

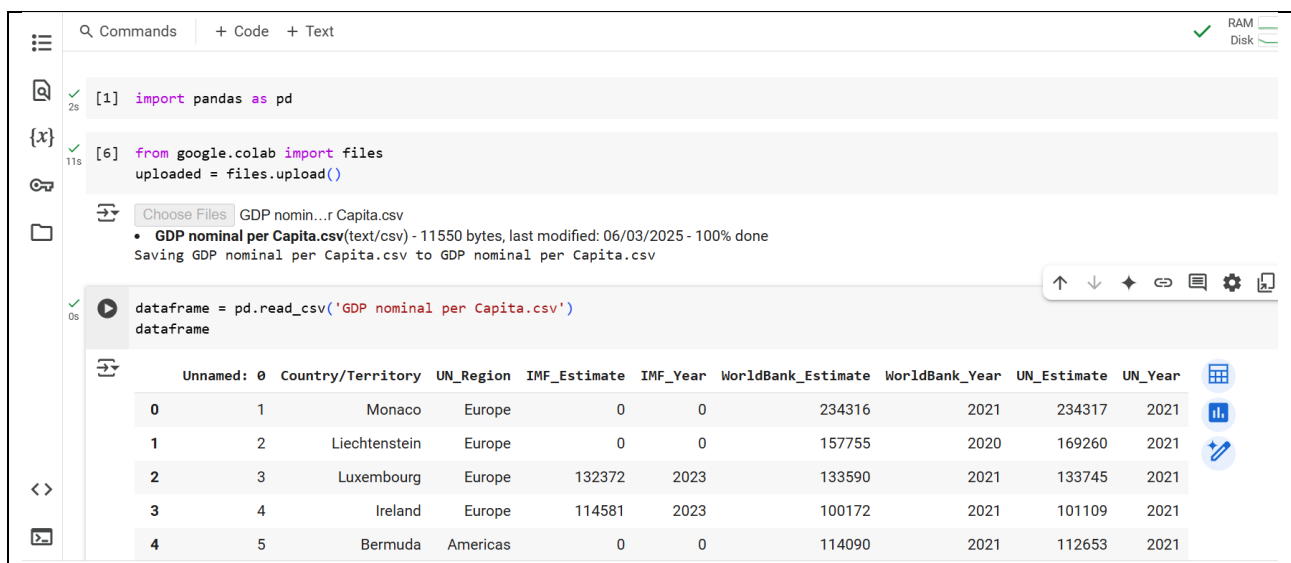
 B2Wgroup

# Data Technician

## Day 4: Task 1

Using the 'GDP (nominal) per Capita.csv' which can be downloaded [here](#), complete the below exercises and paste your input and output. Work individually, but we will work and support each other in the room.

- Read and save the 'GDP (nominal) per Capita' data to a data frame called "df" in Jupyter notebook
- Print the first 10 rows
- Print the last 5 rows
- Print 'Country/Territory' and 'UN\_Region' columns



The screenshot shows a Jupyter Notebook interface with the following code cells:

```
[1] import pandas as pd
```

```
[6] from google.colab import files
uploaded = files.upload()
```

Below the code, a file upload dialog shows 'GDP nominal per Capita.csv' (11550 bytes) being uploaded. The code then reads the file into a DataFrame:

```
dataframe = pd.read_csv('GDP nominal per Capita.csv')
dataframe
```

The output displays the first 10 rows of the DataFrame:

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

First 10 rows

Q Commands + Code + Text

dataframe.head(10)

	Unnamed: 0	Country/Territory	UN_Region	IMF_Estimate	IMF_Year	WorldBank_Estimate	WorldBank_Year	UN_Estimate	UN_Year
0	1	Monaco	Europe	0	0	234316	2021	234317	2021
1	2	Liechtenstein	Europe	0	0	157755	2020	169260	2021
2	3	Luxembourg	Europe	132372	2023	133590	2021	133745	2021
3	4	Ireland	Europe	114581	2023	100172	2021	101109	2021
4	5	Bermuda	Americas	0	0	114090	2021	112653	2021
5	6	Norway	Europe	101103	2023	89154	2021	89242	2021
6	7	Switzerland	Europe	98767	2023	91992	2021	93525	2021
7	8	Singapore	Asia	91100	2023	72794	2021	66822	2021
8	9	Isle of Man	Europe	0	0	87158	2019	0	0
9	10	Cayman Islands	Americas	0	0	86569	2021	85250	2021

The last 5 rows

dataframe.tail(5)

	Unnamed: 0	Country/Territory	UN_Region	IMF_Estimate	IMF_Year	WorldBank_Estimate	WorldBank_Year	UN_Estimate	UN_Year
218	219	Malawi	Africa	496	2023	635	2021	613	2021
219	220	South Sudan	Africa	467	2023	1072	2015	400	2021
220	221	Sierra Leone	Africa	415	2023	480	2021	505	2021
221	222	Afghanistan	Asia	611	2020	369	2021	373	2021
222	223	Burundi	Africa	249	2023	222	2021	311	2021

'Country/Territory' and 'UN\_Region' columns





Commands

+ Code

+ Text



0s



```
dataframe[['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 x 2 columns



## Day 4: Task 2

Back with 'GDP (nominal) per Capita'. As a group, import and work your way through the Day\_4\_Python\_Activity.ipynb notebook which can be found [here](#). There are questions to answer, but also opportunities to have fun with the data – paste your input and output below.

Once complete, and again as a group, work with some more data and have some fun – there is no set agenda for this section, other than to embed the skills developed this week. Paste your input and output below and upon return we'll discuss progress made.

[Additional data found here.](#)



### Number of countries per region

{x} ✓ [76] 0s dataframe.groupby('UN\_Region')['Country/Territory'].count()

Country/Territory

UN_Region	
Africa	55
Americas	48
Asia	51
Europe	48
Oceania	20
World	1

<> dtype: int64

What is European Union [n 1]



name: Country/Territory, Length: 225, dtype: object

```
[71] European_Union = df[df["Country/Territory"].isin(df.loc[df["UN_Region"] == "Europe", "Country/Territory"])]  
European_Union
```

	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
6	Norway	Europe	101103	2023	89154	2021	89242	2021
7	Switzerland	Europe	98767	2023	91992	2021	93525	2021
9	Isle of Man	Europe	0	0	87158	2019	0	0
13	Iceland	Europe	75180	2023	68728	2021	69133	2021
14	Channel Islands	Europe	0	0	75153	2007	0	0
15	Faroe Islands	Europe	0	0	69010	2021	0	0
16	Denmark	Europe	68827	2023	68008	2021	68037	2021
18	Netherlands	Europe	61098	2023	57768	2021	57871	2021
20	Austria	Europe	56802	2023	53638	2021	53840	2021
22	Sweden	Europe	55395	2023	61029	2021	60730	2021
23	Finland	Europe	54351	2023	53655	2021	53703	2021
24	Belgium	Europe	53377	2023	51247	2021	51166	2021
25	San Marino	Europe	52949	2023	45320	2020	50425	2021
28	Germany	Europe	51383	2023	51204	2021	51073	2021
33	United Kingdom	Europe	46371	2023	46510	2021	46542	2021
34	France	Europe	44408	2023	43659	2021	44229	2021
35	Andorra	Europe	44387	2023	42137	2021	42066	2021
36	European Union[n 1]	Europe	39940	2023	38411	2021	31875	2021
40	Malta	Europe	36989	2023	33487	2021	33642	2021
41	Italy	Europe	36812	2023	35658	2021	35579	2021
51	Slovenia	Europe	32214	2023	29291	2021	29135	2021
52	Czech Republic	Europe	31368	2023	26821	2021	26809	2021



35	Andorra	Europe	44387	2023	42137	2021	42066	2021
36	European Union[n 1]	Europe	39940	2023	38411	2021	31875	2021
40	Malta	Europe	36989	2023	33487	2021	33642	2021
41	Italy	Europe	36812	2023	35658	2021	35579	2021
51	Slovenia	Europe	32214	2023	29291	2021	29135	2021
52	Czech Republic	Europe	31368	2023	26821	2021	26809	2021
53	Spain	Europe	31223	2023	30104	2021	30058	2021
54	Estonia	Europe	31209	2023	27944	2021	27991	2021
57	Lithuania	Europe	28094	2023	23723	2021	23844	2021
59	Portugal	Europe	26012	2023	24568	2021	24651	2021
60	Latvia	Europe	25136	2023	21148	2021	21267	2021
62	Slovakia	Europe	23457	2023	21392	2021	21390	2021
63	Greece	Europe	22595	2023	20193	2021	20571	2021
70	Croatia	Europe	20537	2023	17685	2021	16983	2021
72	Poland	Europe	19912	2023	18000	2021	17736	2021
75	Hungary	Europe	19385	2023	18728	2021	18728	2021
78	Romania	Europe	18530	2023	14858	2021	14698	2021
87	Bulgaria	Europe	14893	2023	12222	2021	12207	2021
90	Russia	Europe	14403	2023	12195	2021	12259	2021
103	Montenegro	Europe	11289	2023	9466	2021	9252	2021
106	Serbia	Europe	10849	2023	9230	2021	8643	2021
112	Bosnia and Herzegovina	Europe	8223	2023	7143	2021	7143	2021
115	Belarus	Europe	7944	2023	7302	2021	7121	2021
118	North Macedonia	Europe	7384	2023	6695	2021	6600	2021
120	Albania	Europe	7058	2023	6493	2021	6396	2021
127	Moldova	Europe	6342	2023	5231	2021	4468	2021
133	Kosovo	Europe	5641	2023	5270	2021	5663	2021
143	Ukraine	Europe	4654	2023	4836	2021	4596	2021

< >

Countries in Europe below average



