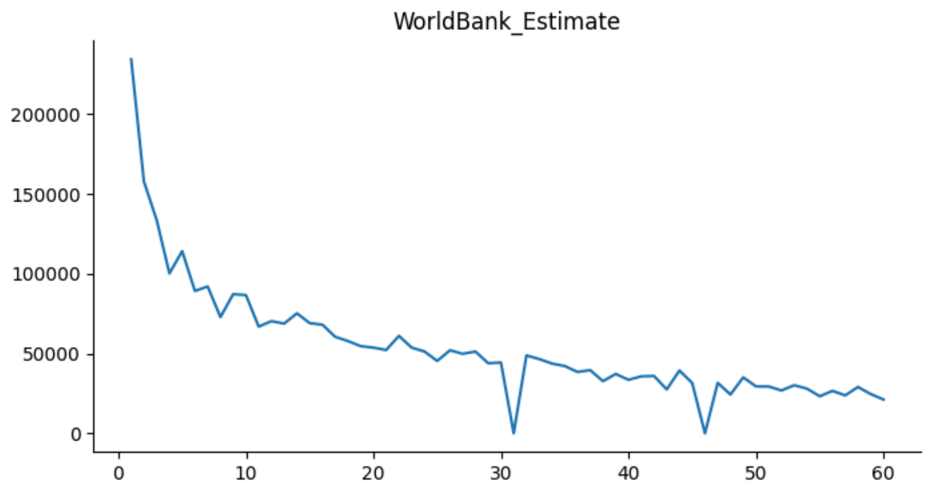
Use this section to explore and inspect dataset.

```
df[["Country/Territory", "UN_Region"]]
```

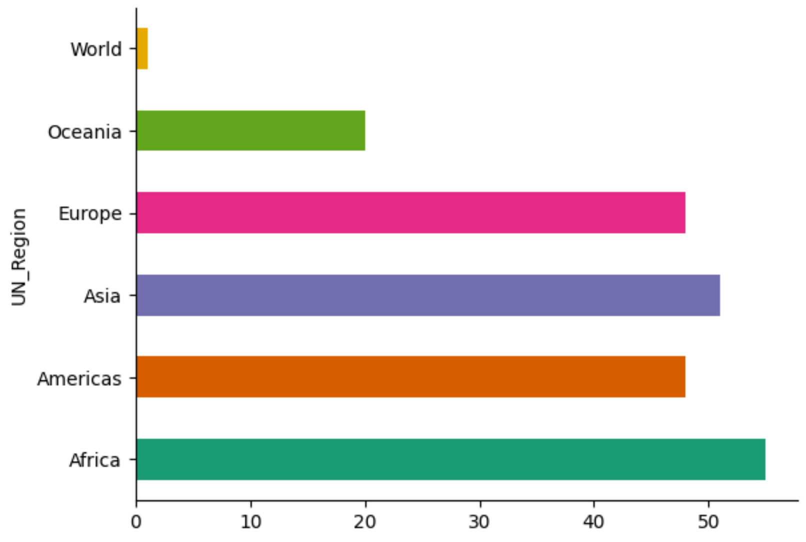
	Country/Territory	UN_Region
1	Monaco	Europe
2	Liechtenstein	Europe
3	Luxembourg	Europe
4	Ireland	Europe
5	Bermuda	Americas
...	...	...
219	Malawi	Africa
220	South Sudan	Africa
221	Sierra Leone	Africa
222	Afghanistan	Asia
223	Burundi	Africa

223 rows × 2 columns

```
from matplotlib import pyplot as plt
_df_12['WorldBank_Estimate'].plot(kind='line', figsize=(8, 4), title='WorldBank_Estimate')
plt.gca().spines[['top', 'right']].set_visible(False)
```



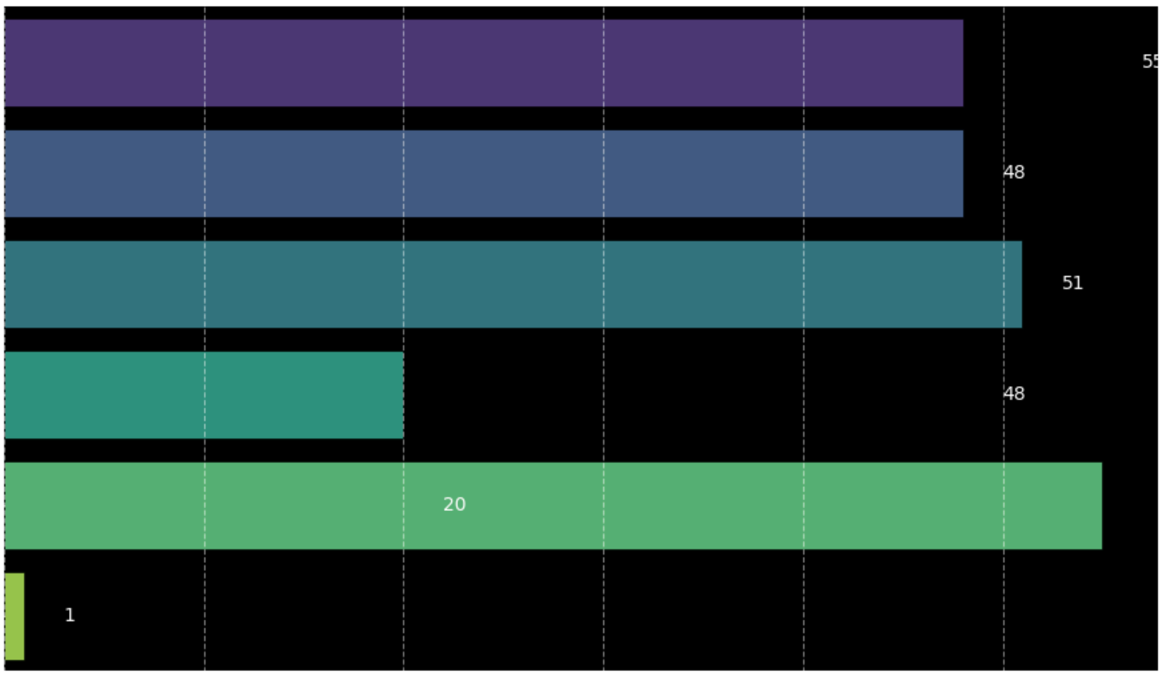
```
from matplotlib import pyplot as plt
import seaborn as sns
df.groupby('UN_Region').size().plot(kind='barh', color=sns.palettes.mpl_palette('Dark2'))
plt.gca().spines[['top', 'right',]].set_visible(False)
```



```
# prompt: make a barchart with lined background and looks beautiful

import matplotlib.pyplot as plt
import seaborn as sns

