CSEL 302

Final Project: Data Analysis and Visualization

Topic:

IQ Levels Analysis

Submitted by:

Lat, Betina Grace C.
Pino, Renalyn N.
BSCS - 2A

TABLE OF CONTENTS

01 **Project Overview**

Key user attributes: Rank, Country, IQ, Education Expenditure, Average Income, and Average Temperature.

02 **Libraries and Data Handling**

Libraries used: Pandas, Matplotlib, Seaborn, Plotly, Numpy, Scikit-Learn.

Data Loading and preprocessing: Loading from CSV, data cleaning and preprocessing, data normalization/standardization.

03 **Data Analysis Technique**

Descriptive statistics: Mean, median, count, standard deviation, variance, skewness, and kurtosis.

Inferential Statistics: A t-test was performed to compare IQ levels based on education expenditure.

Predictive Modeling: A linear regression model was built to predict IQ levels based on features such as education expenditure, average income, and average temperature.

04 **Key Findings**

IQ and Education Expenditure Analysis: Analysis of IQ distribution and education expenditure across different countries.

Correlation and Trends: Examining the relationships between key user attributes like IQ, education expenditure, average income, and average temperature.

Visual Analysis: : Summarizing key visual insights from various plots.

TABLE OF CONTENTS

05 **Advance Analysis**

Geographical Insights: Utilizing choropleth maps for comprehensive regional analysis

06 **Visual Insights**

IQ Range Distribution: A pie chart displaying the distribution of countries based on IQ ranges, highlighting the percentage of countries within each IQ range.

Correlation and Trends: Visualizing the relationships between key user attributes like IQ, education expenditure, average income, and average temperature.

Regional and Rank-Based Patterns: Insights into the leading countries based on various attributes like IQ, income, and education expenditure.

Visualization methods: Bar charts, pie charts, heatmaps, histogram, scatter plot, and choropleth map.

07 **Conclusion**

Summary of insights derived, implications for future strategic decisions.

Appendix

Code Snippets: Provided Python code used for loading, cleaning, transforming data, and generating visualizatins.

Datasets: Sample dataset of IQ Levels for data analysis.

Github Website Link: https://github.com/Betinsss/IQ-levels-analysis

I. **Project Overview**

The purpose of this project is to analyze the key user attributes such as Rank, Country, IQ, Education Expenditure, Average Income, and Average Temperature. We want to know if there's a connection between IQ and the money spent on education or the average income of people. Likewise, we'll also look for patterns in different regions and see if temperature affects IQ. In short, through examining these data points, we aim to uncover insights into the socio-economic and environmental conditions of different countries.

The dataset includes several crucial attributes that provide valuable information about each country. Here is a breakdown of how each attribute will be analyzed to derive meaningful insights:

Country: Identifying the country helps in comparing and contrasting different nations' socio-economic attributes and environmental conditions.

Rank: Knowing where countries stand in terms of key attributes helps us compare their socio-economic status and environmental conditions. It's like placing them on a ladder based on things like education, income, and climate, so we can see which countries are doing well in different areas and where improvements might be needed.

IQ: Analyzing IQ scores can provide insights into the cognitive abilities of populations, which can correlate with education outcomes and economic performance.

Education Expenditure: Examining how much each country spends on education can reveal the importance placed on education by different governments and its impact on national development.

Average Income: This attributes is crucial for understanding the economic well-being of the population in each country. Higher average incomes are typically associated with better living standards and economic stability.

Average Temperature: Knowing the average temperature of a country can provide context for environmental and climatic conditions, which can influence lifestyle, agricultural practices, and economic activities of an individual.

II. Libraries and Data Handling

Libraries Used: Pandas for data manipulation, Matplotlib, Seaborn, and Plotly for data visualization, NumPy for numerical operations, and Scikit-learn for machine learning.

1. **Pandas**: A library for data manipulation and analysis, providing data structures like DataFrames to handle and process large datasets efficiently.

2. **Matplotlib**: A plotting library that creates static, interactive, and animated visualizations in Python.

3. **Seaborn**: A statistical data visualization library based on Matplotlib that makes it easier to create attractive and informative statistical graphics.

4. **Plotly**: An interactive graphing library that enables the creation of dynamic and interactive visualizations in Python.

5. **NumPy**: A library for numerical operations in Python, providing support for arrays and matrices, along with a collection of mathematical functions to operate on these data structures.

6. **Scikit-learn**: A machine learning library that provides simple and efficient tools for data analysis and modeling, including classification, regression, clustering, and dimensionality reduction.

Data Loading: Data is loaded from a CSV file into a DataFrame.

Loading Data from CSV: The dataset is loaded into a Pandas DataFrame from a CSV file, a common practice for data analysis. Using pd.read_csv(), this method converts the structured data into a DataFrame, enabling powerful data manipulation capabilities within Python.

Data Cleaning and Preprocessing: Basic preprocessing steps are performed, such as handling missing values, handling categorial data, and data normalization/standardization is performed.

- **Handling Missing Values**: Missing values in numerical columns are imputed using the SimpleImputer from Scikit-learn. The imputer replaces missing values with the mean, median, most frequent value, or a constant.

- **Handling Categorical Data**: Categorical columns are identified and transformed using encoding techniques like Label Encoding or One-Hot Encoding. This step is crucial for making categorical data usable in machine learning algorithms.

Data Normalization/Standardization:

- **Standardization**: Using the StandardScaler from Scikit-learn, data is standardized so that it has a mean of 0 and a variance of 1.

- **Normalization**: Using the MinMaxScaler from Scikit-learn, data is normalized to a range between 0 and 1.

These steps ensure that the dataset is ready for further analysis and visualization, providing a structured approach to understanding and working with the data.

III. Data Analysis Techniques

Descriptive Statistics: Summary statistics such as mean, median, count, standard deviation, variance, skewness, and kurtosis were used to understand the distribution of the data. These metrics provide a quick overview of the dataset:

- **Mean and Median:** These measures provide insights into the central tendency of numerical data, such as IQ and education expenditure. For instance, the mean IQ can indicate the overall cognitive performance in the dataset, while the median IQ helps to understand the central point of the IQ distribution, aiding in demographic analysis.

- **Count:** The count gives the total number of non-null entries in each column, useful for understanding the size of the data and identifying columns with missing values.

- **Standard Deviation:** This statistic measures the amount of variation or dispersion in the data. A high standard deviation in IQ might indicate significant differences in cognitive abilities across the sample.

- **Variance, Skewness, and Kurtosis:** These metrics provide further insights into the data's variability and distribution shape. Variance indicates the degree of spread in the data, while skewness and kurtosis help identify the asymmetry and peakedness of the data distribution, respectively.