```
[ ] # Countries in Europe below average
```

```
[27] average_GDP = df['IMF_Estimate'].mean()
      average_GDP
```

```
15351.632286995517
```

```
[29] average_gdp = df['IMF_Estimate'].mean()
      europe_data = df[df['UN_Region'] == 'Europe']
      below_average_europe = europe_data[europe_data['IMF_Estimate'] < average_gdp]
      print(below_average_europe[['Country/Territory', 'IMF_Estimate']])
```

	Country/Territory	IMF_Estimate
1	Monaco	0
2	Liechtenstein	0
9	Isle of Man	0
14	Channel Islands	0
15	Faroe Islands	0
87	Bulgaria	14893
90	Russia	14403
103	Montenegro	11289
106	Serbia	10849
112	Bosnia and Herzegovina	8223
115	Belarus	7944
118	North Macedonia	7384
120	Albania	7058
127	Moldova	6342
133	Kosovo	5641
143	Ukraine	4654

Countries in Europe has high GDP greater than UK

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

```
[46] UK_GDP = df[df['Country/Territory'] == 'United Kingdom']['IMF_Estimate'].values[0]
      UK_GDP
```

```
46371
```

```
[47] UK_GDP = df[df['Country/Territory'] == 'United Kingdom']['IMF_Estimate'].values[0]
      europe_data = df[df['UN_Region'] == 'Europe']
      higher_gdp_than_uk = europe_data[europe_data['IMF_Estimate'] > UK_GDP]
      print(higher_gdp_than_uk[['Country/Territory', 'IMF_Estimate']])
```

	Country/Territory	IMF_Estimate
3	Luxembourg	132372
4	Ireland	114581
6	Norway	101103
7	Switzerland	98767
13	Iceland	75180
16	Denmark	68827
18	Netherlands	61098
20	Austria	56802
22	Sweden	55395
23	Finland	54351
24	Belgium	53377
25	San Marino	52949
28	Germany	51383

## Group by UN\_Region

```
df.groupby("UN_Region").sum().drop(columns=["UN_Year"])
```

	Country/Territory	IMF_Estimate	IMF_Year	WorldBank_Estimate	WorldBank_Year	UN_Estimate
UN_Region						
Africa	SeychellesMauritiusEquatorial GuineaGabonBotsw...	155340.0	109242.0	135896.0	109118.0	132986.0
Americas	BermudaCayman IslandsUnited StatesGreenlandCan...	1046631.5	72828.0	891126.0	90925.0	897780.0
Asia	SingaporeQatarIsraelHong KongMacauUnited Arab ...	851311.0	99119.0	709987.0	99022.0	717520.0
Europe	MonacoLiechtensteinLuxembourgIrelandNorwaySwit...	2282589.0	86989.0	2169297.0	96990.0	1949318.0
Oceania	AustraliaNew ZealandNew CaledoniaGuamNorthern ...	333274.0	28322.0	302273.0	38397.0	252275.0
World	World	13440.0	2023.0	12235.0	2021.0	12230.0

```
[132] df.groupby(['UN_Region'], sort = False).sum().drop(columns=['UN_Year'])
```

	Country/Territory	IMF_Estimate	IMF_Year	WorldBank_Estimate	WorldBank_Year	UN_Estimate
UN_Region						
Europe	MonacoLiechtensteinLuxembourgIrelandNorwaySwit...	2282589.0	86989.0	2169297.0	96990.0	1949318.0
Americas	BermudaCayman IslandsUnited StatesGreenlandCan...	1046631.5	72828.0	891126.0	90925.0	897780.0
Asia	SingaporeQatarIsraelHong KongMacauUnited Arab ...	851311.0	99119.0	709987.0	99022.0	717520.0
Oceania	AustraliaNew ZealandNew CaledoniaGuamNorthern ...	333274.0	28322.0	302273.0	38397.0	252275.0
Africa	SeychellesMauritiusEquatorial GuineaGabonBotsw...	155340.0	109242.0	135896.0	109118.0	132986.0
World	World	13440.0	2023.0	12235.0	2021.0	12230.0

## Countries below average by IMF world estimate

Which countries below average by IMF world estimate?

```
[78] IMF_Average = df["IMF_Estimate"].mean()
IMF_Average
```

```
15351.632286995517
```

```
[79] Countries_below_average = df[df["IMF_Estimate"] < IMF_Average]
Countries_below_average
```

	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
5	Bermuda	Americas	0	0	114090	2021	112653	2021
9	Isle of Man	Europe	0	0	87158	2019	0	0
10	Cayman Islands	Americas	0	0	86569	2021	85250	2021
...	...	...	...	...	...	...	...	...
219	Malawi	Africa	496	2023	635	2021	613	2021
220	South Sudan	Africa	467	2023	1072	2015	400	2021
221	Sierra Leone	Africa	415	2023	480	2021	505	2021
222	Afghanistan	Asia	611	2020	369	2021	373	2021
223	Burundi	Africa	249	2023	222	2021	311	2021

159 rows x 8 columns



## IMF Estimate = 0

### IMF estimate 0 values

[81] df[df["IMF\_Estimate"] == 0]



	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
5	Bermuda	Americas	0	0	114090	2021	112653	2021
8	Isle of Man	Europe	0	0	87158	2019	0	0
10	Cayman Islands	Americas	0	0	86569	2021	85250	2021
14	Channel Islands	Europe	0	0	75153	2007	0	0
15	Faroe Islands	Europe	0	0	69010	2021	0	0
19	Greenland	Americas	0	0	54571	2020	58185	2021
31	British Virgin Islands	Americas	0	0	0	0	49444	2021
37	US Virgin Islands	Americas	0	0	39552	2020	0	0
39	New Caledonia	Oceania	0	0	37160	2021	34994	2021
42	Guam	Oceania	0	0	35905	2021	0	0
53	Sint Maarten (Dutch part)	Americas	0	0	28988	2018	26199	2021
61	Northern Mariana Islands	Oceania	0	0	23707	2019	0	0
65	Saint Martin (French part)	Americas	0	0	21921	2014	0	0
68	Turks and Caicos Islands	Americas	0	0	20909	2021	20909	2021
71	French Polynesia	Oceania	0	0	19915	2021	19915	2021
76	Cook Islands	Oceania	0	0	0	0	19264	2021
77	Anguilla	Americas	0	0	0	0	19216	2021
82	Curaçao	Americas	0	0	17718	2021	14183	2021
85	Montserrat	Americas	0	0	0	0	16199	2021
86	American Samoa	Oceania	0	0	15743	2021	0	0
104	Cuba	Americas	0	0	9500	2020	11255	2021
196	Zanzibar	Africa	0	0	0	0	1211	2021
204	Syria	Asia	0	0	533	2020	925	2021
212	North Korea	Asia	0	0	0	0	654	2021



.x}



< >



61	Northern Mariana Islands	Oceania	0	0	23707	2019	0	0
65	Saint Martin (French part)	Americas	0	0	21921	2014	0	0
68	Turks and Caicos Islands	Americas	0	0	20909	2021	20909	2021
71	French Polynesia	Oceania	0	0	19915	2021	19915	2021
76	Cook Islands	Oceania	0	0	0	0	19264	2021
77	Anguilla	Americas	0	0	0	0	19216	2021
82	Curaçao	Americas	0	0	17718	2021	14183	2021
85	Montserrat	Americas	0	0	0	0	16199	2021
86	American Samoa	Oceania	0	0	15743	2021	0	0
104	Cuba	Americas	0	0	9500	2020	11255	2021
196	Zanzibar	Africa	0	0	0	0	1211	2021
204	Syria	Asia	0	0	533	2020	925	2021
212	North Korea	Asia	0	0	0	0	654	2021

✓ 0s completed at 12:51 PM

## Country with highest UN Estimate



### Which country has highest UN Estimate?

```
df[["UN_Estimate"]].max()
```

```
0  
UN_Estimate 234317  
dtype: int64
```

```
[84] df[['Country/Territory', 'UN_Estimate']].max()
```

```
0  
Country/Territory Zimbabwe  
UN_Estimate 234317  
dtype: object
```

✓ 0s completed at 12:56 PM

Country with highest world bank estimate

### Which country has highest Worlbank Estimate?

```
[85] df["WorldBank_Estimate"].max()
```

```
234316
```

```
df[["Country/Territory", "WorldBank_Estimate"]].max()
```

```
0  
Country/Territory Zimbabwe  
WorldBank_Estimate 234316  
dtype: object
```

Country with highest IMF estimate

## Which country has highest IMF Estimate?

```
[90] df["IMF_Estimate"].max()
```

```
132372
```

```
df[["Country/Territory", "IMF_Estimate"]].max()
```

```
0
```

Country/Territory	Zimbabwe
IMF_Estimate	132372

dtype: object

## Replace 0 with null values

```
[ ] # replace 0 with null values
```

```
[106] df = df.replace(0, np.nan)  
df
```

	Country/Territory	UN_Region	IMF_Estimate	IMF_Year	WorldBank_Estimate	WorldBank_Year	UN_Estimate	UN_Year
1	Monaco	Europe	NaN	NaN	234316.0	2021.0	234317.0	2021
2	Liechtenstein	Europe	NaN	NaN	157755.0	2020.0	169260.0	2021
3	Luxembourg	Europe	132372.0	2023.0	133590.0	2021.0	133745.0	2021
4	Ireland	Europe	114581.0	2023.0	100172.0	2021.0	101109.0	2021
5	Bermuda	Americas	NaN	NaN	114090.0	2021.0	112653.0	2021
...	...	...	...	...	...	...	...	...
219	Malawi	Africa	496.0	2023.0	635.0	2021.0	613.0	2021
220	South Sudan	Africa	467.0	2023.0	1072.0	2015.0	400.0	2021
221	Sierra Leone	Africa	415.0	2023.0	480.0	2021.0	505.0	2021
222	Afghanistan	Asia	611.0	2020.0	369.0	2021.0	373.0	2021
223	Burundi	Africa	249.0	2023.0	222.0	2021.0	311.0	2021

223 rows x 8 columns

## The average of world bank estimate and UN estimate