```

```
# Assuming 'df' is your DataFrame and 'UN_Region' column exists
# Create the bar chart with a lined background
plt.figure(figsize=(10, 6))  # Adjust figure size as needed

# Use a dark background for a more dramatic effect
plt.style.use('dark_background')

# Create the barplot
sns.countplot(y='UN_Region', data=df, palette='viridis', edgecolor='black', linewidth=0.5)

# Customize the plot
plt.title('Number of Countries/Territories per UN Region', fontsize=16)
plt.xlabel('Count', fontsize=12)
plt.ylabel('UN Region', fontsize=12)

# Add a subtle grid for better readability
plt.grid(axis='x', linestyle='--', alpha=0.5)

# Customize spines
plt.gca().spines[['top', 'right']].set_visible(False)
plt.gca().spines['left'].set_visible(False) # Hide left spine
plt.gca().spines['bottom'].set_linewidth(1.5)  # Set bottom spine width

# Add annotations (optional)
for i, v in enumerate(df.groupby('UN_Region').size()):
    plt.text(v + 2, i, str(v), color='white', va='center')


# Enhance visual appeal
plt.tight_layout()  # Adjust layout to prevent labels from overlapping

plt.show()

```

```
<ipython-input-125-e7d7e1b224b0>:14: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `le

  sns.countplot(y='UN_Region', data=df, palette='viridis', edgecolor='black', linewidth=0.5)
```



```
1 Start coding or generate with AI.
```

```
1 Start coding or generate with AI.
```

```
1 Start coding or generate with AI.
```

```
1 Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
# number of countries per region
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
#What is European Union[n 1]?
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
# Countries in Europe below avarege
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
## Which countries in Europe has higher GDP than UK?
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

groupby()

[Learn more about groupby]

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

Which countries below average by IMF world estimate?

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

IMF estimate 0 values

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

Which country has highest UN Estimate?

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

Which country has highest Worlbank Estimate?

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

Which country has highest IMF Estimate?

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

Filling 0 Values by average

```
import numpy as np
```

```
# replace 0 with null values
```

```
Start coding or generate with AI.
```

```
# Calculate the average of 'Worldbank_Estimate' and 'UN_Estimate' columns

```