Inferential Statistics: A t-test was performed to compare IQ levels based on education expenditure. By dividing the data into high and low education expenditure groups, the t-test helps infer if there is a statistically significant difference in IQ levels between these groups.

Predictive Modeling: A linear regression model was built to predict IQ levels based on features such as education expenditure, average income, and average temperature. The model's performance was evaluated using metrics like Mean Squared Error (MSE) and R^2 score to assess its predictive accuracy. This predictive modeling approach helps in understanding the impact of various factors on IQ and forecasting future trends based on the identified relationships.

IV. Key Findings

IQ and Education Expenditure Analysis: Analysis of IQ distribution and education expenditure across different countries.

- **IQ Range Distribution:** Understanding the distribution of countries by IQ ranges helps identify the general intelligence levels in various regions. For instance, visualizing the percentage of countries within specific IQ ranges can highlight areas with higher or lower average IQ, guiding educational and developmental initiatives.

- **Education Expenditure:** Analyzing how much different countries spend on education in relation to IQ levels can reveal correlations between investment in education and intelligence outcomes. This insight can inform policy decisions and funding allocations to improve educational systems.

Correlation and Trends: Examining the relationships between key user attributes like IQ, education expenditure, average income, and average temperature.

- **Correlation Between Variables:** Heatmaps displaying correlations between variables such as IQ, education expenditure, and average income, and average temperature help identify significant relationships and patterns. For instance, a strong correlation between education expenditure and IQ suggests the impact of educational investment on intelligence.

Regional and Rank-Based Patterns: Exploration of patterns based on country rank and regional attributes.

- **Country Rank Analysis:** Bar charts showing patterns based on country rank can highlight the leading countries in terms of IQ, income, and education expenditure. This aids in benchmarking and understanding regional disparities and strengths.

- **Regional Distribution:** Choropleth maps visualizing attributes like IQ, average income, and education expenditure by country provide a geographical perspective, making it easier to identify regional trends and focus areas for improvement.

Visual Analysis: Summarizing key visual insights from various plots.

- **Pie Charts:** Useful for showing the proportional distribution of attributes like IQ ranges among countries, making it easy to see which ranges are most common.

- **Histograms:** These charts help in understanding the distribution of attributes like average income and education expenditure, identifying common values, and spotting outliers.

- **Scatter Plot:** Scatter plots, such as those showing the relationship between education expenditure and IQ, provide a clear visual representation of how changes in one variable may influence another.

Impact of the Findings in Education and Economic Sectors: These findings can help experts in the education and economic sectors make informed decisions by identifying countries with lower IQ levels and correlating them with education expenditure, thus guiding targeted investments. Understanding the strong relationships between IQ, average income, education expenditure, and average temperature can influence policy adjustments to prioritize educational funding where it's most needed. Additionally, visual insights into regional patterns can help strategists design localized initiatives to improve overall intelligence and economic outcomes.

These findings are invaluable as they provide a comprehensive view of user attributes and their interrelationships, offering predictive insights that can guide strategic decisions and improve educational and developmental outcomes across different regions.

V. Advanced Analysis

- **Geographical Insights:** Utilizing choropleth maps for comprehensive regional analysis

- **Choropleth Maps by Country:** By employing choropleth maps to visualize variables like rank, IQ, education expenditure, average income, and average temperature, the analysis gains a geographical dimension that is essential for strategic planning. This approach allows for a clear visual representation of how these key metrics vary across different countries, highlighting regional disparities and trends that are vital for informed decision-making.

- **Visualizing Country-Specific Data:** Choropleth maps provide a detailed view of country-specific data, making it easier to identify which countries lead in certain metrics and which ones lag behind. This visualization helps in pinpointing areas that require targeted interventions or further research. For instance, countries with lower education expenditure but high average IQ might indicate efficient educational systems or other contributing factors that need exploration.

- **Strategic Regional Analysis:** With the country-based visualization, stakeholders can perform a more nuanced regional analysis, comparing performance across different countries and continents. This is crucial for understanding localized challenges and opportunities, allowing for the tailoring of policies and strategies to fit the unique needs of each region. For example, countries with high average income but lower ranks in IQ might benefit from educational reforms or investments in cognitive development programs.

These analyses help experts understand broader information by visualizing how key metrics like IQ, education expenditure, average temperature, and average income vary across countries. This geographical insight reveals regional disparities and patterns, guiding targeted strategies and interventions to address specific regional needs effectively.

VI. Visual Insights

Data Visualization

Various plots such as bar charts, pie charts, and heatmaps are used to visualize patterns and trends in the dataset. Visual representations of data are employed to understand trends, patterns, and outliers more intuitively, specifically regarding key user attributes:

- **Bar Charts:** These are used to compare the frequency or count of categories across different groups. For example, a bar chart could compare the number of countries in each IQ range or illustrate the distribution of countries by rank.

- **Pie Charts:** These charts are excellent for showing the proportional distribution of categories. They could be used to display the percentage share of each IQ range among countries, making it easy to see which range is most common.

- **Heatmaps: These** can be effective for visualizing the intensity of data, making them ideal for spotting correlations and patterns across multiple variables. In the context of this dataset, a heatmap could visualize the correlation between IQ and education expenditure, highlighting any trends or relationships.

- **Histograms and Scatter Plots:** These are helpful for visualizing the distribution of numerical data. A histogram of IQ scores or a scatter plot of IQ vs. education expenditure could provide insights into the data distribution and relationships between variables.

    - **Choropleth Maps:** These maps are used to visualize data distribution across geographic regions. For example, a choropleth map could show the average IQ scores by country, providing a geographical perspective on IQ distribution.

These visualization techniques are essential for gaining insights into the dataset and making informed decisions based on data patterns and trends. Descriptive statistics provide the numerical background necessary to understand the data, while visualization techniques bring this data to life, making it easier for stakeholders to digest and make strategic decisions.

Analysis:

- **IQ Range Distribution:** A pie chart displaying the distribution of countries based on IQ ranges, highlighting the percentage of countries within each IQ range.

- **Correlation and Trends:** Visualizing the relationships between key user attributes like IQ, education expenditure, average income, and average temperature.

- **Regional and Rank-Based Patterns:** Insights into the leading countries based on various attributes like IQ, income, and education expenditure.

IQ Range Distribution:

- **Pie Chart:** This chart is used to show the distribution of countries by IQ range. By visualizing this distribution, stakeholders can quickly understand the proportion of countries falling within specific IQ ranges (70-85, 85-100, 100-115, 115-130). For instance, if a significant number of countries are in the 85-100 IQ range, educational policies and resources can be tailored to this demographic to improve intellectual development.