```
# Calculate the average of 'Worldbank_Estimate' and 'UN_Estimate' columns
df["avg_worldbank_un"] = df[["WorldBank_Estimate", "UN_Estimate"]].mean(axis=1)
df
```

	Country/Territory	UN_Region	IMF_Estimate	IMF_Year	WorldBank_Estimate	WorldBank_Year	UN_Estimate	UN_Year	avg_worldbank_un
1	Monaco	Europe	234316.5	NaN	234316.0	2021.0	234317.0	2021	234316.5
2	Liechtenstein	Europe	163507.5	NaN	157755.0	2020.0	169260.0	2021	163507.5
3	Luxembourg	Europe	132372.0	2023.0	133590.0	2021.0	133745.0	2021	133667.5
4	Ireland	Europe	114581.0	2023.0	100172.0	2021.0	101109.0	2021	100640.5
5	Bermuda	Americas	113371.5	NaN	114090.0	2021.0	112653.0	2021	113371.5
...	...	...	...	...	...	...	...	...	...
219	Malawi	Africa	496.0	2023.0	635.0	2021.0	613.0	2021	624.0
220	South Sudan	Africa	467.0	2023.0	1072.0	2015.0	400.0	2021	736.0
221	Sierra Leone	Africa	415.0	2023.0	480.0	2021.0	505.0	2021	492.5
222	Afghanistan	Asia	611.0	2020.0	369.0	2021.0	373.0	2021	371.0
223	Burundi	Africa	249.0	2023.0	222.0	2021.0	311.0	2021	266.5

223 rows x 9 columns

## Fill the null values in IMF column with the calculated average

```
[114] # Fill the null values in 'imf' column with the calculated average
df["IMF_Estimate"] = df["IMF_Estimate"].fillna(avg_worldbank_un)
df
```

	Country/Territory	UN_Region	IMF_Estimate	IMF_Year	WorldBank_Estimate	WorldBank_Year	UN_Estimate	UN_Year	avg_worldbank_un
1	Monaco	Europe	234316.5	NaN	234316.0	2021.0	234317.0	2021	234316.5
2	Liechtenstein	Europe	163507.5	NaN	157755.0	2020.0	169260.0	2021	163507.5
3	Luxembourg	Europe	132372.0	2023.0	133590.0	2021.0	133745.0	2021	133667.5
4	Ireland	Europe	114581.0	2023.0	100172.0	2021.0	101109.0	2021	100640.5
5	Bermuda	Americas	113371.5	NaN	114090.0	2021.0	112653.0	2021	113371.5
...	...	...	...	...	...	...	...	...	...
219	Malawi	Africa	496.0	2023.0	635.0	2021.0	613.0	2021	624.0
220	South Sudan	Africa	467.0	2023.0	1072.0	2015.0	400.0	2021	736.0
221	Sierra Leone	Africa	415.0	2023.0	480.0	2021.0	505.0	2021	492.5
222	Afghanistan	Asia	611.0	2020.0	369.0	2021.0	373.0	2021	371.0
223	Burundi	Africa	249.0	2023.0	222.0	2021.0	311.0	2021	266.5

223 rows x 9 columns

## Drop the temporary 'avg\_worldbank\_un' column

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

```
df = df.drop(columns=['avg_worldbank_un'])
df
```

	Country/Territory	UN_Region	IMF_Estimate	IMF_Year	WorldBank_Estimate	WorldBank_Year	UN_Estimate	UN_Year
1	Monaco	Europe	234316.5	NaN	234316.0	2021.0	234317.0	2021
2	Liechtenstein	Europe	163507.5	NaN	157755.0	2020.0	169260.0	2021
3	Luxembourg	Europe	132372.0	2023.0	133590.0	2021.0	133745.0	2021
4	Ireland	Europe	114581.0	2023.0	100172.0	2021.0	101109.0	2021
5	Bermuda	Americas	113371.5	NaN	114090.0	2021.0	112653.0	2021
...	...	...	...	...	...	...	...	...
219	Malawi	Africa	496.0	2023.0	635.0	2021.0	613.0	2021
220	South Sudan	Africa	467.0	2023.0	1072.0	2015.0	400.0	2021
221	Sierra Leone	Africa	415.0	2023.0	480.0	2021.0	505.0	2021
222	Afghanistan	Asia	611.0	2020.0	369.0	2021.0	373.0	2021
223	Burundi	Africa	249.0	2023.0	222.0	2021.0	311.0	2021

223 rows x 8 columns

## Checking Missing Values

```
[114] newdf = df.bfill()
print(newdf.to_string())
```

	Country/Territory	UN_Region	IMF_Estimate	IMF_Year	WorldBank_Estimate	WorldBank_Year	UN_Estimate	UN_Year
1	Monaco	Europe	234316.5	2023.0	234316.0	2021.0	234317.0	2021
2	Liechtenstein	Europe	163507.5	2023.0	157755.0	2020.0	169260.0	2021
3	Luxembourg	Europe	132372.0	2023.0	133590.0	2021.0	133745.0	2021
4	Ireland	Europe	114581.0	2023.0	100172.0	2021.0	101109.0	2021
5	Bermuda	Americas	113371.5	2023.0	114090.0	2021.0	112653.0	2021
6	Norway	Europe	101103.0	2023.0	89154.0	2021.0	89242.0	2021
7	Switzerland	Europe	98767.0	2023.0	91992.0	2021.0	93525.0	2021
8	Singapore	Asia	91100.0	2023.0	72794.0	2021.0	66822.0	2021
9	Isle of Man	Europe	87158.0	2023.0	87158.0	2019.0	85250.0	0
10	Cayman Islands	Americas	85909.5	2023.0	86569.0	2021.0	85250.0	2021
11	Qatar	Asia	83891.0	2023.0	66838.0	2021.0	66799.0	2021
12	United States	Americas	80034.0	2023.0	70249.0	2021.0	69185.0	2021
13	Iceland	Europe	75180.0	2023.0	68728.0	2021.0	69133.0	2021
14	Channel Islands	Europe	75153.0	2023.0	75153.0	2007.0	68037.0	0
15	Faroe Islands	Europe	69010.0	2023.0	69010.0	2021.0	68037.0	0
16	Denmark	Europe	68827.0	2023.0	68008.0	2021.0	68037.0	2021
17	Australia	Oceania	64964.0	2023.0	60443.0	2021.0	66916.0	2021
18	Netherlands	Europe	61098.0	2023.0	57768.0	2021.0	57871.0	2021
19	Greenland	Americas	56378.0	2023.0	54571.0	2020.0	58185.0	2021
20	Austria	Europe	56802.0	2023.0	53638.0	2021.0	53840.0	2021
21	Israel	Asia	55535.0	2023.0	52171.0	2021.0	54111.0	2021
22	Sweden	Europe	55395.0	2023.0	61029.0	2021.0	60730.0	2021
23	Finland	Europe	54351.0	2023.0	53655.0	2021.0	53703.0	2021
24	Belgium	Europe	53377.0	2023.0	51247.0	2021.0	51166.0	2021
25	San Marino	Europe	52949.0	2023.0	45320.0	2020.0	50425.0	2021
26	Canada	Americas	52722.0	2023.0	51988.0	2021.0	52112.0	2021
27	Hong Kong	Asia	52429.0	2023.0	49801.0	2021.0	49259.0	2021
28	Germany	Europe	51383.0	2023.0	51204.0	2021.0	51073.0	2021
29	Macau	Asia	50571.0	2023.0	43874.0	2021.0	43555.0	2021
30	United Arab Emirates	Asia	49451.0	2023.0	44316.0	2021.0	43295.0	2021
31	British Virgin Islands	Americas	49444.0	2023.0	48781.0	2021.0	49444.0	2021
32	New Zealand	Oceania	48826.0	2023.0	48781.0	2021.0	48824.0	2021
33	United Kingdom	Europe	46371.0	2023.0	46510.0	2021.0	46542.0	2021
34	France	Europe	44408.0	2023.0	43659.0	2021.0	44229.0	2021
35	Andorra	Europe	44387.0	2023.0	42137.0	2021.0	42066.0	2021
36	European Union[n 1]	Europe	39940.0	2023.0	38411.0	2021.0	31875.0	2021
37	US Virgin Islands	Americas	39552.0	2023.0	39552.0	2020.0	32716.0	0
38	Puerto Rico	Americas	38567.0	2023.0	32641.0	2021.0	32716.0	2021
39	New Caledonia	Oceania	36077.0	2023.0	37160.0	2021.0	34994.0	2021
40	Malta	Europe	36989.0	2023.0	33487.0	2021.0	33642.0	2021
41	Italy	Europe	36812.0	2023.0	35658.0	2021.0	35579.0	2021

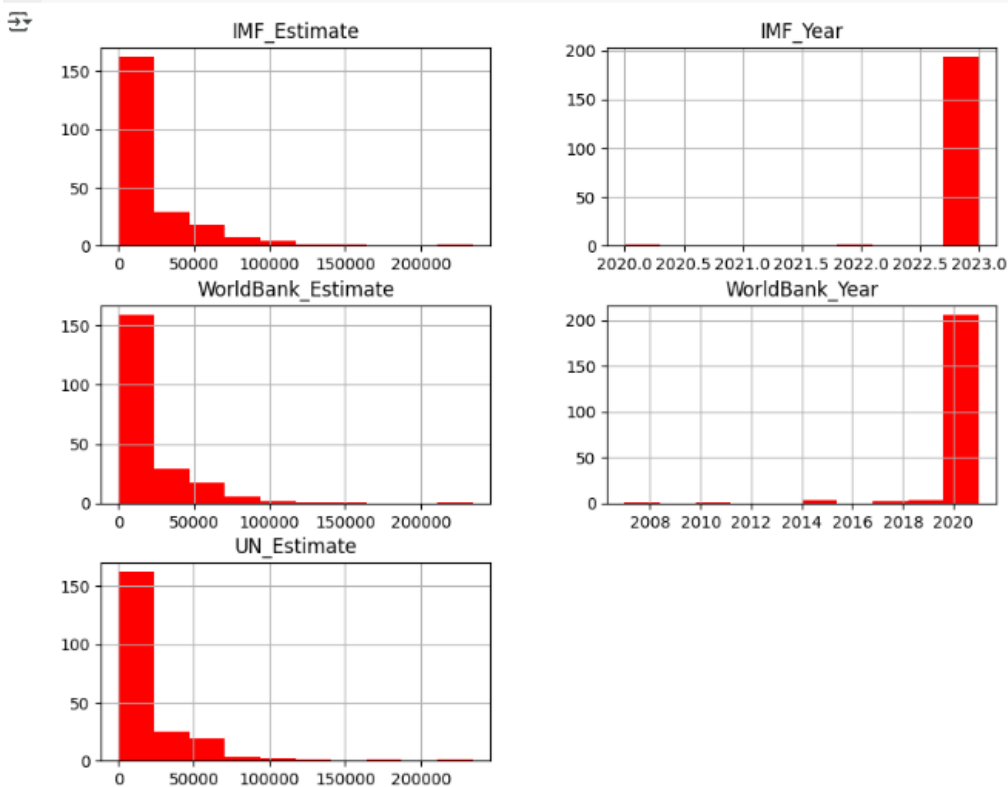