```
# Fill the null values in 'imf' column with the calculated average

```

```
Start coding or generate with AI.
```

```
# Drop the temporary 'avg_worldbank_un' column if not needed

```

```
Start coding or generate with AI.
```

Visit this link to learn more about ffill

Visit this link to learn more about bfill

Checking Missing Values

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

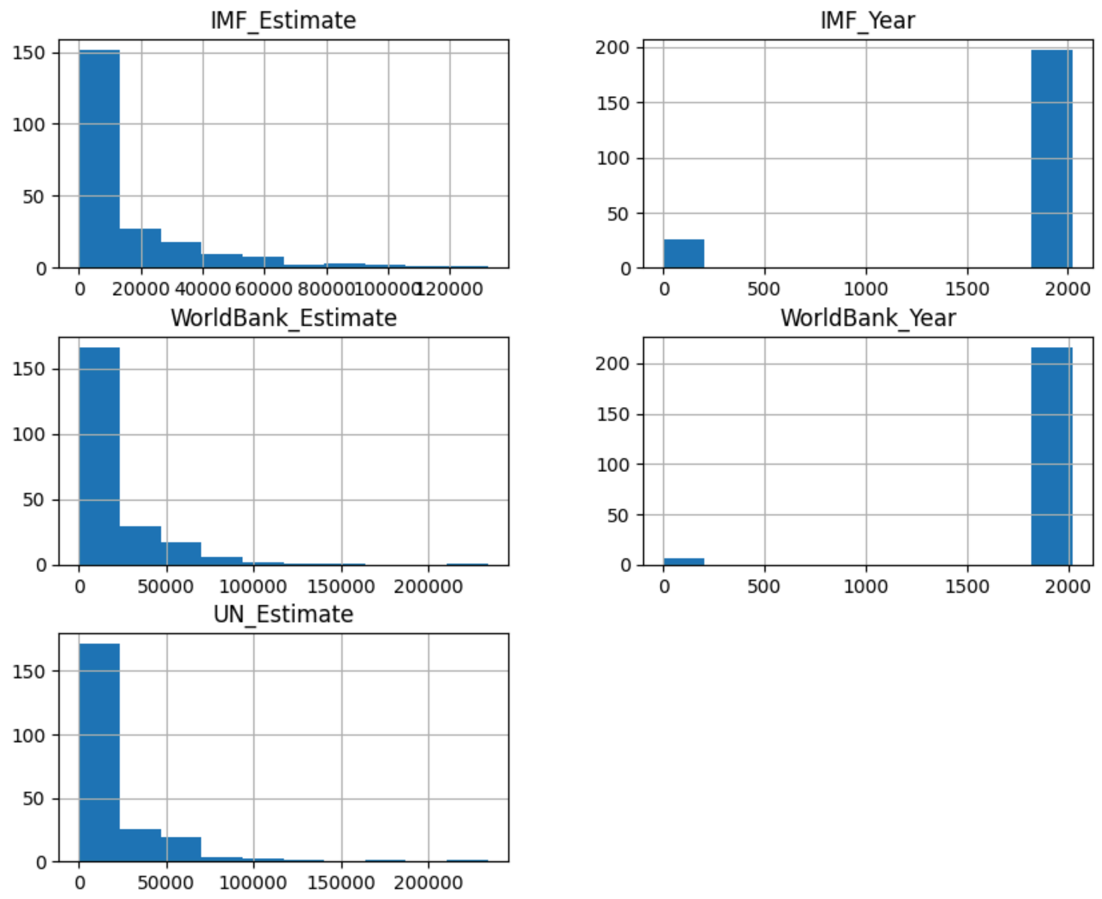
```
Start coding or generate with AI.
```

Visualization

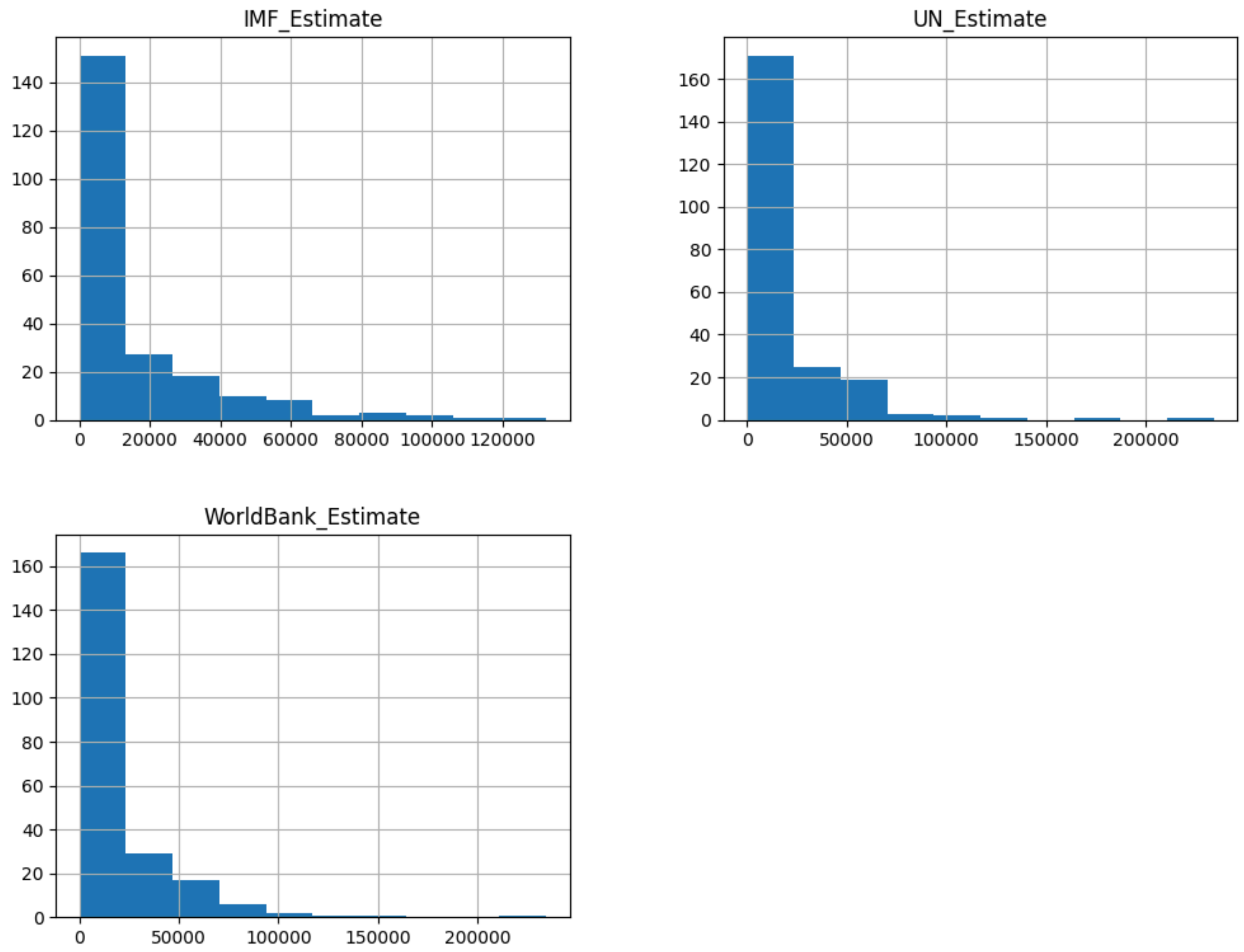
```
import matplotlib.pyplot as plt
import seaborn as sns
```

Histogram

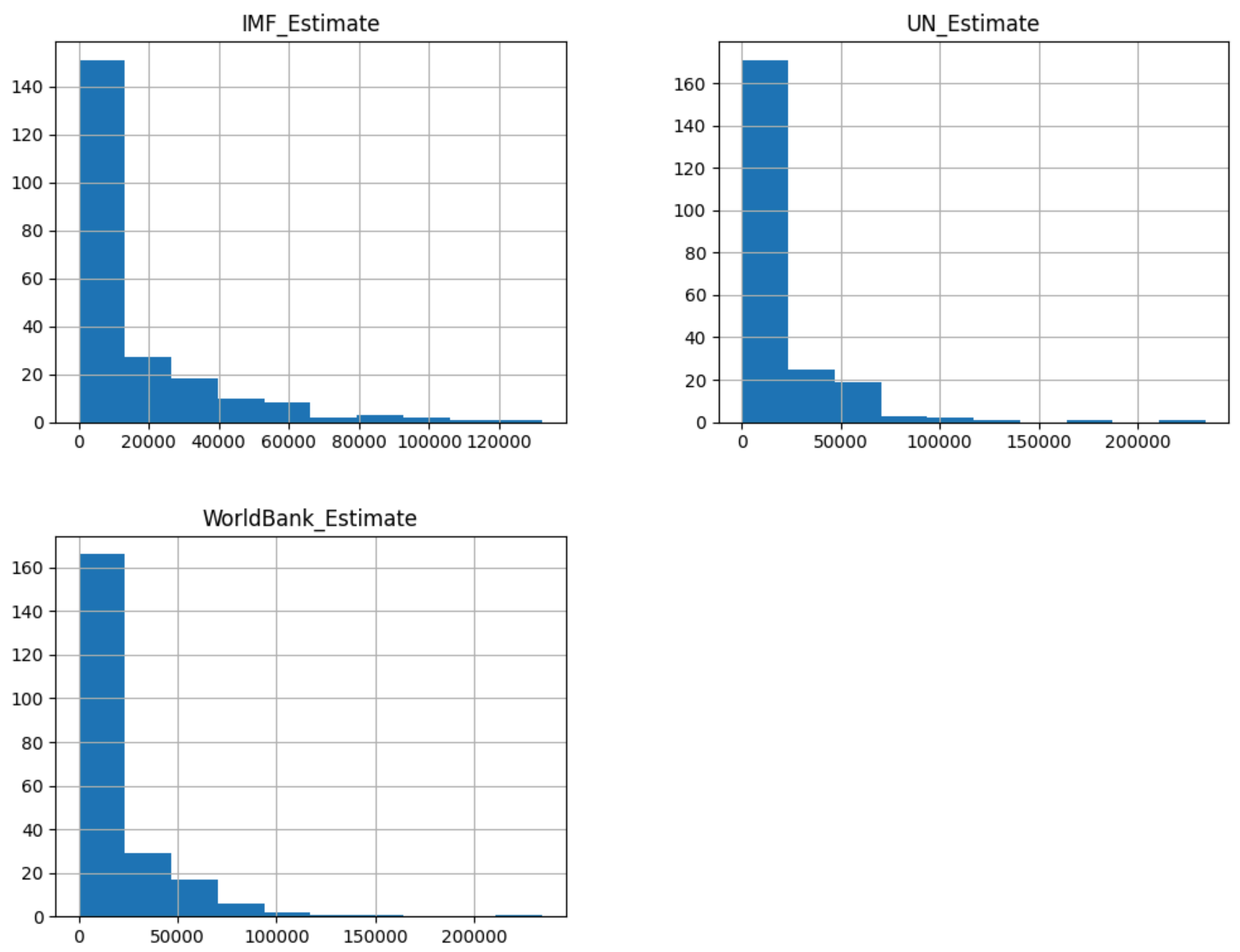
```
df.hist(figsize=(10,8))
plt.show()
```



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

plt.show()
```



```
df[["IMF_Estimate", "UN_Estimate", "WorldBank_Estimate"]].hist(bins=10, figsize=(12,9))

plt.show()
```



```
df["WorldBank_Estimate"].agg(["min","max"])
```

