

Worldwide Average IQ Levels Analysis with Python

Certainly! To explore the "Worldwide Average IQ Levels and Socioeconomic Factors Dataset" for your Jupyter file project, you can use Python along with popular data analysis libraries such as Pandas, NumPy, and Matplotlib. Below is a brief description of how you can approach this project:

Project Description Objective: The goal of this Jupyter file project is to analyze and gain insights into the relationship between average IQ levels and socioeconomic factors across different countries using the "Worldwide Average IQ Levels and Socioeconomic Factors Dataset."

Import Library

```
In [2]: import pandas as pd
```

```
In [3]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import seaborn as sns
```

```
C:\Users\Syed Arif\anaconda3\lib\site-packages\scipy\__init__.py:146: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for this version of SciPy (detected version 1.25.1)
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}")
```

Uploading Csv file

```
In [4]: df = pd.read_csv(r"C:\Users\Syed Arif\Desktop\IQ_level.csv")
```

Data Preprocessing

.head()

head is used show to the By default = 5 rows in the dataset

```
In [5]: df.head()
```

```
Out[5]:
```

	rank	country	IQ	education_expenditure	avg_income	avg_temp
0	1	Hong Kong	106	1283.0	35304.0	26.2
1	2	Japan	106	1340.0	40964.0	19.2
2	3	Singapore	106	1428.0	41100.0	31.5
3	4	Taiwan	106	NaN	NaN	26.9
4	5	China	104	183.0	4654.0	19.1

.tail()

tail is used to show rows by Descending order

```
In [6]: df.tail()
```

```
Out[6]:
```

	rank	country	IQ	education_expenditure	avg_income	avg_temp
103	104	Equatorial Guinea	56	NaN	7625.0	29.9
104	105	Gambia	55	14.0	648.0	32.9
105	106	Guatemala	55	92.0	2830.0	32.1
106	107	Sierra Leone	52	16.0	412.0	30.4
107	108	Nepal	51	22.0	595.0	24.6

.shape

It show the total no of rows & Column in the dataset

```
In [7]: df.shape
```

```
Out[7]: (108, 6)
```

.Columns

It show the no of each Column

```
In [8]: df.columns
```

```
Out[8]: Index(['rank', 'country', 'IQ', 'education_expenditure', 'avg_income',  
              'avg_temp'],  
              dtype='object')
```

.dtypes

This Attribute show the data type of each column

```
In [9]: df.dtypes
```

```
Out[9]: rank                int64
country                object
IQ                    int64
education_expenditure  float64
avg_income             float64
avg_temp              float64
dtype: object
```

.unique()

In a column, It show the unique value of specific column.

```
In [10]: df["country"].unique()
```

```
Out[10]: array(['Hong Kong\xa0', 'Japan', 'Singapore', 'Taiwan\xa0', 'China',
'South Korea', 'Netherlands', 'Finland', 'Canada', 'North Korea',
'Luxembourg', 'Macao\xa0', 'Germany', 'Switzerland', 'Estonia',
'Australia', 'United Kingdom', 'Greenland\xa0', 'Iceland',
'Austria', 'Hungary', 'New Zealand', 'Belgium', 'Norway', 'Sweden',
'Denmark', 'Cambodia', 'France', 'United States', 'Poland',
'Czechia', 'Russia', 'Spain', 'Ireland', 'Italy', 'Croatia',
'Lithuania', 'Israel', 'Mongolia', 'Portugal', 'Bermuda\xa0',
'Bulgaria', 'Greece', 'Ukraine', 'Vietnam', 'Kazakhstan',
'Malaysia', 'Myanmar', 'Thailand', 'Serbia', 'Brunei', 'Chile',
'Costa Rica', 'Iraq', 'Romania', 'Argentina', 'Mauritius',
'Mexico', 'Turkey', 'Georgia', 'Sri Lanka', 'Montenegro', 'Cuba',
'Brazil', 'Philippines', 'Colombia', 'Laos', 'Venezuela',
'Albania', 'United Arab Emirates', 'Dominican Republic',
'Puerto Rico\xa0', 'Afghanistan', 'Iran', 'Pakistan', 'Indonesia',
'Kuwait', 'Oman', 'Qatar', 'Bolivia', 'Ecuador', 'Egypt',
'Algeria', 'India', 'Saudi Arabia', 'Sudan', 'Syria', 'Bangladesh',
'Chad', 'East Timor', 'Kenya', 'Zimbabwe', 'El Salvador',
'Morocco', 'South Africa', 'Niger', 'Somalia', 'Nigeria',
'Ethiopia', 'Cameroon', 'Congo', 'Ghana', 'Ivory Coast',
'Equatorial Guinea', 'Gambia', 'Guatemala', 'Sierra Leone',
'Nepal'], dtype=object)
```