```
[115]
newdf = df.ffill()

print(newdf.to_string())
```

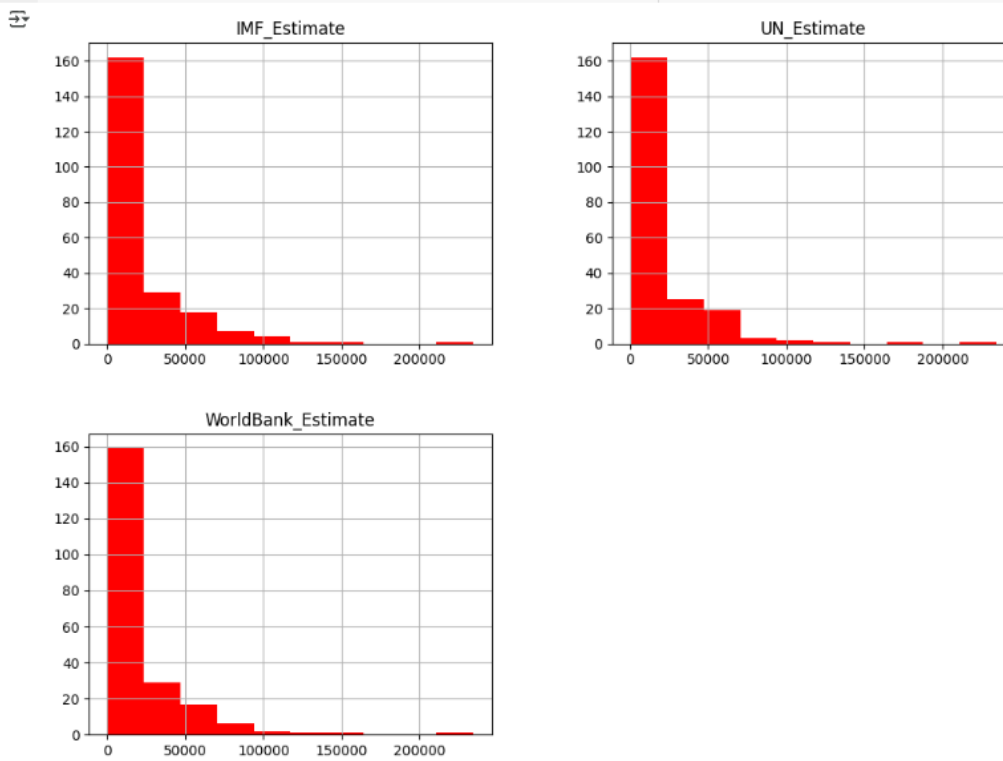
166	São Tomé and Príncipe	Africa	2695.0	2023.0	2361.0	2021.0	2486.0	2021
167	Ivory Coast	Africa	2646.0	2023.0	2549.0	2021.0	2539.0	2021
168	India	Asia	2681.0	2023.0	2257.0	2021.0	2274.0	2021
169	Nicaragua	Americas	2589.0	2023.0	2046.0	2021.0	2046.0	2021
170	Congo	Africa	2584.0	2023.0	2290.0	2021.0	2200.0	2021
171	Uzbekistan	Asia	2563.0	2023.0	1983.0	2021.0	2032.0	2021
172	Mauritania	Africa	2475.0	2023.0	2166.0	2021.0	2166.0	2021
173	Bangladesh	Asia	2469.0	2023.0	2458.0	2021.0	2450.0	2021
174	Solomon Islands	Oceania	2299.0	2023.0	2305.0	2021.0	2306.0	2021
175	Nigeria	Africa	2280.0	2023.0	2066.0	2021.0	2019.0	2021
176	Kenya	Africa	2269.0	2023.0	2082.0	2021.0	2082.0	2021
177	Haiti	Americas	2173.0	2023.0	1830.0	2021.0	1664.0	2021
178	Ghana	Africa	2024.0	2023.0	2363.0	2021.0	2409.0	2021
179	Kiribati	Oceania	1989.0	2023.0	1607.0	2021.0	1765.0	2021
180	Cambodia	Asia	1896.0	2023.0	1625.0	2021.0	1608.0	2021
181	Laos	Asia	1858.0	2023.0	2536.0	2021.0	2569.0	2021
182	Zimbabwe	Africa	1851.0	2023.0	1774.0	2021.0	1508.0	2021
183	Kyrgyzstan	Asia	1736.0	2023.0	1277.0	2021.0	1339.0	2021
184	Senegal	Africa	1719.0	2023.0	1637.0	2021.0	1637.0	2021
185	Cameroon	Africa	1699.0	2023.0	1667.0	2021.0	1668.0	2021
186	Pakistan	Asia	1658.0	2022.0	1505.0	2021.0	1480.0	2021
187	Guinea	Africa	1549.0	2023.0	1189.0	2021.0	1185.0	2021
188	Ethiopia	Africa	1475.0	2023.0	925.0	2021.0	825.0	2021
189	Timor-Leste	Asia	1425.0	2023.0	2741.0	2021.0	1517.0	2021
190	Zambia	Africa	1423.0	2023.0	1137.0	2021.0	1095.0	2021
191	Benin	Africa	1390.0	2023.0	1319.0	2021.0	1361.0	2021
192	Nepal	Asia	1377.0	2023.0	1208.0	2021.0	1159.0	2021
193	Comoros	Africa	1360.0	2023.0	1578.0	2021.0	1631.0	2021
194	Tanzania	Africa	1348.0	2023.0	1099.0	2021.0	1136.0	[n 10]2021
195	Tajikistan	Asia	1277.0	2023.0	897.0	2021.0	897.0	2021
196	Zanzibar	Africa	1211.0	2023.0	897.0	2021.0	1211.0	2021
197	Lesotho	Africa	1208.0	2023.0	1094.0	2021.0	1040.0	2021
198	Myanmar	Asia	1180.0	2023.0	1210.0	2021.0	1089.0	2021
199	Uganda	Africa	1105.0	2023.0	884.0	2021.0	930.0	2021
200	Togo	Africa	990.0	2023.0	973.0	2021.0	944.0	2021
201	Sudan	Africa	975.0	2023.0	752.0	2021.0	786.0	2021
202	Guinea-Bissau	Africa	974.0	2023.0	795.0	2021.0	759.0	2021
203	Rwanda	Africa	970.0	2023.0	822.0	2021.0	822.0	2021
204	Syria	Asia	729.0	2023.0	533.0	2020.0	925.0	2021
205	Burkina Faso	Africa	900.0	2023.0	893.0	2021.0	893.0	2021
206	Mali	Africa	889.0	2023.0	874.0	2021.0	875.0	2021
207	Gambia	Africa	861.0	2023.0	772.0	2021.0	772.0	2021
208	Liberia	Africa	805.0	2023.0	676.0	2021.0	471.0	2021
209	Eritrea	Africa	715.0	2023.0	644.0	2011.0	623.0	2021
210	DR Congo	Africa	695.0	2023.0	577.0	2021.0	551.0	2021
211	Chad	Africa	667.0	2023.0	686.0	2021.0	955.0	2021
212	North Korea	Asia	654.0	2023.0	686.0	2021.0	654.0	2021

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

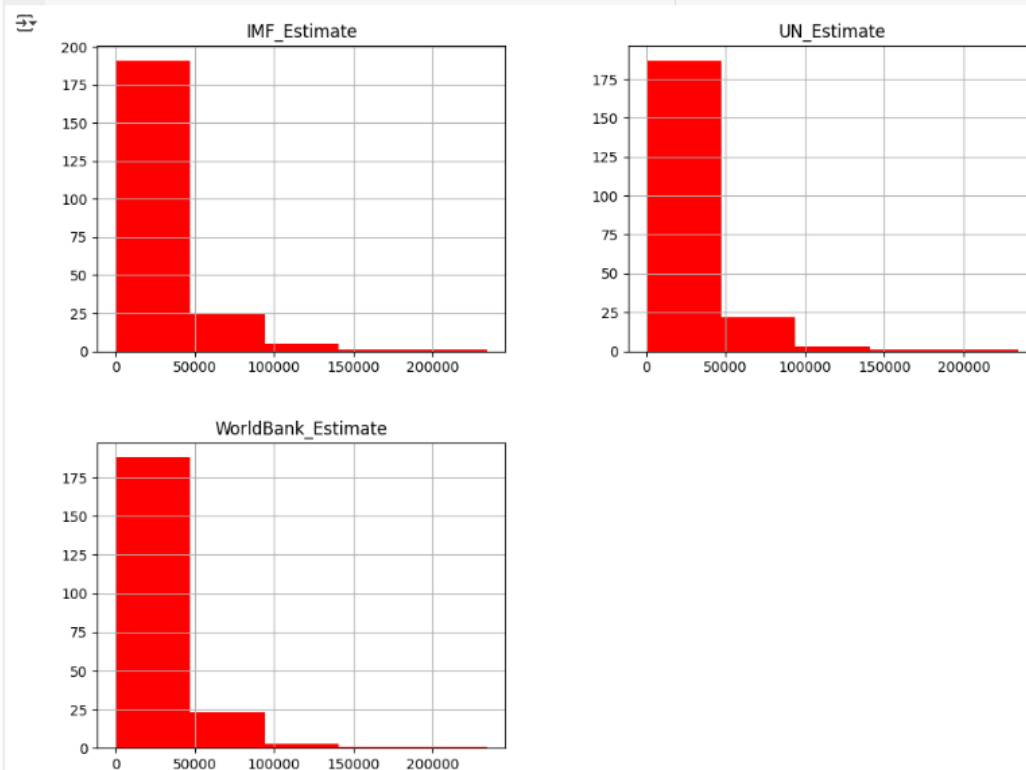