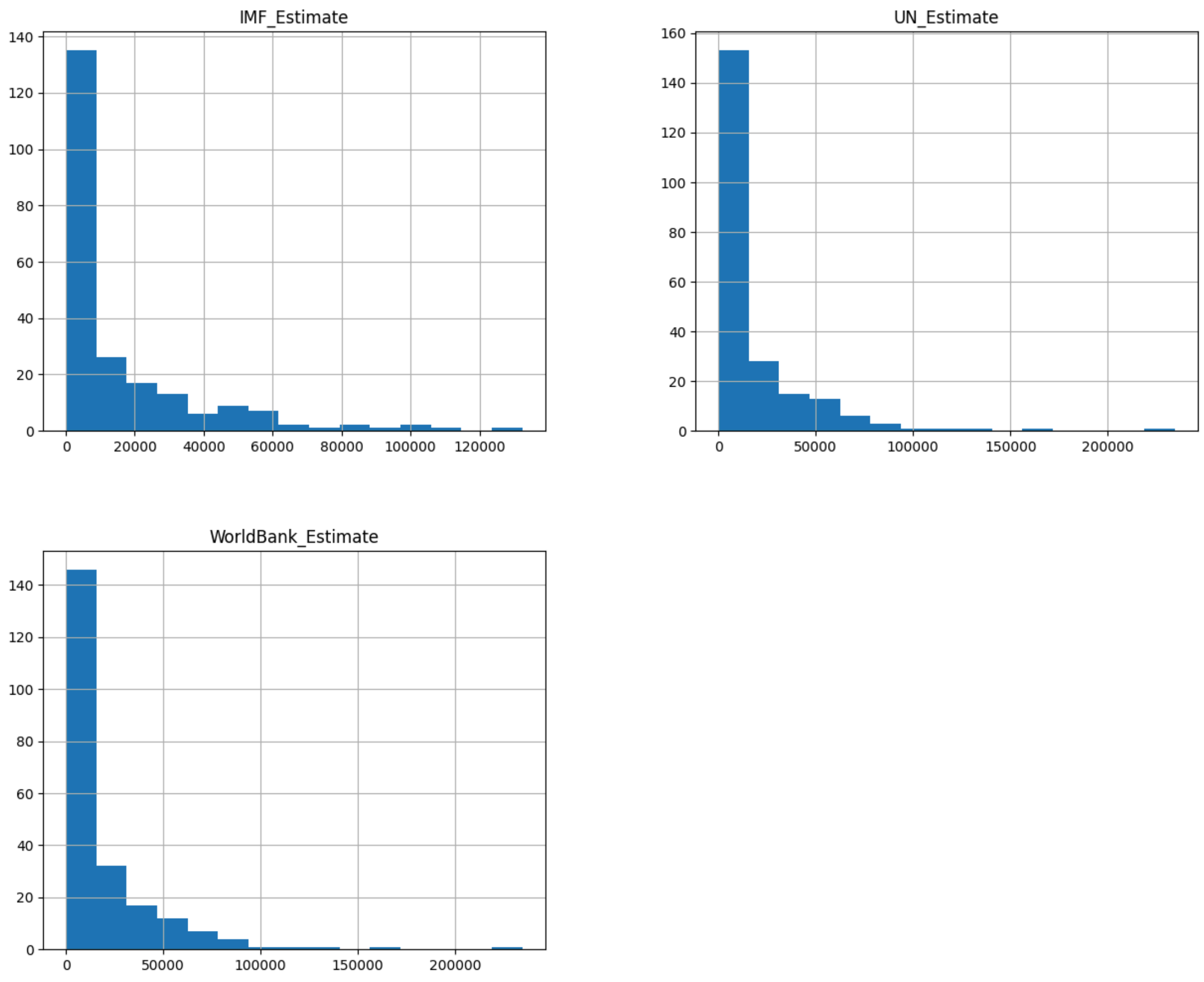
```
234316/5
#1 bin size if bins=5
```

```
df[df["WorldBank_Estimate"]<=46863.2]["WorldBank_Estimate"].count()
```

```
234316/10
#1 bin size if bins not given any number
```

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

plt.show()
```

```
df[["IMF_Estimate", "UN_Estimate", "WorldBank_Estimate"]].hist(bins=15, figsize=(15,12))

#23400/15 = 15300
plt.show()
```



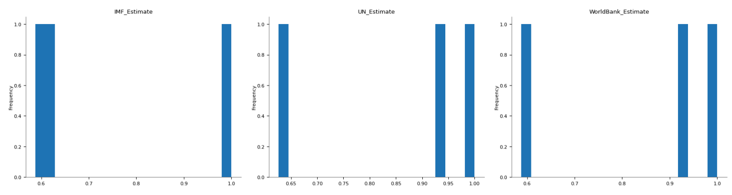
Correlation Heatmap