.nunique()

It will show the total no of unique value from whole data frame

```
In [11]: df.nunique()
```

```
Out[11]: rank                108
country                108
IQ                     40
education_expenditure    97
avg_income              106
avg_temp                91
dtype: int64
```

.describe()

It show the Count, mean , median etc

```
In [12]: df.describe()
```

```
Out[12]:
```

	rank	IQ	education_expenditure	avg_income	avg_temp
count	108.00000	108.000000	103.000000	106.000000	108.000000
mean	54.50000	85.972222	903.058252	17174.650943	23.858333
std	31.32092	12.998532	1166.625835	20871.092773	8.392232
min	1.00000	51.000000	1.000000	316.000000	0.400000
25%	27.75000	78.750000	81.500000	2263.250000	17.250000
50%	54.50000	88.000000	336.000000	7533.000000	25.850000
75%	81.25000	97.000000	1360.000000	30040.000000	31.275000
max	108.00000	106.000000	5436.000000	108349.000000	36.500000

.value_counts

It Shows all the unique values with their count

```
In [13]: df["country"].value_counts()
```

```
Out[13]: Hong Kong      1
Albania      1
Bolivia      1
Qatar        1
Oman          1
..
Spain        1
Russia       1
Czechia      1
Poland        1
Nepal         1
Name: country, Length: 108, dtype: int64
```

.isnull()

It shows the how many null values

```
In [14]: df.isnull()
```

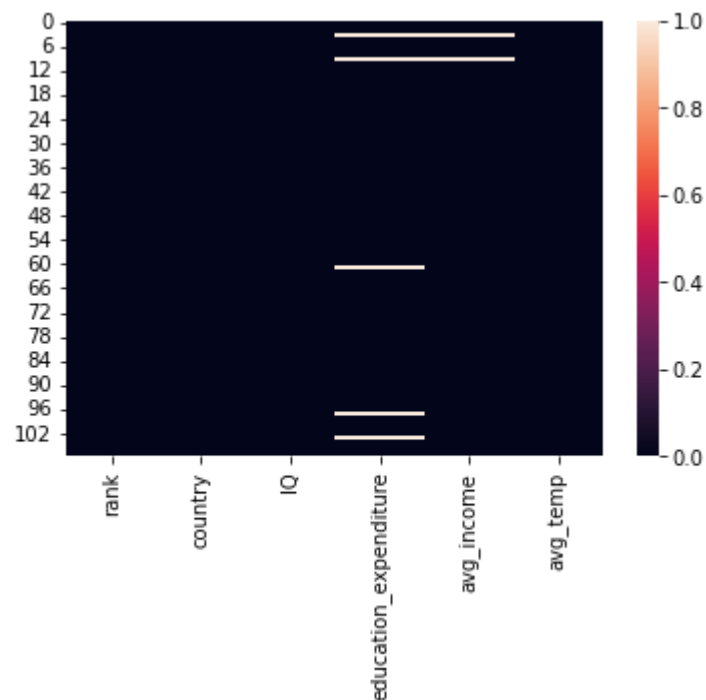
```
Out[14]:
```

	rank	country	IQ	education_expenditure	avg_income	avg_temp
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	True	True	False
4	False	False	False	False	False	False
...
103	False	False	False	True	False	False
104	False	False	False	False	False	False
105	False	False	False	False	False	False
106	False	False	False	False	False	False
107	False	False	False	False	False	False