```
df[["IMF_Estimate", "UN_Estimate", "WorldBank_Estimate"]].hist(figsize=(12,9), color="red")
plt.show()
plt.show()
```



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



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

```
WorldBank_Estimate
```

min	222.0
max	234316.0

dtype: float64

```
✓ [60] 234316/5  
0s #1 bin size if bins=5
```

```
→ 46863.2
```

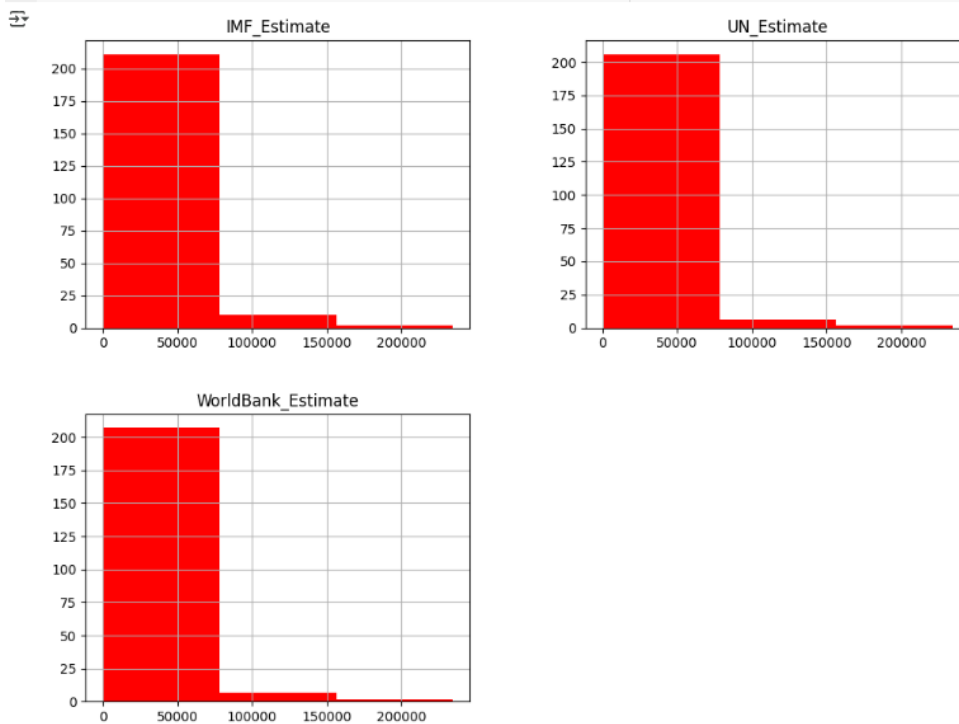
```
✓ [61] df[df["WorldBank_Estimate"]<=46863.2]["WorldBank_Estimate"].count()  
0s
```

```
→ 188
```

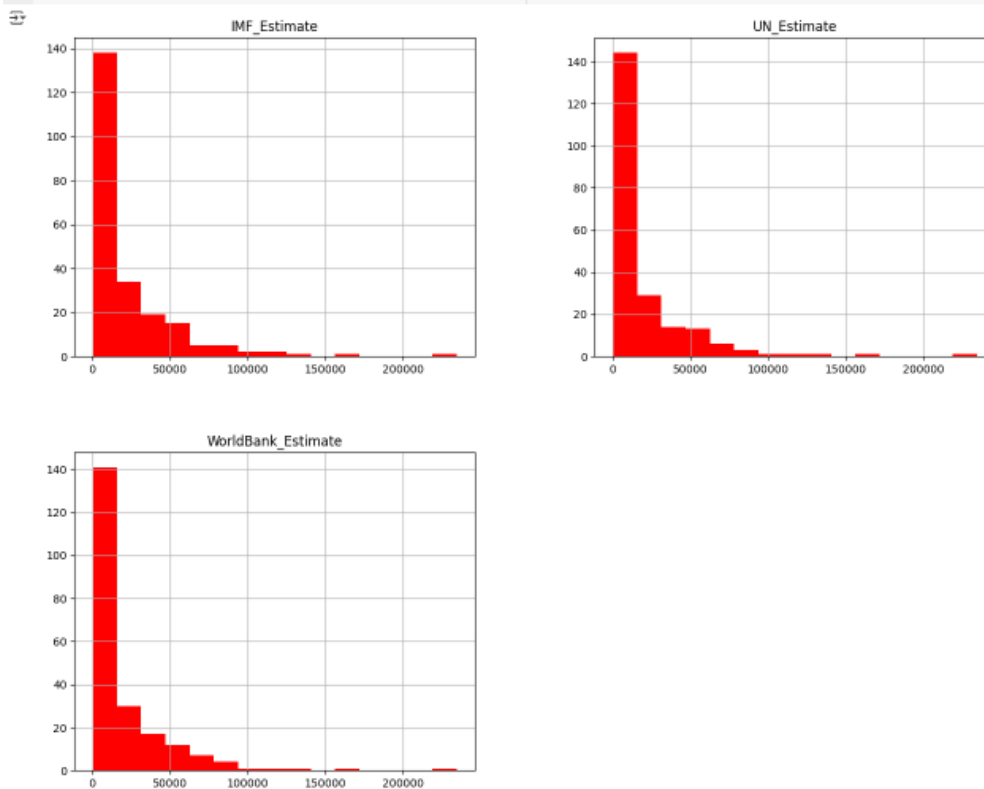
```
✓ [62] 234316/10  
0s #1 bin size if bins not given any number
```

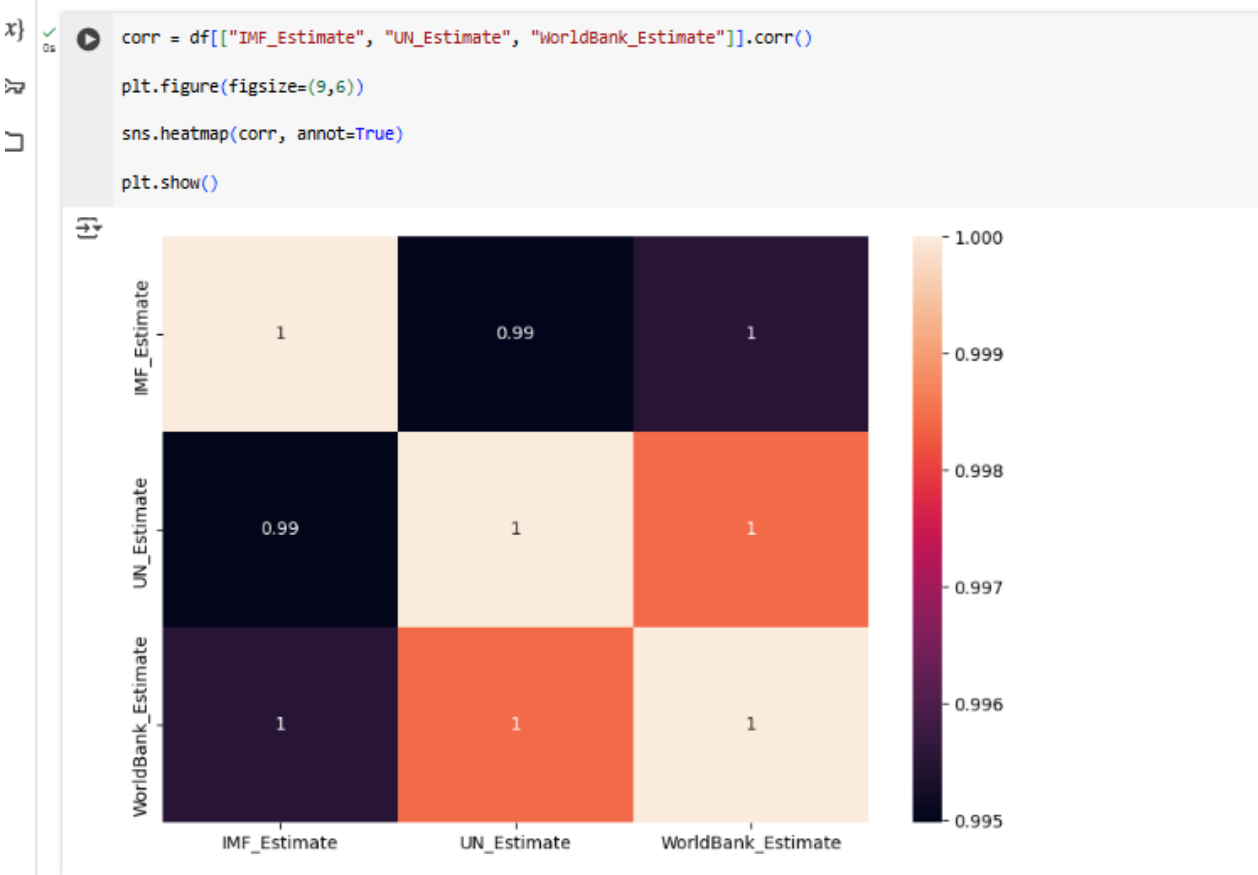
```
→ 23431.6
```

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