```
#creating correlation for the whole dataset - if you dont know the columns you are finding correlation for than you can run a wider
numerical_df = df.select_dtypes(include=[int, float])
corr = numerical_df.corr()
```

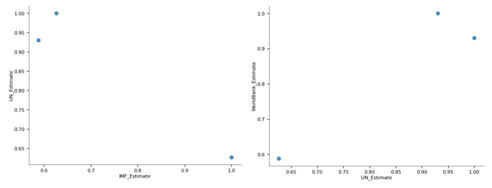
```
df[["IMF_Estimate", "UN_Estimate", "WorldBank_Estimate"]].corr()
```

	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

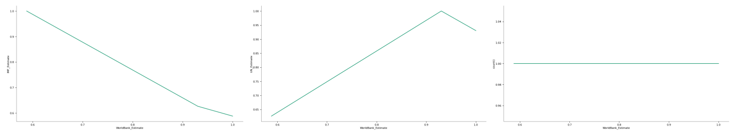
Distributions



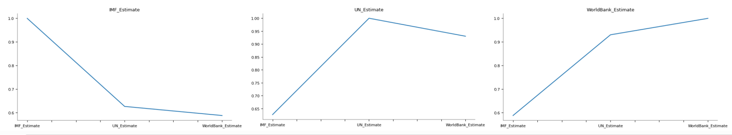
2-d distributions



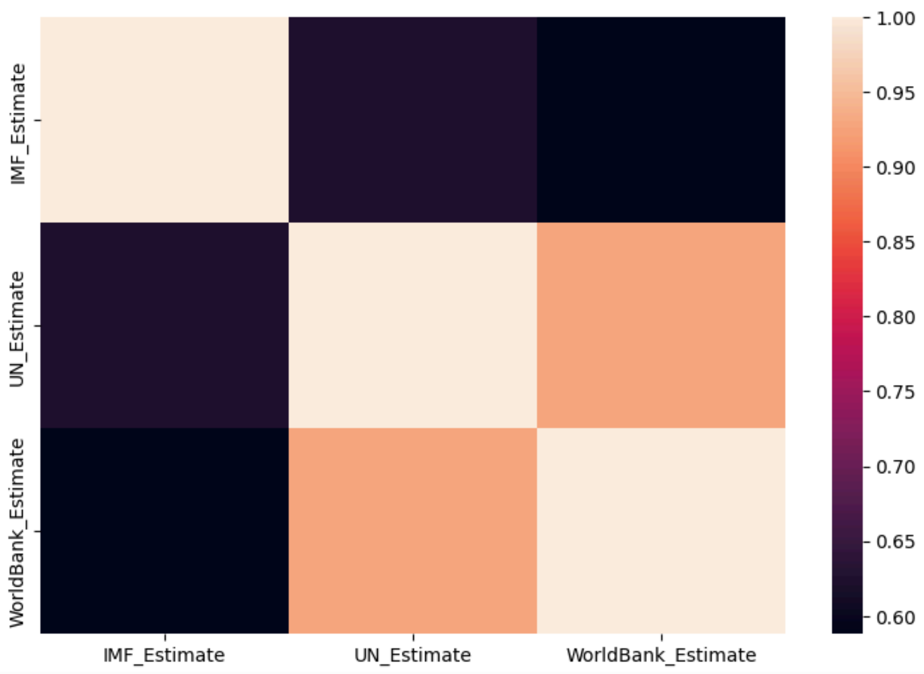
Time series



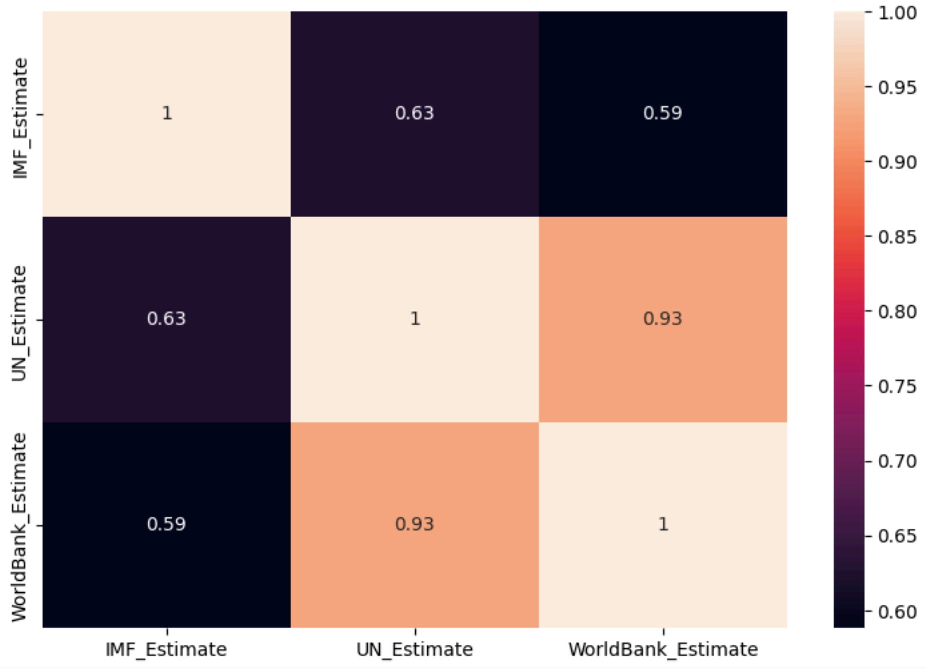
Values



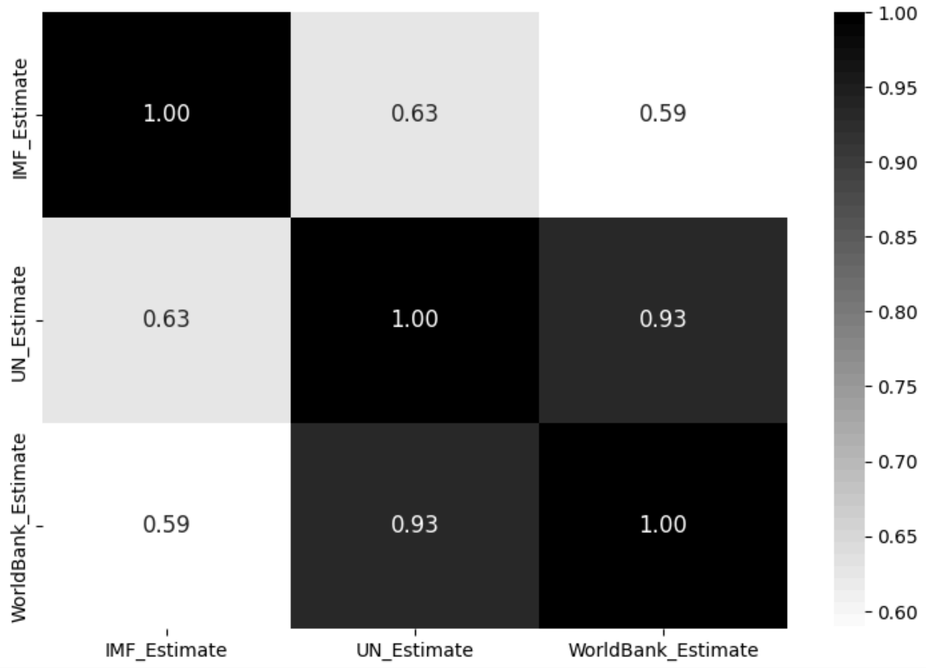
```
corr = df[["IMF_Estimate", "UN_Estimate", "WorldBank_Estimate"]].corr()