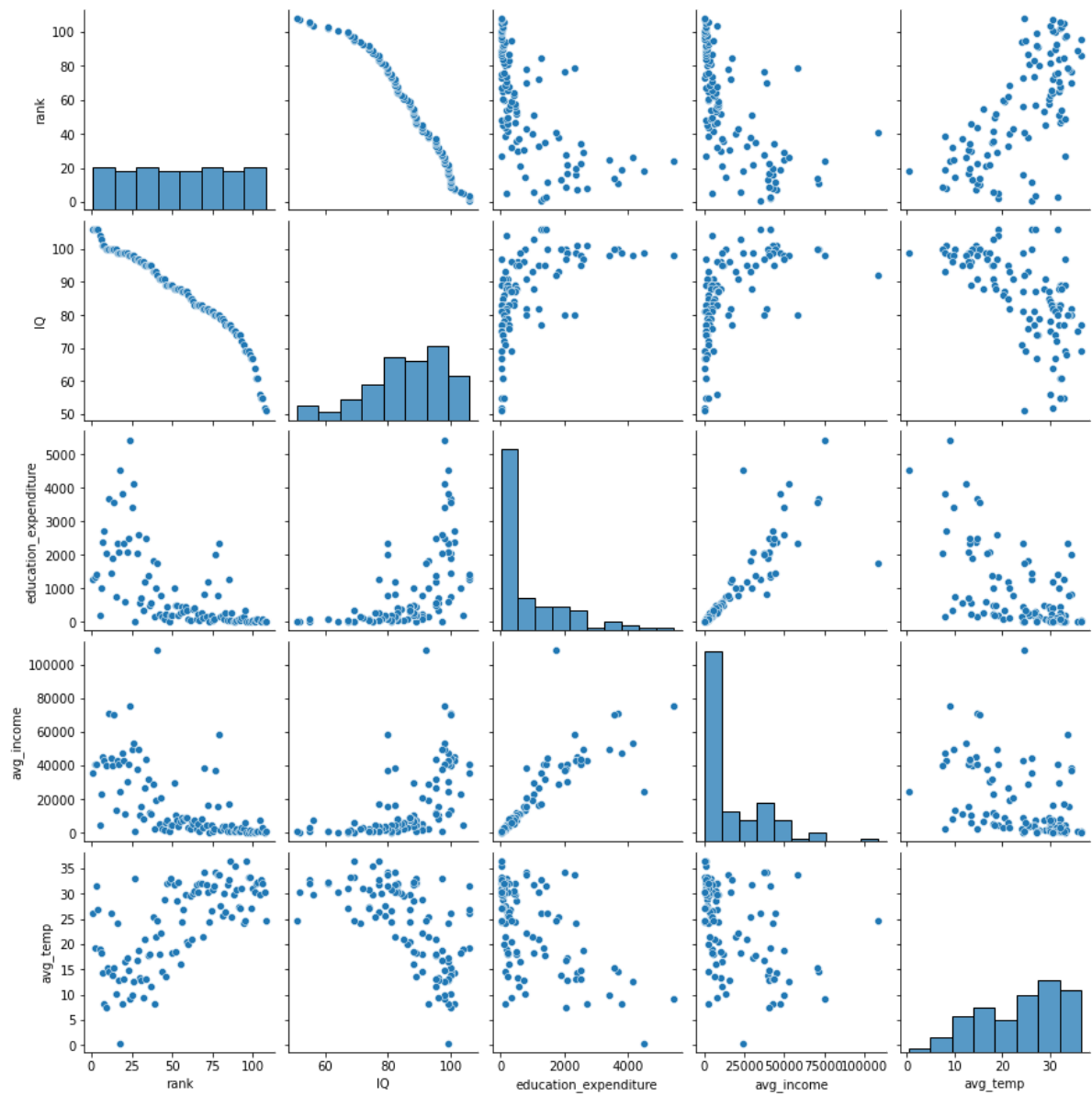
108 rows × 6 columns

```
In [15]: sns.heatmap(df.isnull())
```

```
Out[15]: <AxesSubplot:>
```