- **Implications:** Understanding the IQ range distribution helps in designing and implementing educational and developmental programs that are better suited to the intellectual capabilities of different countries. This can lead to more effective policies and strategies, potentially enhancing overall educational outcomes and intellectual growth across various regions.

Correlation and Trends:

- **Heatmap Analysis:** Heatmaps displaying correlations between variables such as IQ, education expenditure, average income, and average temperature help identify significant relationships and patterns. For instance, a strong correlation between education expenditure and IQ suggests the impact of educational investment on intelligence.

- **Strategic Insights:** Understanding the correlations between these variables allows stakeholders to make informed decisions regarding resource allocation and policy-making. For example, if a high correlation is observed between education expenditure and IQ, efforts to increase educational funding in regions with lower IQ scores could be prioritized to foster intellectual development and economic growth.

Regional and Rank-Based Patterns:

- **Visualization of Country Ranks:** Bar charts showing patterns based on country rank can highlight the leading countries in terms of IQ, income, and education expenditure. This aids in benchmarking and understanding regional disparities and strengths. For example, countries with higher ranks in education expenditure might show better IQ scores, emphasizing the impact of educational investments.

- **Strategic Decisions:** These insights help stakeholders identify areas that require improvement and allocate resources efficiently. Understanding which countries excel in certain attributes allows for targeted initiatives to reduce disparities and enhance overall regional development. For instance, regions with lower income but higher IQ scores might benefit from increased economic investment to maximize their intellectual potential.

VII. Conclusion

This project has provided a comprehensive analysis of key user attributes such as Country, IQ, Education Expenditure, Average Income, and Average Temperature. Thus, through utilizing advanced data handling and visualization techniques, we have uncovered critical insights into the socio-economic and environmental conditions across different countries. The use of libraries like Pandas, Matplotlib, Seaborn, Plotly, NumPy, and Scikit-learn has enabled the transformation of raw data into meaningful intelligence that highlights significant correlations and regional patterns.

These insights are invaluable for experts sectors, guiding data-driven decision-making. Hence, by identifying connections between IQ and factors like education expenditure and average income, as well as understanding the impact of environmental conditions, stakeholders can formulate targeted strategies to improve educational outcomes and economic stability. Moreover, visual tools such as choropleth maps offer a clear geographical perspective, facilitating strategic regional analysis and the implementation of localized initiatives.

In line with this, the importance of data-driven decision-making cannot be overstated. These analyses not only reveal current trends and disparities but also provide a predictive lens through which future strategies can be shaped. This approach ensures that policies and interventions are tailored to address specific regional needs, ultimately driving socio-economic development and enhancing the overall well-being of populations. Likewise, the potential for future analysis remains vast, with continuous data collection and advanced modeling offering even deeper insights into the complex interrelationships that shape our world.

Appendix

Code Snippets: Provided Python code used for loading, cleaning, transforming data, and generating visualizations.

Datasets: Sample dataset of IQ Levels for data analysis.

Github Website Link: https://github.com/Betinsss/IQ-levels-analysis

Import Libraries

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import plotly.express as px
import os
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.impute import SimpleImputer
import plotly.graph_objects as go
import plotly.offline as pyo
import plotly.io as pio
```

Uploading Csv file

```
data = pd.read_csv('/content/08_IQ Levels Analysis.csv')
```

Data Preprocessing

.head()

```
data.head()
```

	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()

```
data.tail()
```

	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

```
data.shape
```

```
(108, 6)
```

.columns

```
data.columns
```

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

.dtypes

```
data.dtypes
```

```
rank                     int64
country                 object
IQ                       int64
education_expenditure    float64
avg_income               float64
avg_temp                 float64
dtype: object
```

.unique()

```
data["country"].unique()
```

```
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()

```
data.nunique()
```

```
rank                    108
country                 108
IQ                       40
education_expenditure    97
avg_income              106
avg_temp                 91
dtype: int64
```

.describe()

```
data.describe()
```

	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

```
data["country"].value_counts()
```

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

Handling Missing Values

.isnull()

```
data.isnull()
```

	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

Handling Categorical Data

```
# Identify categorical columns
categorical_cols = data.select_dtypes(include=['object']).columns
print(categorical_cols)

# Label Encoding for ordinal categorical variables
label_encoder = LabelEncoder()
for col in categorical_cols:
    data[col] = label_encoder.fit_transform(data[col])

# One-Hot Encoding for nominal categorical variables
data_encoded = pd.get_dummies(data, columns=categorical_cols)
```

```
Index(['country'], dtype='object')
```

Data Normalization/Standardization (if needed)

```
from sklearn.preprocessing import StandardScaler, MinMaxScaler

# Standardization (mean=0, variance=1)
scaler = StandardScaler()
data_standardized = pd.DataFrame(scaler.fit_transform(data), columns=data.columns)

# Normalization (range 0-1)
scaler = MinMaxScaler()
data_normalized = pd.DataFrame(scaler.fit_transform(data), columns=data.columns)
```

Impute missing values for numerical columns

```
# Handle missing values
imputer = SimpleImputer(strategy='mean')  # You can use 'median', 'most_frequent', or 'constant' strategies
data_imputed = pd.DataFrame(imputer.fit_transform(data.select_dtypes(include=[np.number])), columns=data.select_dtypes(include=[np.numbe

# If there are non-numeric columns, we need to concatenate them back after imputation
non_numeric_data = data.select_dtypes(exclude=[np.number])
data_imputed = pd.concat([data_imputed, non_numeric_data], axis=1)
```