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()
```



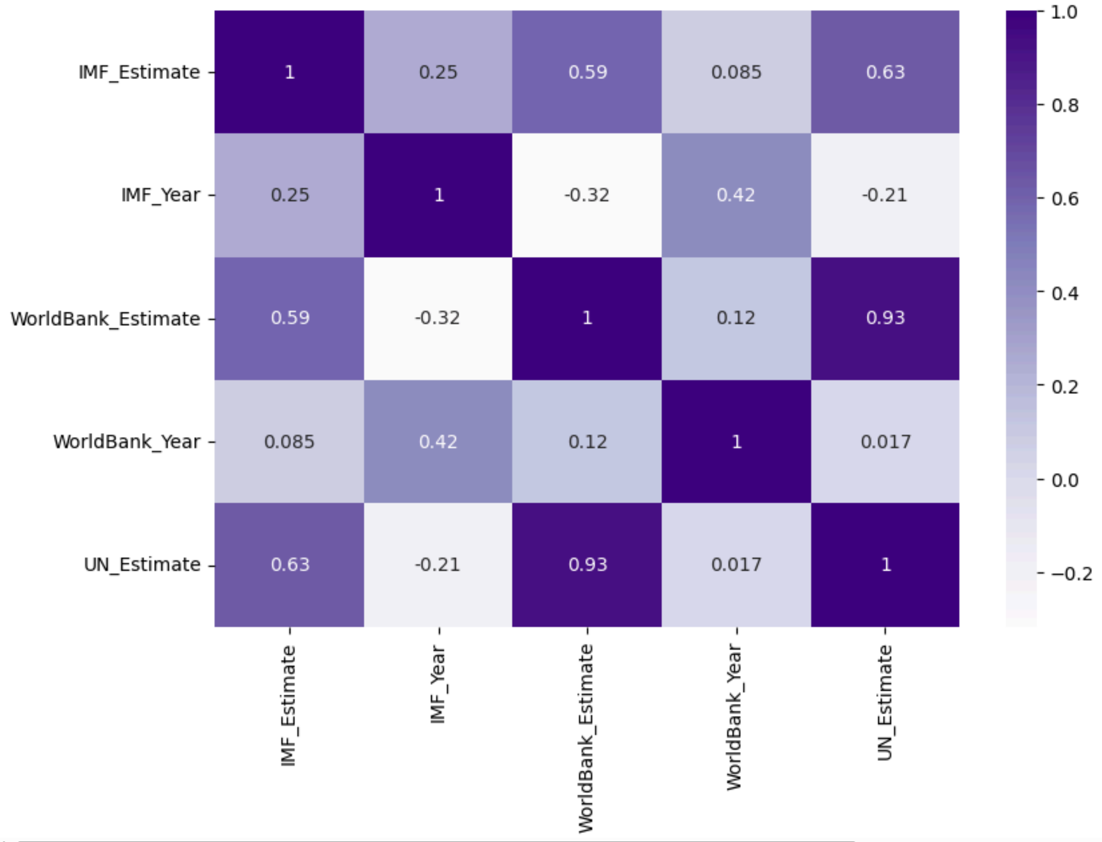
```
corr = df[["IMF_Estimate", "UN_Estimate", "WorldBank_Estimate"]].corr()

plt.figure(figsize=(9,6))

sns.heatmap(corr, annot=True, fmt=".2f", cmap = 'gist_yarg', 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()
```

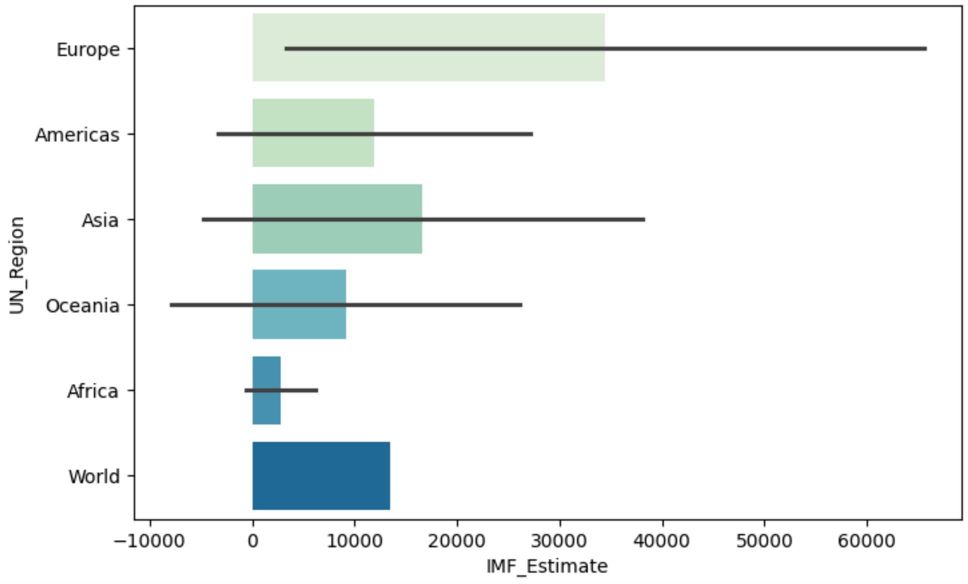


	IMF_Estimate	IMF_Year	WorldBank_Estimate	WorldBank_Year	UN_Estimate
IMF_Estimate	1	0.25	0.59	0.085	0.63
IMF_Year	0.25	1	-0.32	0.42	-0.21
WorldBank_Estimate	0.59	-0.32	1	0.12	0.93
WorldBank_Year	0.085	0.42	0.12	1	0.017
UN_Estimate	0.63	-0.21	0.93	0.017	1

Bar plot

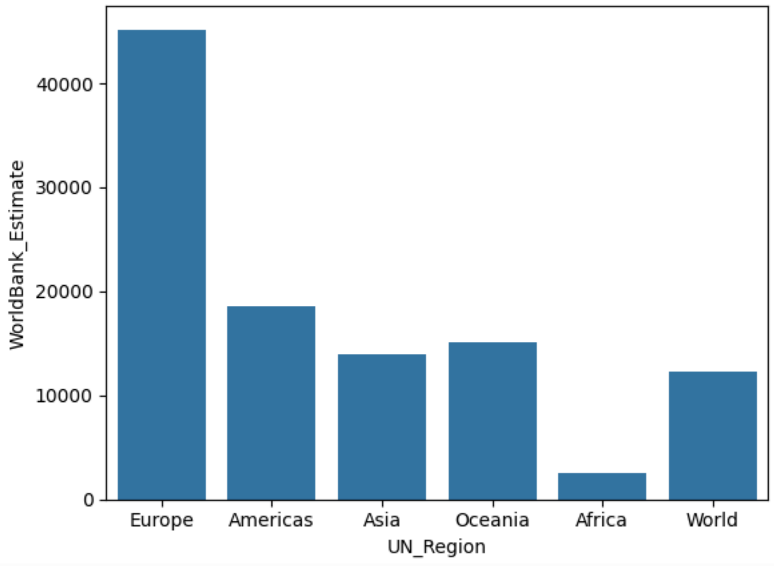
```
import seaborn as sns
import matplotlib.pyplot as plt