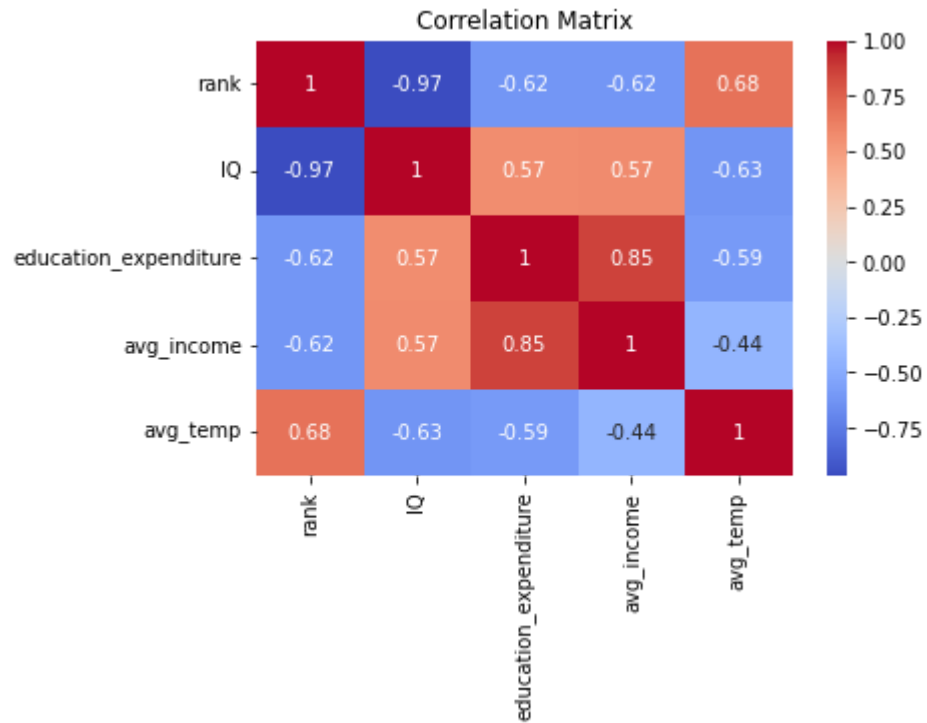


```
In [16]: # Visualizations
sns.pairplot(df)
plt.show()
```



```
In [18]: # Correlation matrix
correlation_matrix = df.corr()

# Visualize the correlation matrix
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
```

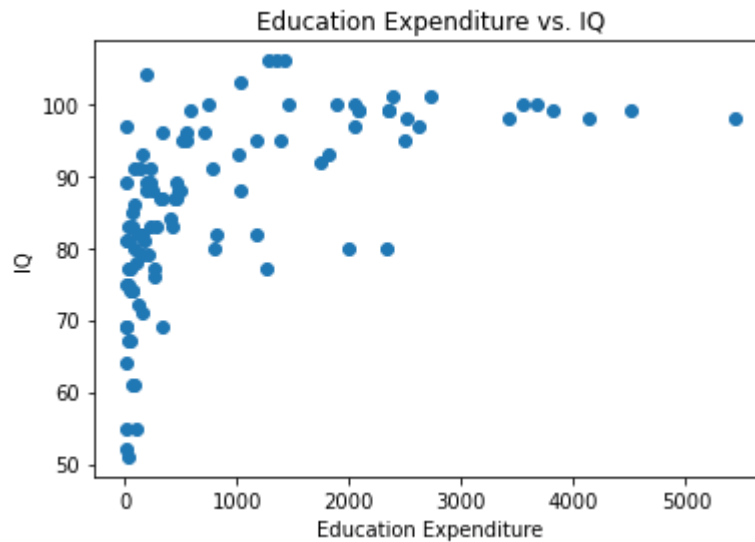


```
In [20]: df.corr()
```

Out[20]:

	rank	IQ	education_expenditure	avg_income	avg_temp
rank	1.000000	-0.967082	-0.616040	-0.616719	0.683438
IQ	-0.967082	1.000000	0.568237	0.569947	-0.628097
education_expenditure	-0.616040	0.568237	1.000000	0.854779	-0.591675
avg_income	-0.616719	0.569947	0.854779	1.000000	-0.439526
avg_temp	0.683438	-0.628097	-0.591675	-0.439526	1.000000

```
In [34]: # Scatter plot to analyze the impact of education expenses on IQ
plt.scatter(df['education_expenditure'], df['IQ'])
plt.xlabel('Education Expenditure')
plt.ylabel('IQ')
plt.title('Education Expenditure vs. IQ')
plt.show()
```