Descriptive Statistics

```
# Display basic descriptive statistics
print(data_imputed.describe())

# Additional descriptive statistics
print("Mean:\n", data_imputed.mean())
print("Median:\n", data_imputed.median())
print("Standard Deviation:\n", data_imputed.std())
print("Variance:\n", data_imputed.var())
print("Skewness:\n", data_imputed.skew())
print("Kurtosis:\n", data_imputed.kurt())
```

```
             rank     country          IQ  education_expenditure     avg_income  \
count  108.000000  108.000000  108.000000             108.000000     108.000000
mean    54.500000   53.500000   85.972222             903.058252   17174.650943
std     31.320921   31.320921   12.998532            1139.042128   20675.115731
min      1.000000    0.000000   51.000000               1.000000     316.000000
25%     27.750000   26.750000   78.750000              90.000000    2307.750000
50%     54.500000   53.500000   88.000000             411.000000    7605.500000
75%     81.250000   80.250000   97.000000            1297.250000   29838.000000
max    108.000000  107.000000  106.000000            5436.000000  108349.000000

         avg_temp
count  108.000000
mean    23.858333
std      8.392232
min      0.400000
25%     17.250000
50%     25.850000
75%     31.275000
max     36.500000
Mean:
 rank                    54.500000
country                 53.500000
IQ                      85.972222
```

```
education_expenditure      903.058252
avg_income               17174.650943
avg_temp                    23.858333
dtype: float64
Median:
 rank                       54.50
country                     53.50
IQ                          88.00
education_expenditure      411.00
avg_income                7605.50
avg_temp                    25.85
dtype: float64
Standard Deviation:
 rank                       31.320920
country                     31.320920
IQ                          12.998532
education_expenditure     1139.042128
avg_income               20675.115731
avg_temp                     8.392232
dtype: float64
Variance:
 rank                      9.810000e+02
country                   9.810000e+02
IQ                        1.689618e+02
education_expenditure     1.297417e+06
avg_income                4.274684e+08
avg_temp                  7.042956e+01
dtype: float64
Skewness:
 rank                       0.000000
country                    0.000000
IQ                        -0.696320
education_expenditure      1.730124
avg_income                 1.645693
```

Inferential Statistics

Perform a t-test to compare IQ levels based on education expenditure.

```
from scipy.stats import ttest_ind

# Divide the data based on median education expenditure
median_education_expenditure = data_imputed['education_expenditure'].median()
high_education_expenditure = data_imputed[data_imputed['education_expenditure'] > median_education_expenditure]['IQ']
low_education_expenditure = data_imputed[data_imputed['education_expenditure'] <= median_education_expenditure]['IQ']

# Perform t-test
t_stat, p_value = ttest_ind(high_education_expenditure, low_education_expenditure)
print(f'T-statistic: {t_stat}, P-value: {p_value}')
```

```
T-statistic: 7.229563744144697, P-value: 7.8454869076436696-11
```

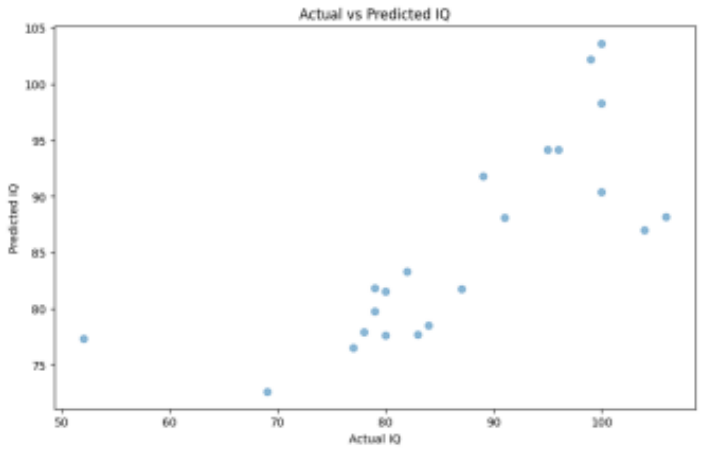
Predictive Model

Build a linear regression model to predict IQ based on education expenditure, average income, and average temperature.

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Define the feature set and target variable
X = data_imputed[['education_expenditure', 'avg_income', 'avg_temp']]
y = data_imputed['IQ']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize and train the model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f'Mean Squared Error: {mse}')
print(f'R^2 Score: {r2}')
```

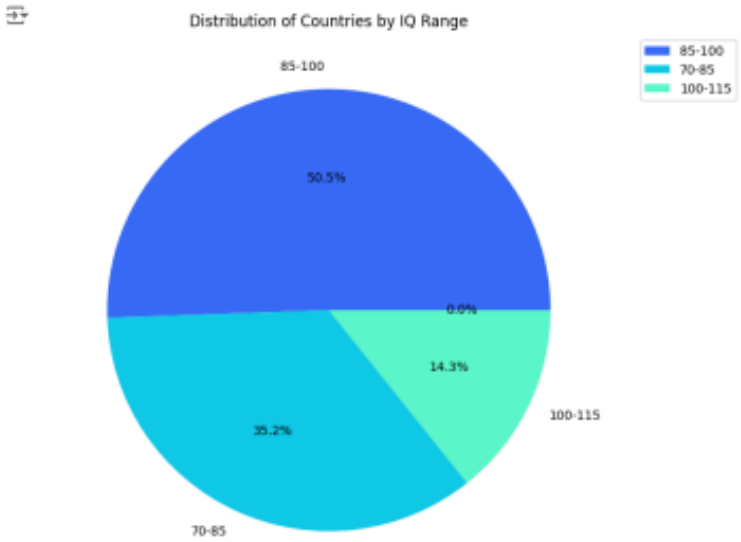
```
Mean Squared Error: 68.31731478151806
R^2 Score: 0.5670230940412904
```

```
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, alpha=0.5)
plt.xlabel('Actual IQ')
plt.ylabel('Predicted IQ')
plt.title('Actual vs Predicted IQ')
plt.show()
```

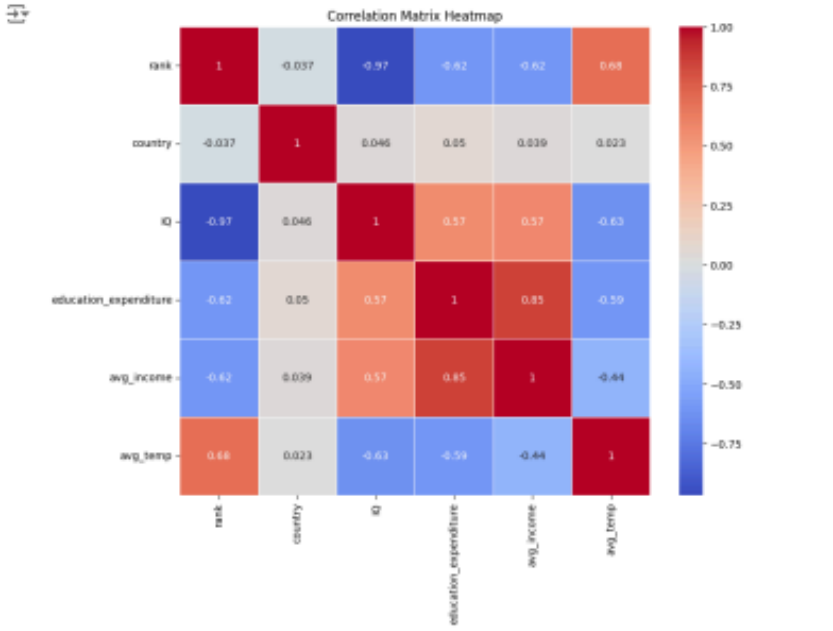


Visualization

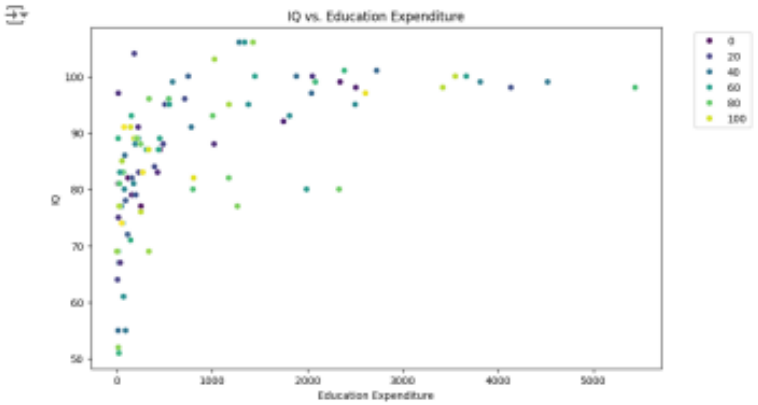
```
# Create IQ ranges
bins = [70, 85, 100, 115, 130]
labels = ['70-85', '85-100', '100-115', '115-130']