# Set figure size
fig = plt.figure(figsize=(8,5))

# Create a bar plot comparing IMF GDP estimates by region
sns.barplot(x="IMF_Estimate", y="UN_Region", data=df, ci="sd", palette="GnBu")

# Show the plot
plt.show()
```

```
<ipython-input-74-c01b2327fa78>:8: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(x="IMF_Estimate", y="UN_Region", data=df, ci="sd", palette="GnBu")
<ipython-input-74-c01b2327fa78>:8: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `le

  sns.barplot(x="IMF_Estimate", y="UN_Region", data=df, ci="sd", palette="GnBu")
```

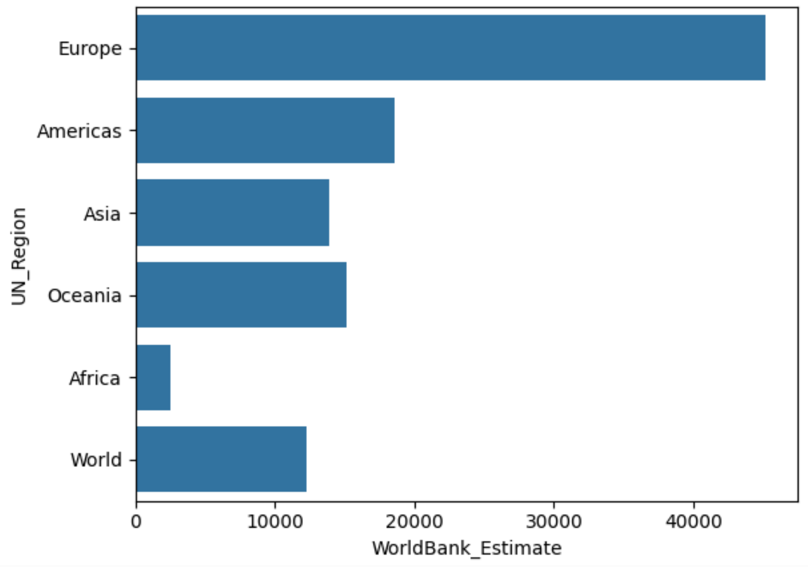


```
df.head()
```

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

plt.show()
```



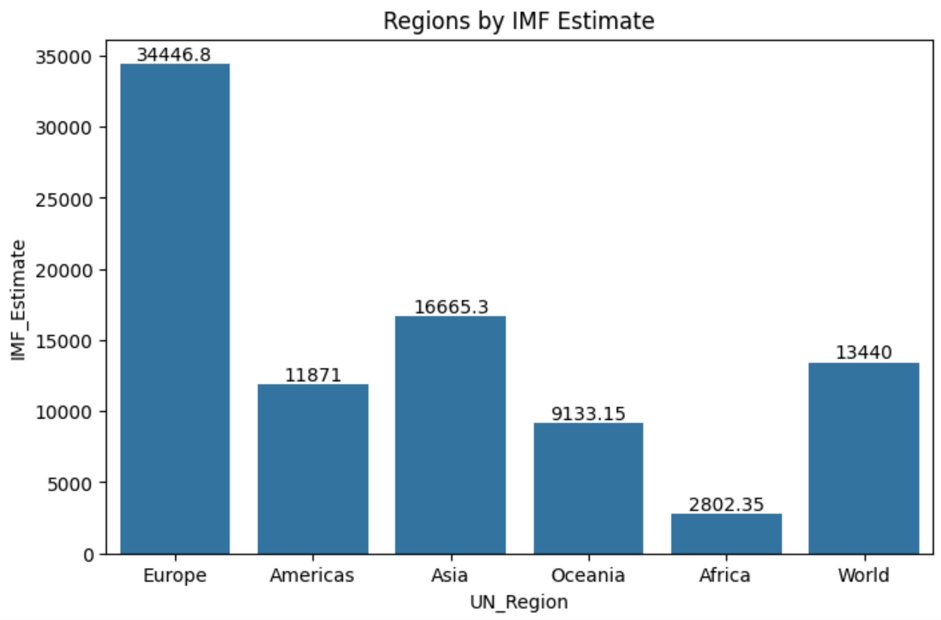
```
sns.barplot(x="WorldBank_Estimate", y="UN_Region", data=df, errorbar=None)

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()
```

Show hidden output

```
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()
```