```
df[["IMF_Estimate", "UN_Estimate", "WorldBank_Estimate"]].hist(bins=15, figsize=(15,12), color="red")  
#23400/15 = 15300  
plt.show()
```



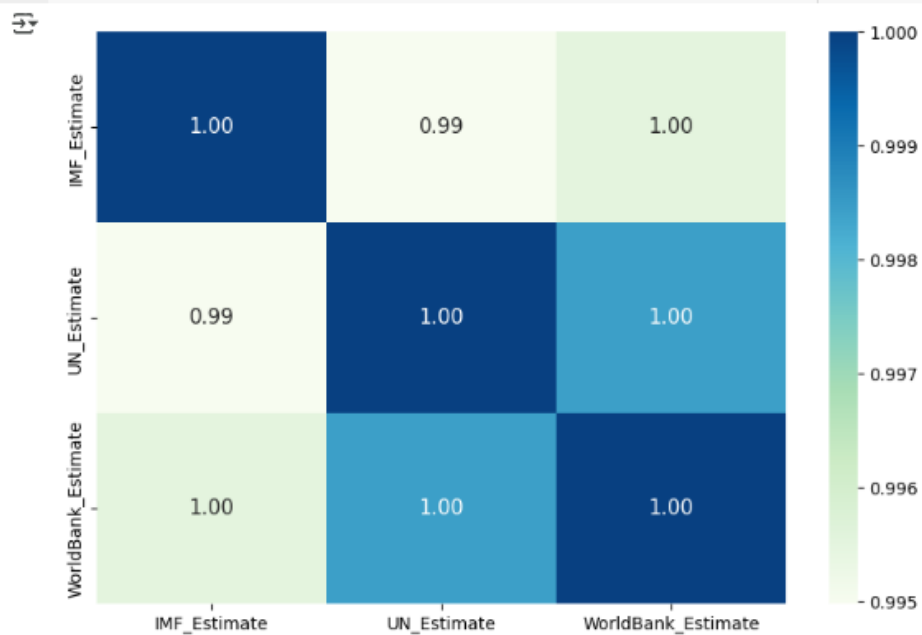


```
corr = df[["IMF_Estimate", "UN_Estimate", "WorldBank_Estimate"]].corr()

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

sns.heatmap(corr, annot=True, fmt=".2f", cmap = 'GnBu', annot_kws={"size": 12})

plt.show()
```



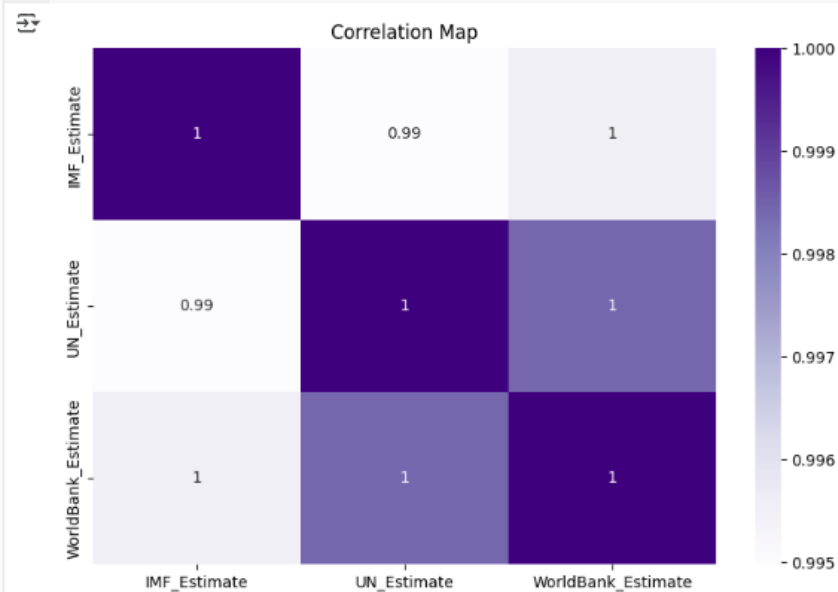
```
corr = df[["IMF_Estimate", "UN_Estimate", "WorldBank_Estimate"]].corr()

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

sns.heatmap(corr, annot=True, cmap = 'Purples')

plt.title("Correlation Map")

plt.show()
```



```

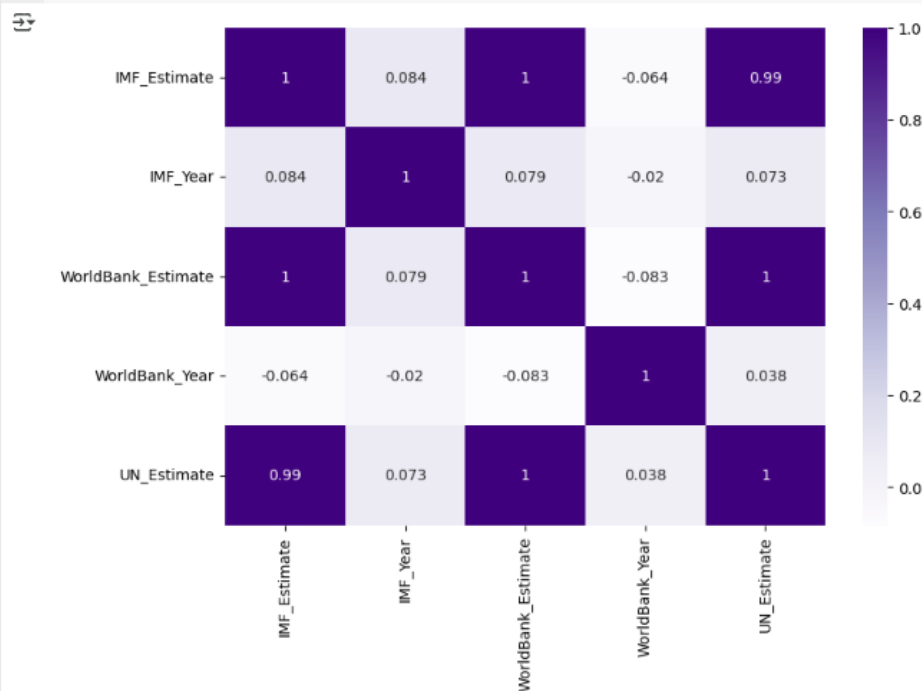
corr = df.select_dtypes(include=[int, float]).corr()

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

sns.heatmap(corr, annot=True, cmap = 'Purples')

plt.show()

```



```
[72] df.head()
```

	Country/Territory	UN_Region	IMF_Estimate	IMF_Year	WorldBank_Estimate	WorldBank_Year	UN_Estimate	UN_Year
1	Monaco	Europe	234316.5	NaN	234316.0	2021.0	234317.0	2021
2	Liechtenstein	Europe	163507.5	NaN	157755.0	2020.0	169260.0	2021
3	Luxembourg	Europe	132372.0	2023.0	133590.0	2021.0	133745.0	2021
4	Ireland	Europe	114581.0	2023.0	100172.0	2021.0	101109.0	2021
5	Bermuda	Americas	113371.5	NaN	114090.0	2021.0	112653.0	2021

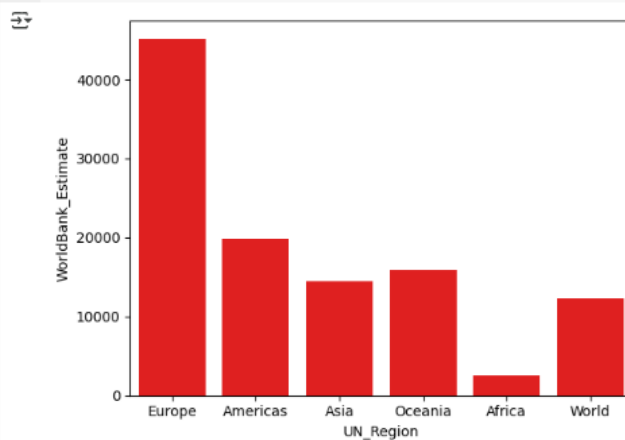
Next steps: [Generate code with df](#) [View recommended plots](#) [New interactive sheet](#)

```

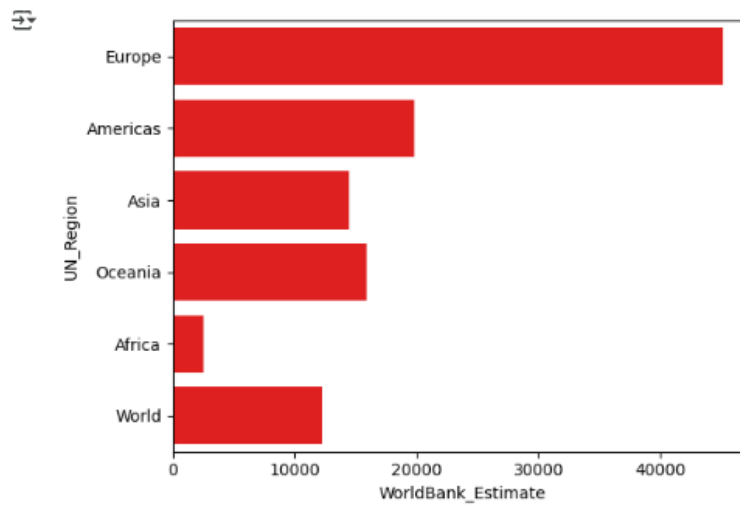
sns.barplot(x="UN_Region", y="WorldBank_Estimate", data=df, errorbar=None, color="red")

plt.show()