data['IQ_range'] = pd.cut(data['IQ'], bins=bins, labels=labels, right=False)

# Pie Chart: Distribution of Countries by IQ Range
iq_range_distribution = data['IQ_range'].value_counts()
plt.figure(figsize=(8, 8))
iq_range_distribution.plot(kind='pie', autopct='%1.1f%%', colors=sns.color_palette('rainbow'))
plt.title('Distribution of Countries by IQ Range')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
plt.ylabel('')
plt.show()
```



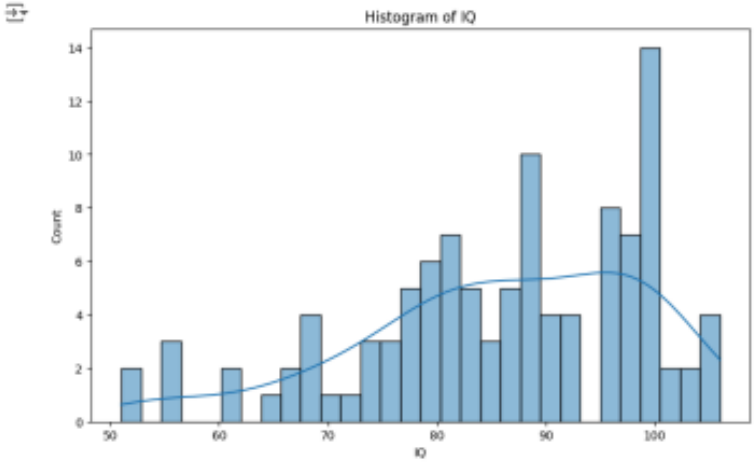

```
# Heatmap: Correlation Between Variables
plt.figure(figsize=(10, 8))
numeric_data = data.select_dtypes(include=[np.number])  # Select only numeric columns for the heatmap
corr_matrix = numeric_data.corr()
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Matrix Heatmap')
plt.show()
```



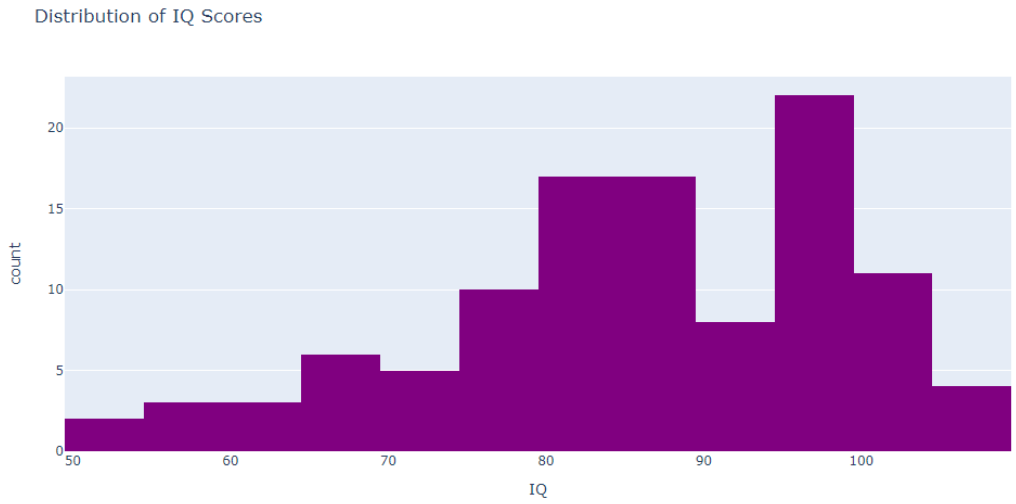

```
# Scatter Plot: IQ vs. Education Expenditure
plt.figure(figsize=(10, 6))
sns.scatterplot(x='education_expenditure', y='IQ', data=data, hue='country', palette='viridis')
plt.title('IQ vs. Education Expenditure')
plt.xlabel('Education Expenditure')
plt.ylabel('IQ')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



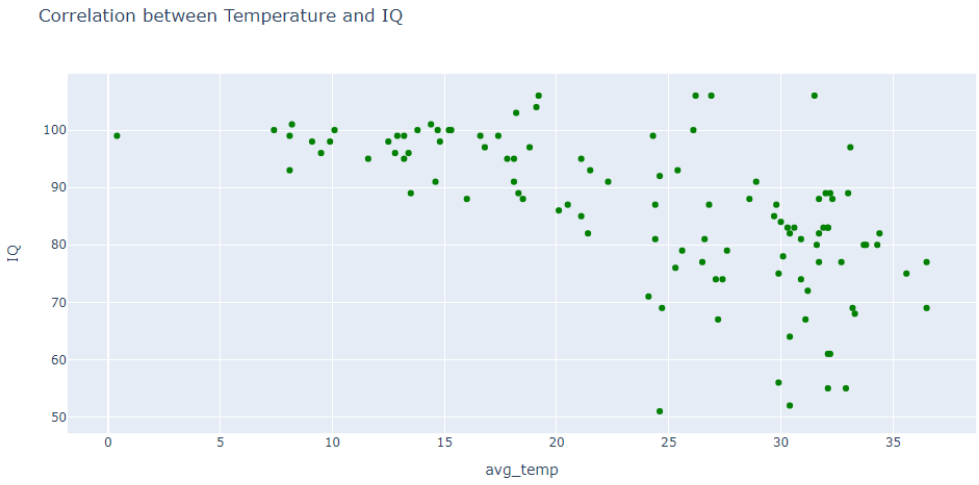

```
# Histogram
plt.figure(figsize=(10, 6))
sns.histplot(data['IQ'], bins=30, kde=True)
plt.title('Histogram of IQ')
plt.show()
```