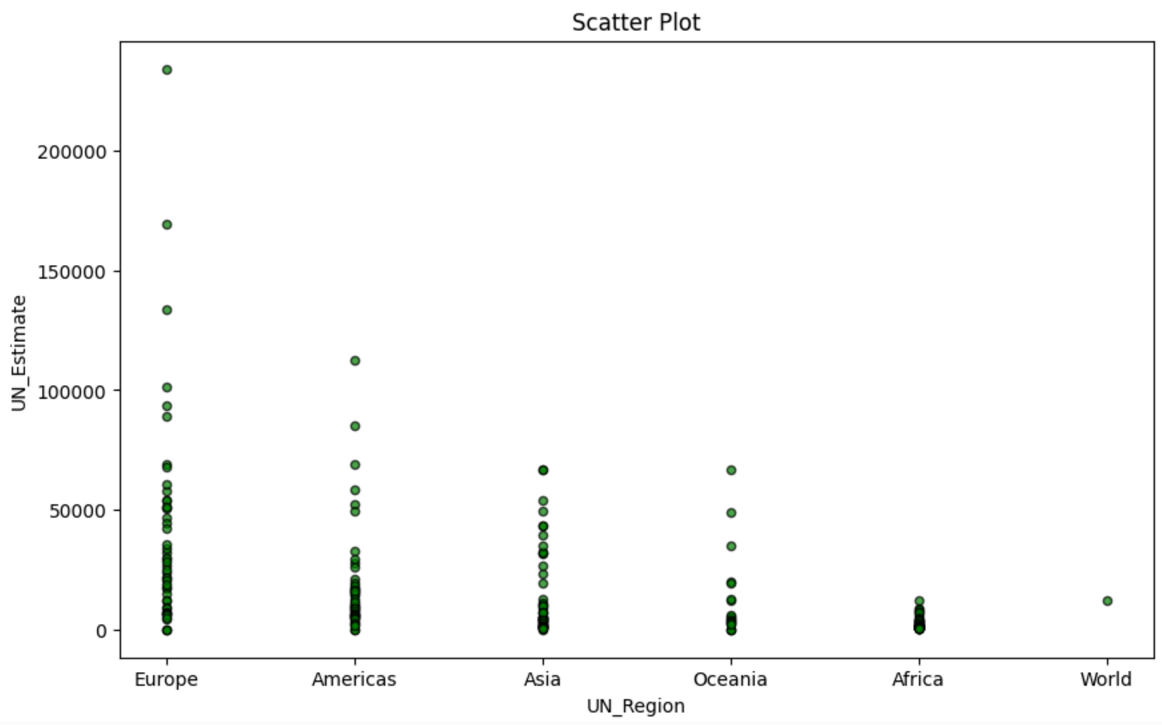


```
Start coding or generate with AI.
```

Scatter Plot

```
df.plot(x='UN_Region', y='UN_Estimate', kind='scatter', alpha=0.7, color="green", edgecolor="black",
        figsize=(10,6),
        title="Scatter Plot")
```

```
plt.show()
```



```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

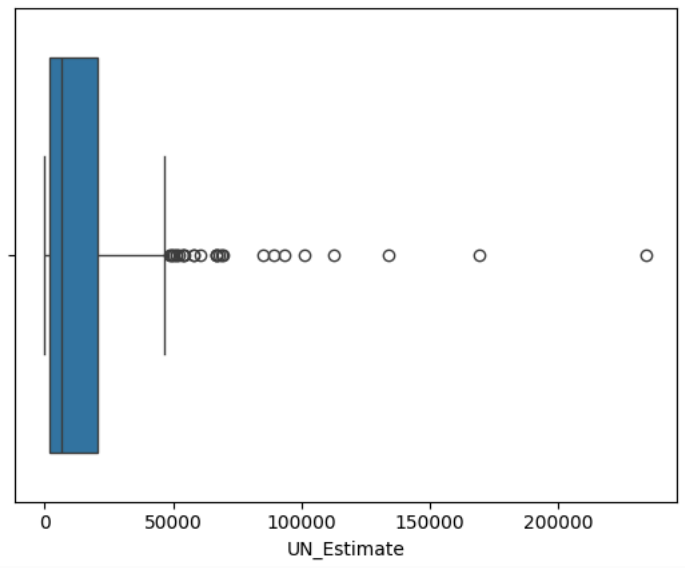
```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

Boxplot and Outliers

image.png

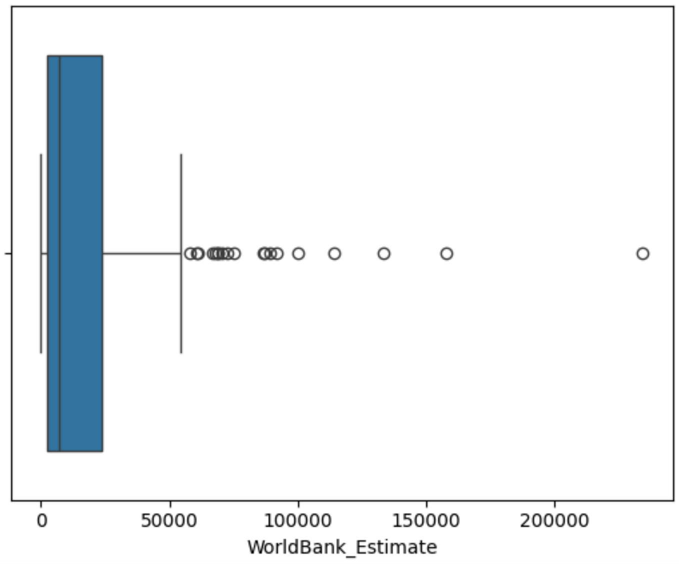
```
sns.boxplot(x=df["UN_Estimate"])

plt.show()
```



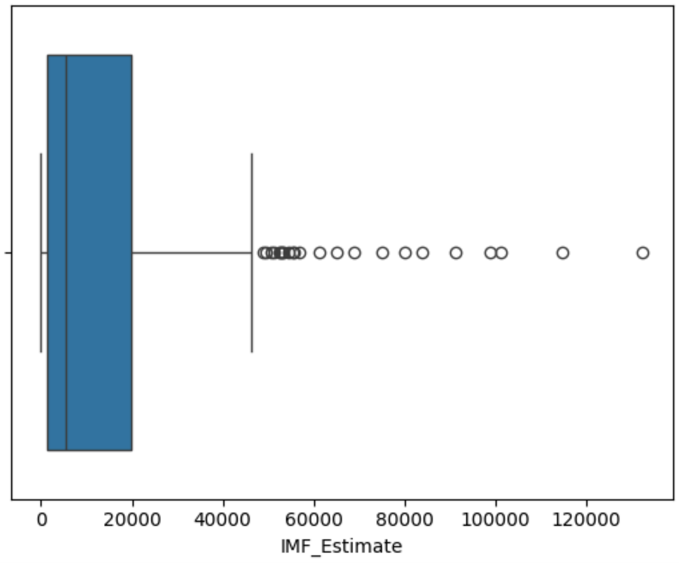
```
df[df["UN_Estimate"]>50000].head()
```

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

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

plt.show()
```



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

plt.show()
```



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

```
df.UN_Estimate.mean()
```

```
df.shape
```

```
Start coding or generate with AI.
```

```
import seaborn as sns
import matplotlib.pyplot as plt

```

```
# Create count plot excluding 'World' category
ax = sns.countplot(x="UN_Region", data=df[~(df.UN_Region=="World")], palette="Accent"

# Add count labels on top of bars
ax.bar_label(ax.containers[0])
for labels in ax.containers:
    ax.bar_label(labels)

# Add title
plt.title("Number of Countries in each continent")

#Add axis titles
plt.xlabel("UN Region")
plt.ylabel("Number of Countries")

# Set background color to light gray
ax.set_facecolor('papayawhip')
```