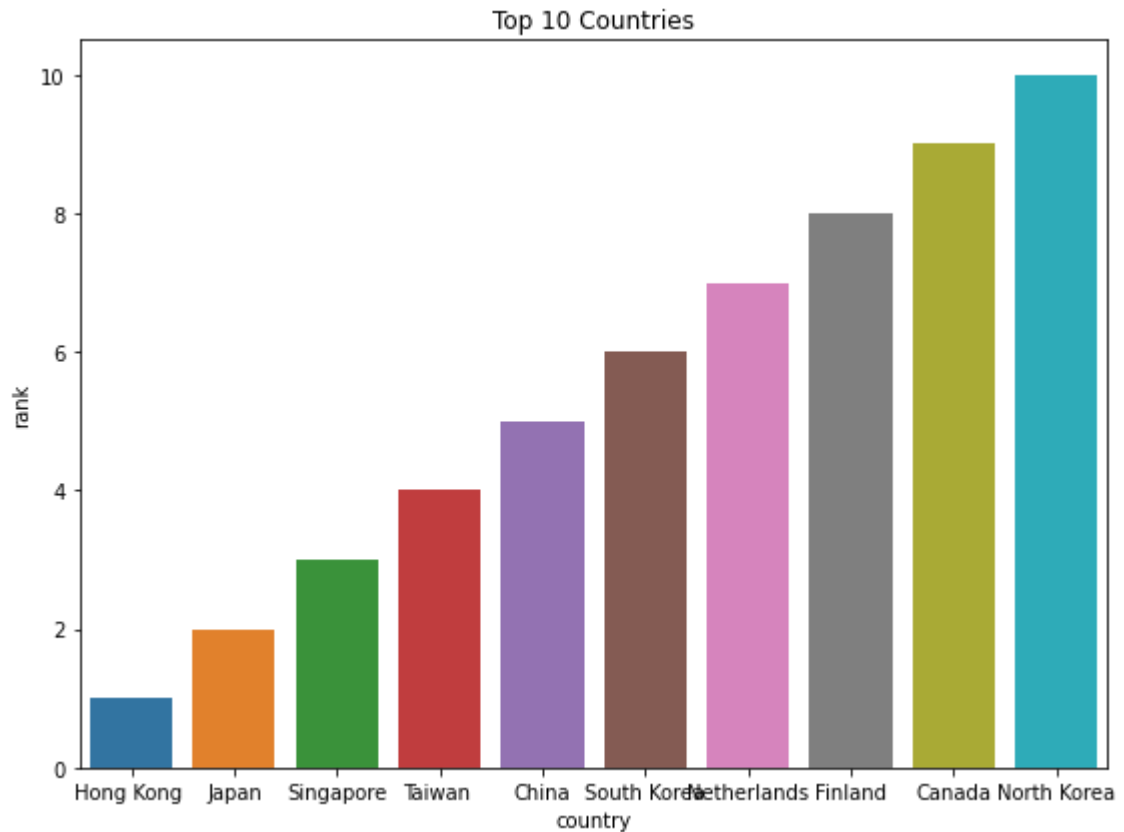


```
In [38]: # Select the top 10 countries
top_10_countries = df.sort_values(by='rank').head(10)

# Display the result
print(top_10_countries)
```

	rank	country	IQ	education_expenditure	avg_income	avg_temp
0	1	Hong Kong	106	1283.0	35304.0	26.2
1	2	Japan	106	1340.0	40964.0	19.2
2	3	Singapore	106	1428.0	41100.0	31.5
3	4	Taiwan	106	NaN	NaN	26.9
4	5	China	104	183.0	4654.0	19.1
5	6	South Korea	103	1024.0	22805.0	18.2
6	7	Netherlands	101	2386.0	45337.0	14.4
7	8	Finland	101	2725.0	42706.0	8.2
8	9	Canada	100	2052.0	40207.0	7.4
9	10	North Korea	100	NaN	NaN	15.3


```
In [40]: plt.figure(figsize=(8, 6))
sns.barplot(data=top_10_countries , x='country', y = 'rank')
plt.title('Top 10 Countries')
plt.xlabel('country')
plt.ylabel('rank')
plt.tight_layout()
plt.show()
```