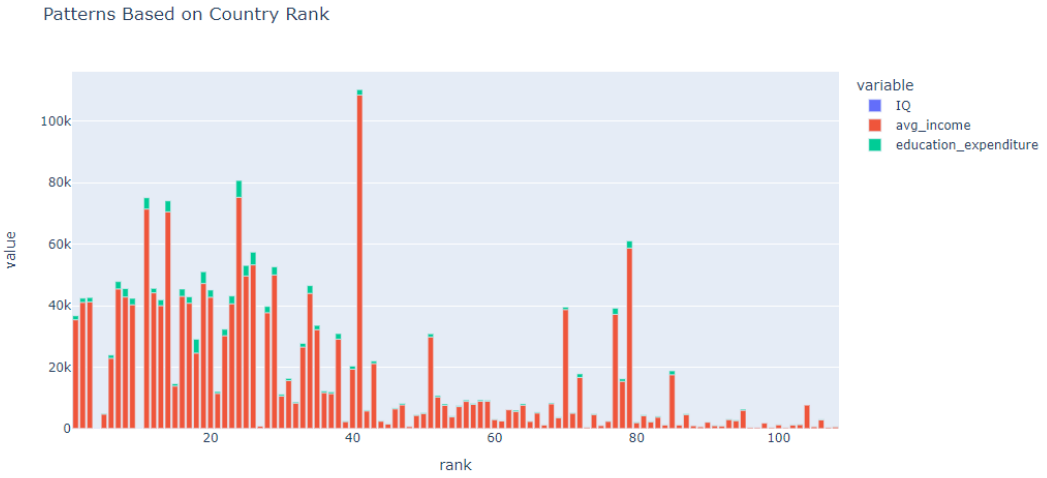

```
fig = px.histogram(data, x="IQ", title="Distribution of IQ Scores")
fig.update_traces(marker=dict(color='purple'))
fig.show()
```



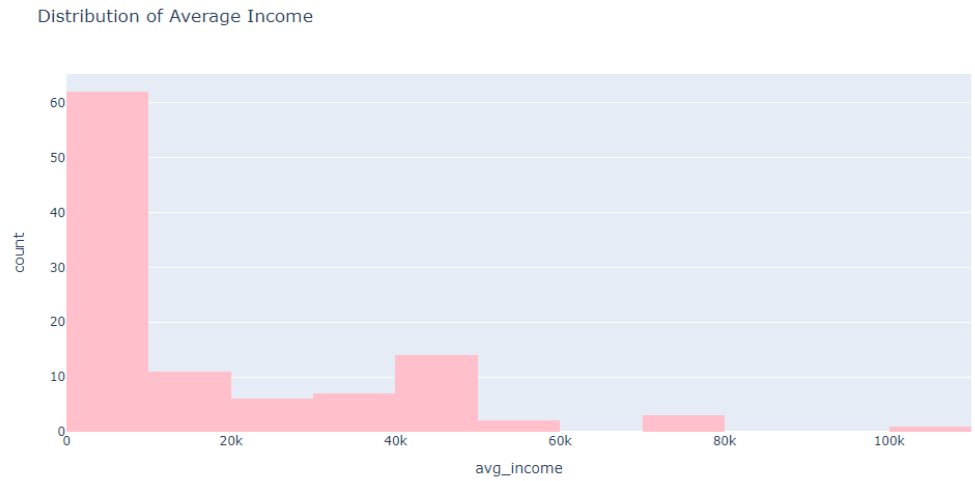

```
[ ]  fig = px.scatter(data, x="avg_temp", y="IQ", title="Correlation between Temperature and IQ")
     fig.update_traces(marker=dict(color='green'))
     fig.show()
```



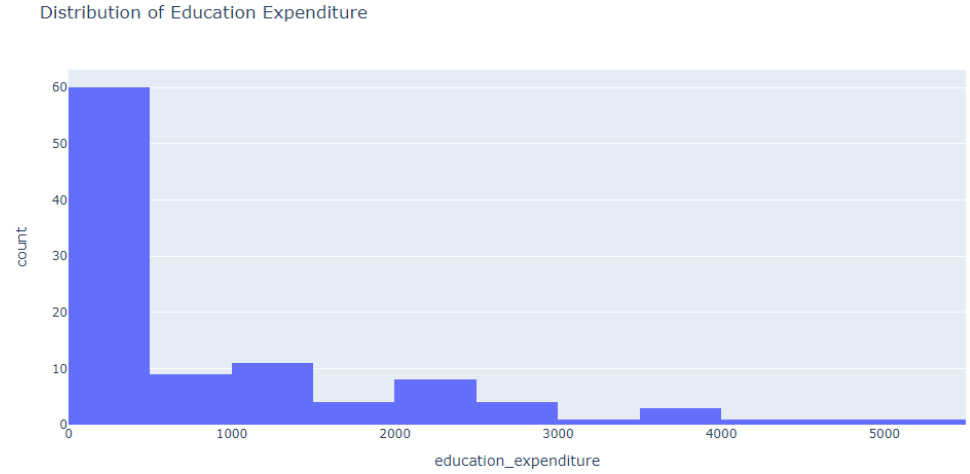

```
[ ]  fig = px.bar(data, x="rank", y=["IQ", "avg_income", "education_expenditure"], title="Patterns Based on Country Rank")
     fig.show()
```



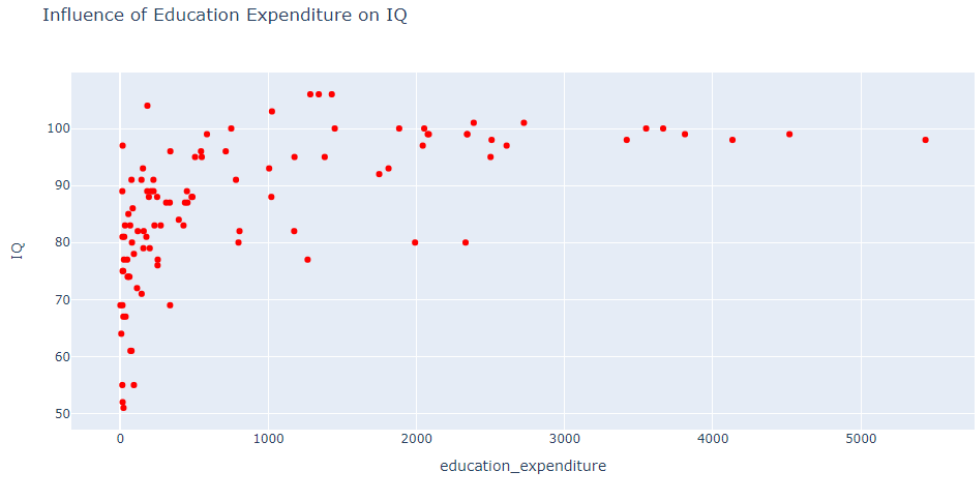
```
[ ]  fig = px.histogram(data, x="avg_income", title="Distribution of Average Income")
     fig.update_traces(marker=dict(color='pink'))
     fig.show()
```