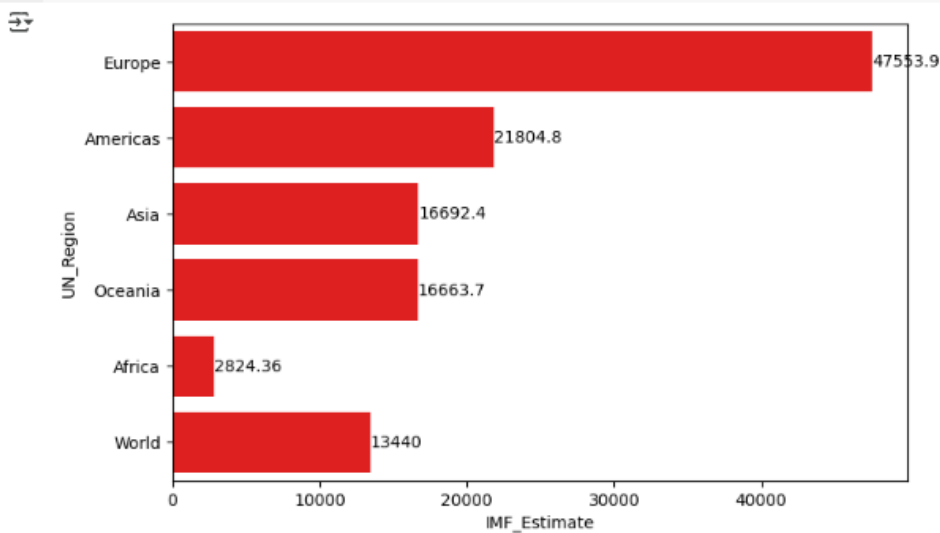
```



```
[76] sns.barplot(x="WorldBank_Estimate", y="UN_Region", data=df, errorbar=None, color="red")
plt.show()
```



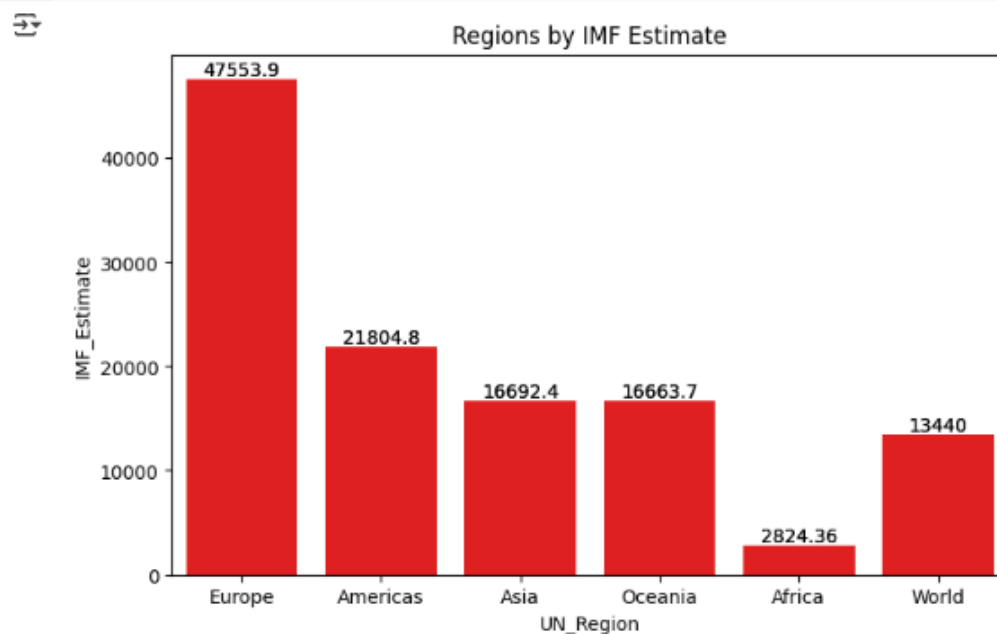
```
fig = plt.figure(figsize = (8,5))
ax = sns.barplot(x = "IMF_Estimate", y = "UN_Region",
data = df, errorbar = None, color="red")
ax.bar_label(ax.containers[0])
plt.show()
```



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

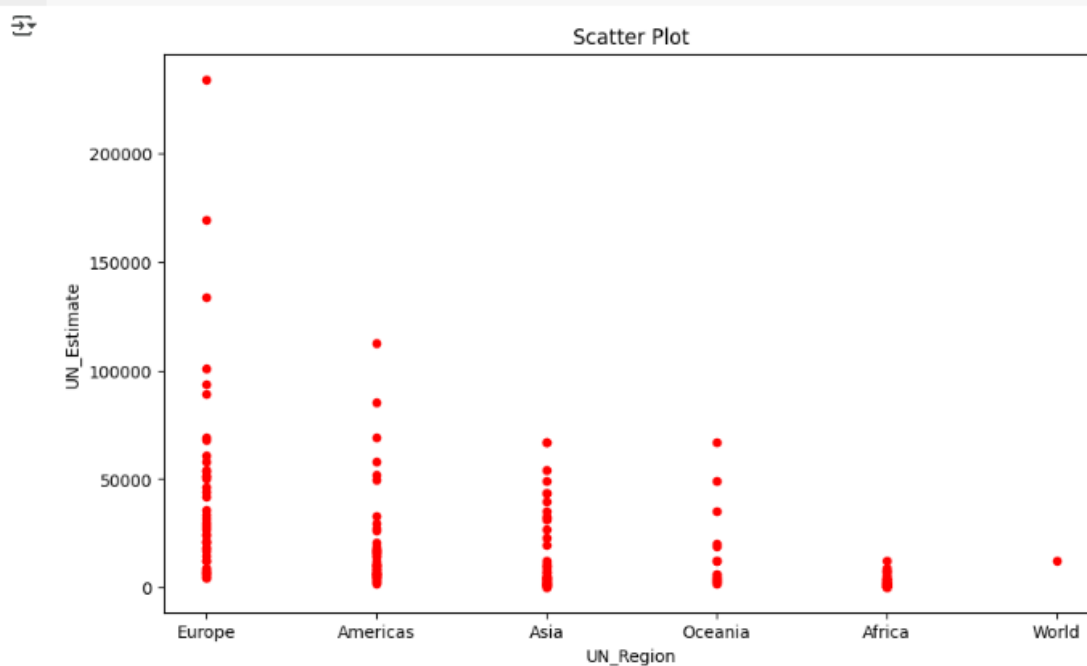
ax.bar_label(ax.containers[0])

ax.set_title("Regions by IMF Estimate")
plt.show()
```

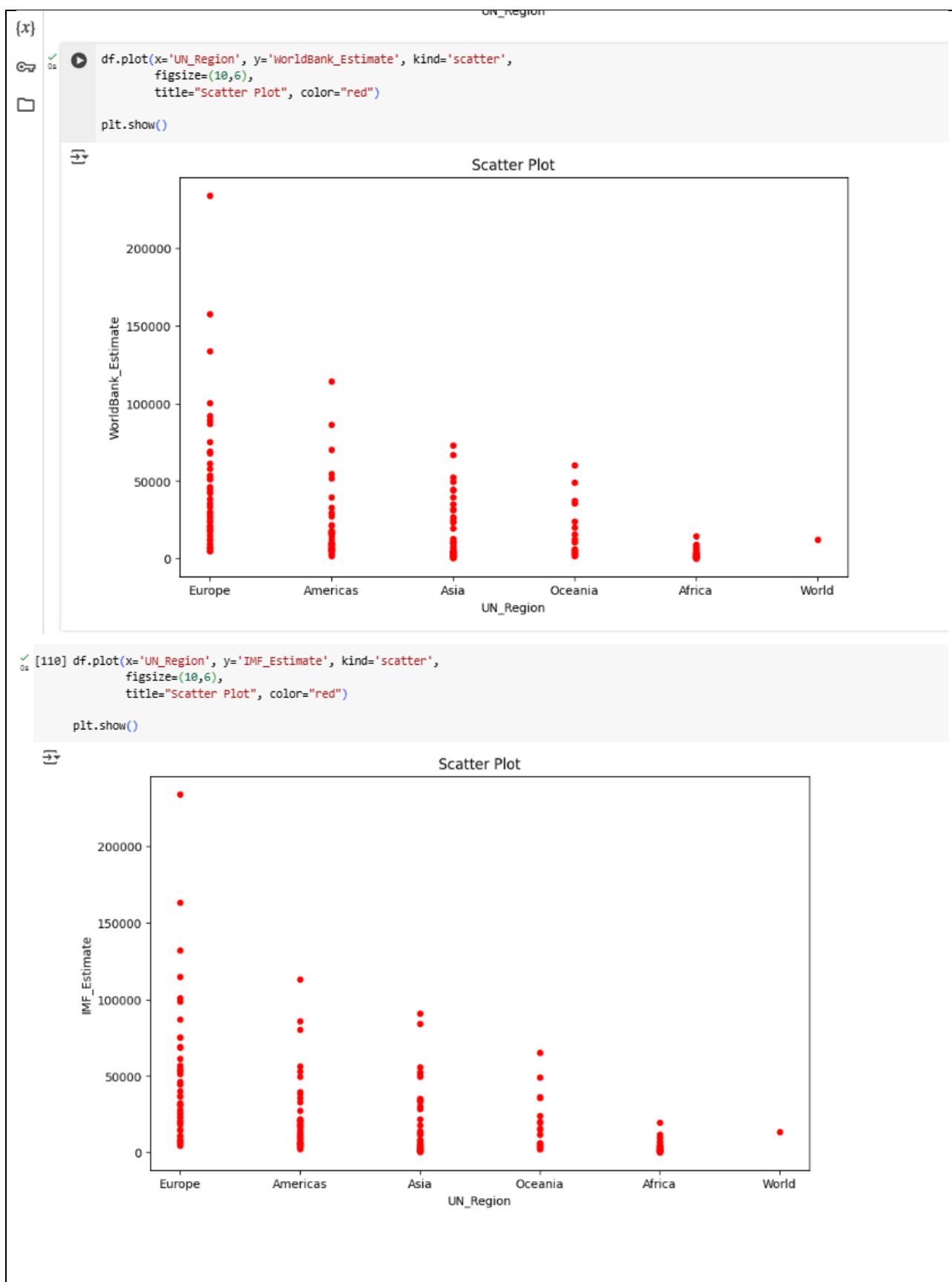


```
df.plot(x='UN_Region', y='UN_Estimate', kind='scatter',
        figsize=(10,6),
        title="Scatter Plot", color="red")

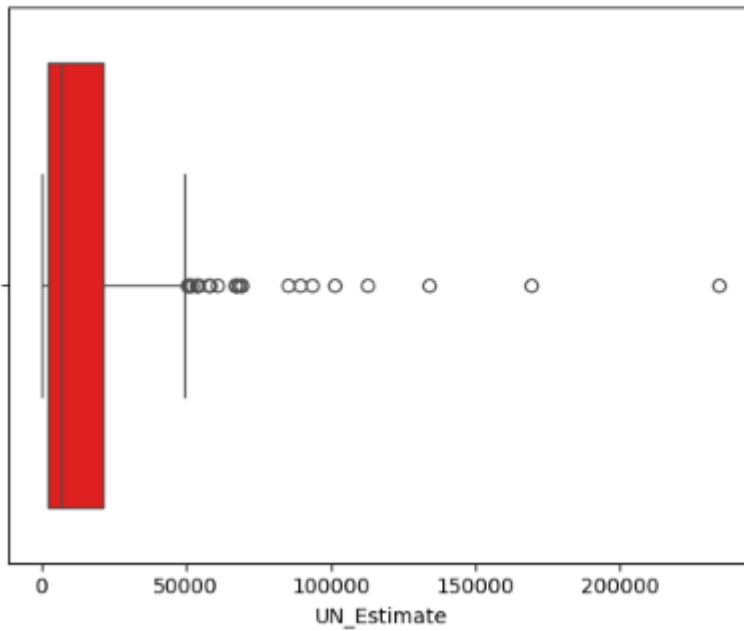
plt.show()
```







```
[81] sns.boxplot(x=df["UN_Estimate"], color="red")
plt.show()
```