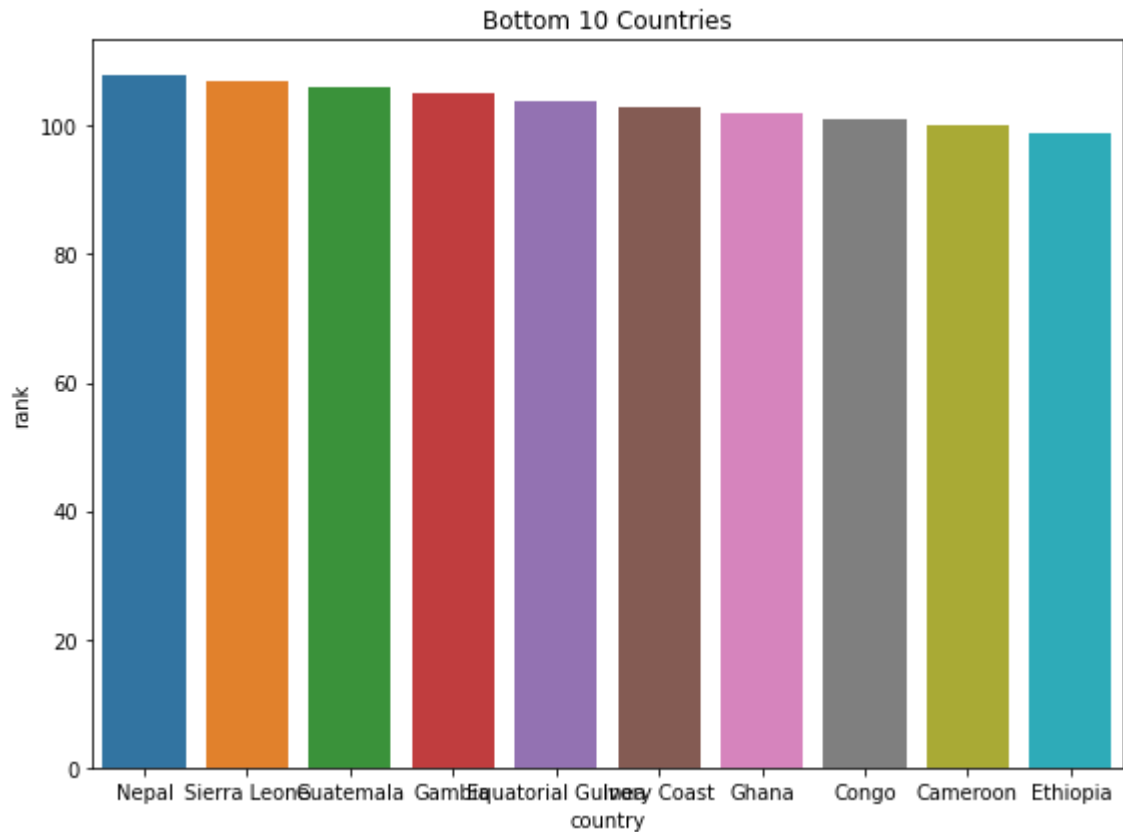


```
In [42]: # Select the top 10 countries
Bottom_10_countries = df.sort_values(by='rank',ascending = False).head(10)
Bottom_10_countries
```

Out[42]:

	rank	country	IQ	education_expenditure	avg_income	avg_temp
107	108	Nepal	51	22.0	595.0	24.6
106	107	Sierra Leone	52	16.0	412.0	30.4
105	106	Guatemala	55	92.0	2830.0	32.1
104	105	Gambia	55	14.0	648.0	32.9
103	104	Equatorial Guinea	56	NaN	7625.0	29.9
102	103	Ivory Coast	61	69.0	1289.0	32.2
101	102	Ghana	61	76.0	1166.0	32.1
100	101	Congo	64	7.0	316.0	30.4
99	100	Cameroon	67	36.0	1234.0	31.1
98	99	Ethiopia	67	21.0	379.0	27.2

```
In [49]: plt.figure(figsize=(8, 6))
sns.barplot(data=Bottom_10_countries , x='country', y ='rank')
plt.title('Bottom 10 Countries')
plt.xlabel('country')
plt.ylabel('rank')
plt.tight_layout()
plt.show()
```



```
In [44]: df[df['country'] == "Pakistan"]
```

Out[44]:

	rank	country	IQ	education_expenditure	avg_income	avg_temp
74	75	Pakistan	81	27.0	985.0	30.9

```
In [45]: # Scatter plot to analyze the impact of education expenses on IQ
plt.scatter(df['avg_income'], df['IQ'])
plt.xlabel('Average income')
plt.ylabel('IQ')
plt.title('Average income vs. IQ')
plt.show()
```