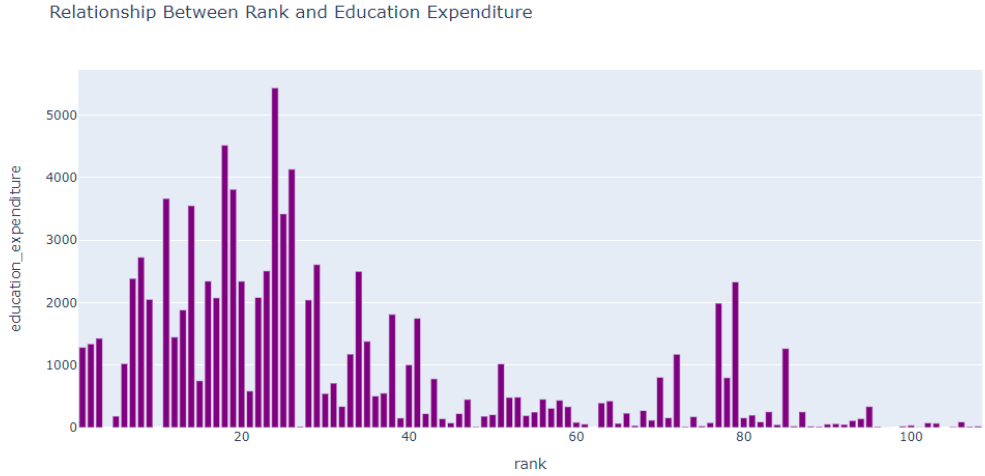

```
     fig = px.histogram(data, x="education_expenditure", title="Distribution of Education Expenditure")
     fig.show()
```



```
[ ]  fig = px.scatter(data, x="education_expenditure", y="IQ", title="Influence of Education Expenditure on IQ")
     fig.update_traces(marker=dict(color='red'))
     fig.show()
```

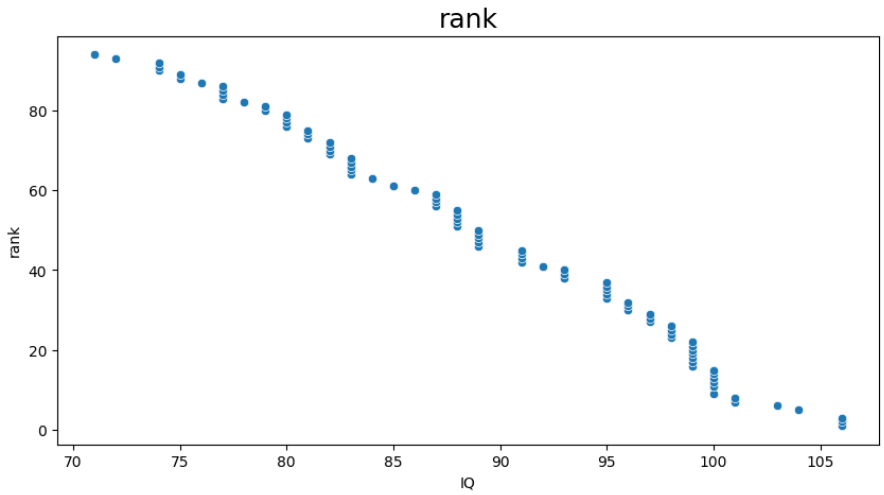


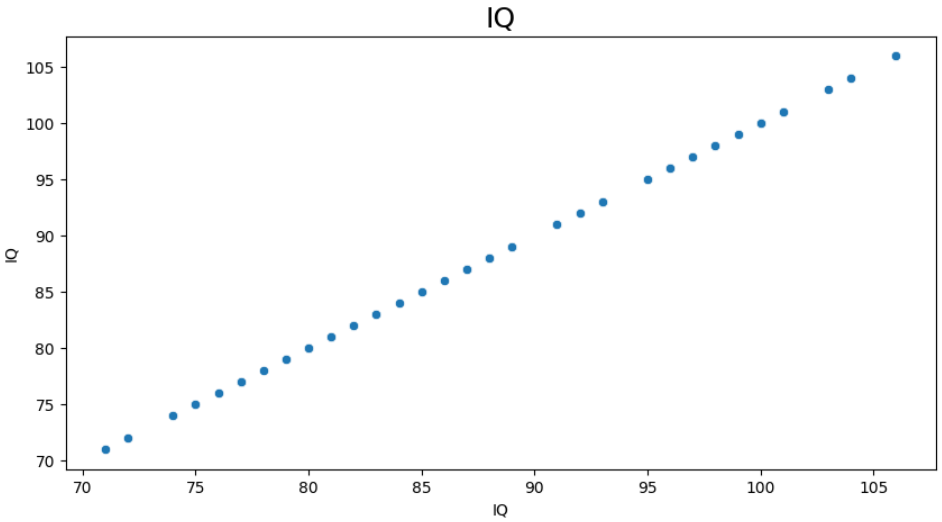

```
fig = px.bar(data, x="rank", y="education_expenditure", title="Relationship Between Rank and Education Expenditure")
fig.update_traces(marker=dict(color='purple'))
fig.show()
```