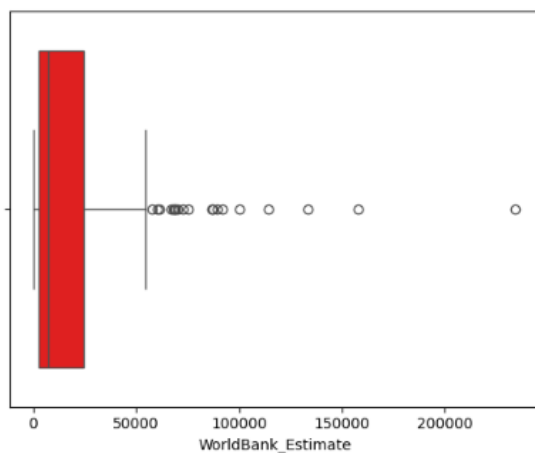


```
[82] df[df["UN_Estimate"]>50000].head()
```



	Country/Territory	UN_Region	IMF_Estimate	IMF_Year	WorldBank_Estimate	WorldBank_Year	UN_Estimate	UN_Year
1	Monaco	Europe	234316.5	NaN	234316.0	2021.0	234317.0	2021
2	Liechtenstein	Europe	163507.5	NaN	157755.0	2020.0	169260.0	2021
3	Luxembourg	Europe	132372.0	2023.0	133590.0	2021.0	133745.0	2021
4	Ireland	Europe	114581.0	2023.0	100172.0	2021.0	101109.0	2021
5	Bermuda	Americas	113371.5	NaN	114090.0	2021.0	112653.0	2021

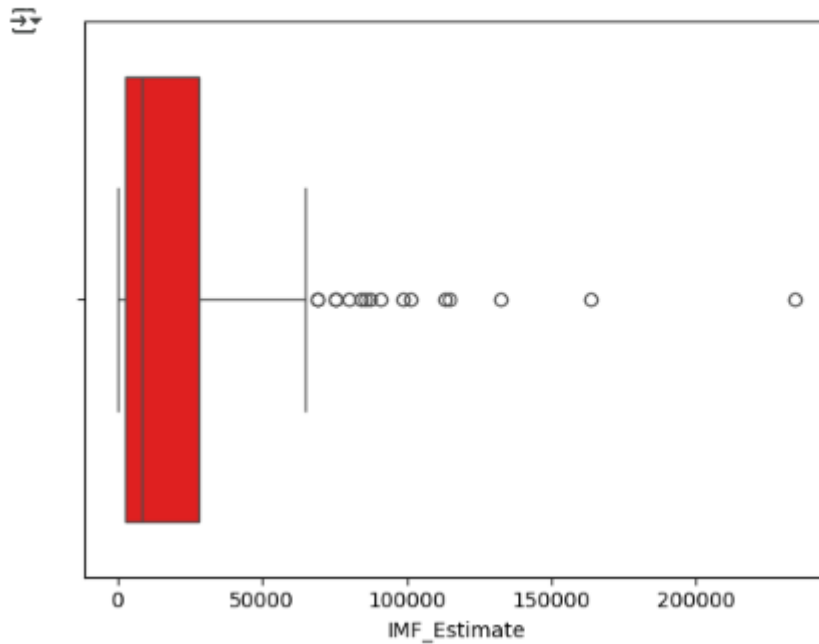
```
sns.boxplot(x=df["WorldBank_Estimate"], color="red")
plt.show()
```



```

sns.boxplot(x=df["IMF_Estimate"], color="red")
plt.show()

```



```

[86] df[df["UN_Estimate"]>100000]

```

	Country/Territory	UN_Region	IMF_Estimate	IMF_Year	WorldBank_Estimate	WorldBank_Year	UN_Estimate	UN_Year
1	Monaco	Europe	234316.5	NaN	234316.0	2021.0	234317.0	2021
2	Liechtenstein	Europe	163507.5	NaN	157755.0	2020.0	169260.0	2021
3	Luxembourg	Europe	132372.0	2023.0	133590.0	2021.0	133745.0	2021
4	Ireland	Europe	114581.0	2023.0	100172.0	2021.0	101109.0	2021
5	Bermuda	Americas	113371.5	NaN	114090.0	2021.0	112653.0	2021

```

[87] df.UN_Estimate.mean()

```

```

18514.528037383177

```

```

df.shape

```

```

(223, 8)

```

{x}

▼ Create another dataframe called data excluding 5 countries with highest UN estimate

```
[89] data = df[-(df["UN_Estimate"]>100000)]
```

```
[90] data.head()
```

	Country/Territory	UN_Region	IMF_Estimate	IMF_Year	WorldBank_Estimate	WorldBank_Year	UN_Estimate	UN_Year
6	Norway	Europe	101103.0	2023.0	89154.0	2021.0	89242.0	2021
7	Switzerland	Europe	98767.0	2023.0	91992.0	2021.0	93525.0	2021
8	Singapore	Asia	91100.0	2023.0	72794.0	2021.0	66822.0	2021
9	Isle of Man	Europe	87158.0	NaN	87158.0	2019.0	NaN	0
10	Cayman Islands	Americas	85909.5	NaN	86569.0	2021.0	85250.0	2021

Next steps: [Generate code with data](#) [View recommended plots](#) [New interactive sheet](#)

```
[91] data.shape
```

```
(218, 8)
```

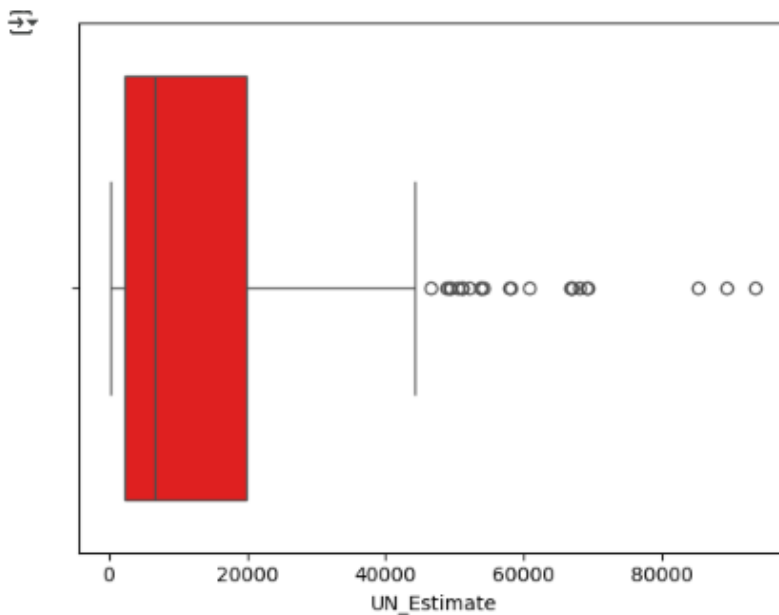
```
[92] data.UN_Estimate.mean()
```

```
15363.755980861244
```

```
[93] df.UN_Estimate.mean()
```

```
18514.528037383177
```

```
[95] sns.boxplot(x=data["UN_Estimate"], color="red")  
plt.show()
```





## Removing outliers

image.png

image.png

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

2331.75

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

21359.25

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

19027.5

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

49900.5

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

-26209.5

```
[101] df_filtered = df[(df["UN_Estimate"] < upper_boundary) & (df["UN_Estimate"] > lower_boundary)]
```

```
[102] df_filtered.head()
```

	Country/Territory	UN_Region	IMF_Estimate	IMF_Year	WorldBank_Estimate	WorldBank_Year	UN_Estimate	UN_Year
27	Hong Kong	Asia	52429.0	2023.0	49801.0	2021.0	49259.0	2021
29	Macau	Asia	50571.0	2023.0	43874.0	2021.0	43555.0	2021
30	United Arab Emirates	Asia	49451.0	2023.0	44316.0	2021.0	43295.0	2021
31	British Virgin Islands	Americas	49444.0	NaN	NaN	NaN	49444.0	2021
32	New Zealand	Oceania	48826.0	2023.0	48781.0	2021.0	48824.0	2021

Next steps: [Generate code with df\\_filtered](#) [View recommended plots](#) [New interactive sheet](#)



```
✓ [103] df_filtered.shape  
      # there were 223 rows - 196 = 27 outliers dropped
```

```
↕ (190, 8)
```

```
✓ [104] df_filtered.UN_Estimate.mean()
```

```
↕ 10488.947368421053
```

```
✓ [105] df.UN_Estimate.mean()
```

```
↕ 18514.528037383177
```

```
✓ [106] #how can we create a table with following  
      df_filtered.WorldBank_Estimate.mean()
```

```
↕ 10355.304347826086
```

```
✓ [107] df.WorldBank_Estimate.mean()
```

```
↕ 19540.805555555555
```

```
✓ [108] df_filtered.IMF_Estimate.mean()
```

```
↕ 11697.976315789474
```

```
✓ [109] df.IMF_Estimate.mean()
```

```
↕ 20998.14125560538
```

## Course Notes

It is recommended to take notes from the course, use the space below to do so, or use the revision guide shared with the class:



We have included a range of additional links to further resources and information that you may find useful, these can be found within your revision guide.

## **END OF WORKBOOK**

**Please check through your work thoroughly before submitting and update the table of contents if required.**

**Please send your completed work booklet to your trainer.**