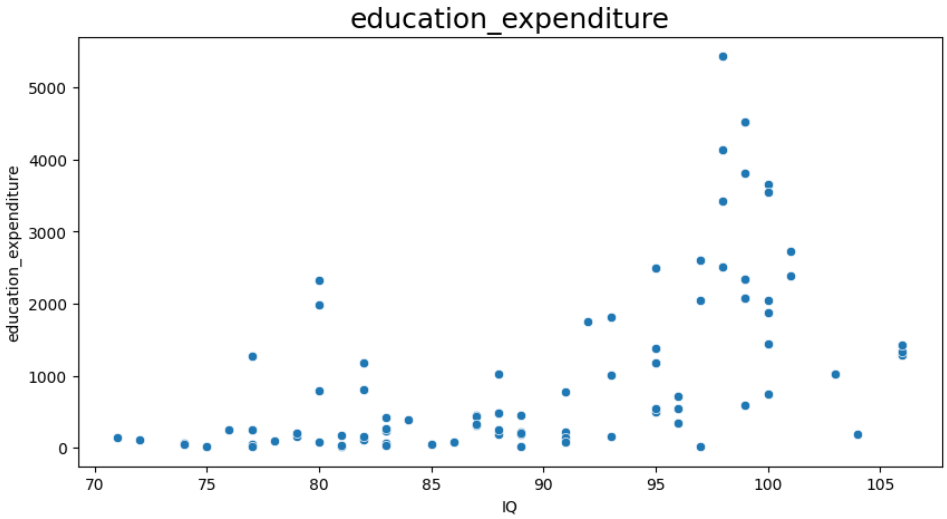


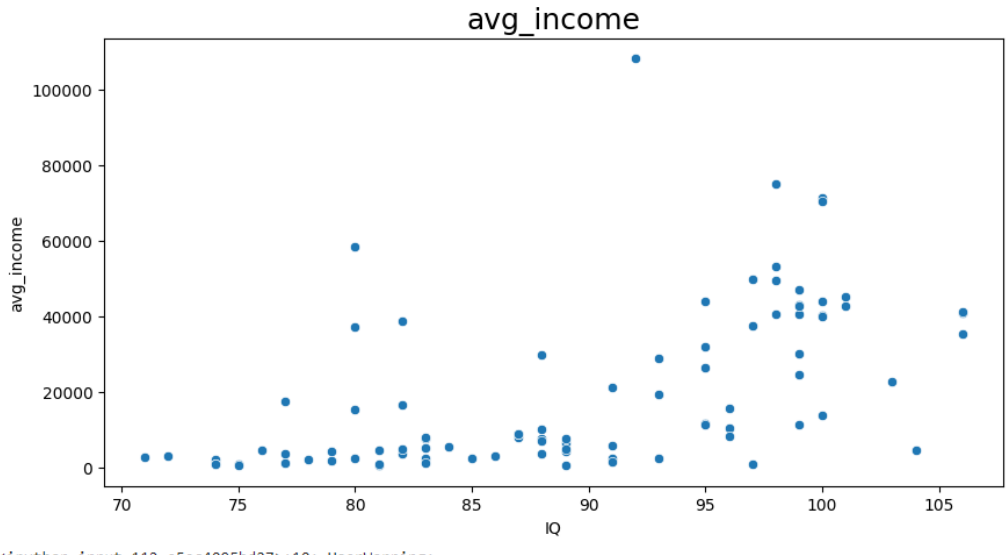
```
for i in cols:

    if i == 'IQ':
        fig, ax = plt.subplots(figsize=(10, 5))
        sns.scatterplot(data=data, x="IQ", y="IQ", palette="rainbow")
        plt.title("IQ", fontsize = 18)
        plt.show()
    else:
        fig, ax = plt.subplots(figsize=(10, 5))
        sns.scatterplot(data=data, x="IQ", y= i, palette="Set2")
        plt.title(i, fontsize = 18)
        plt.show()
```

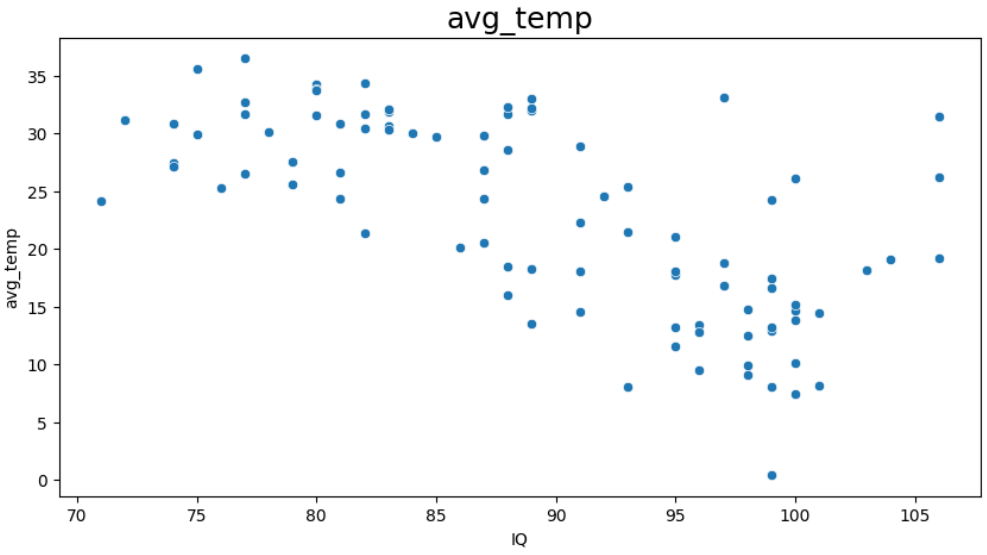
```
<ipython-input-112-e5ec4095bd27>:10: UserWarning:

Ignoring `palette` because no `hue` variable has been assigned.
```





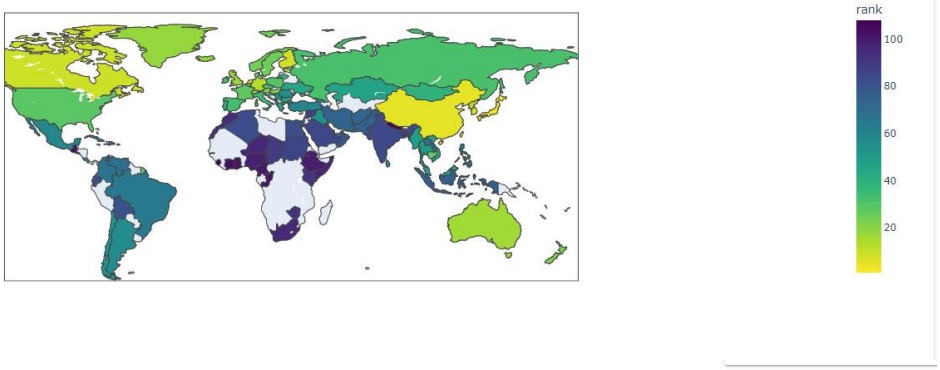

education_expenditure
education_expenditure
IQ
0
1000
2000
3000
4000
5000
70
75
80
85
90
95
100
105


avg_income
avg_income
IQ
0
20000
40000
60000
80000
100000
70
75
80
85
90
95
100
105

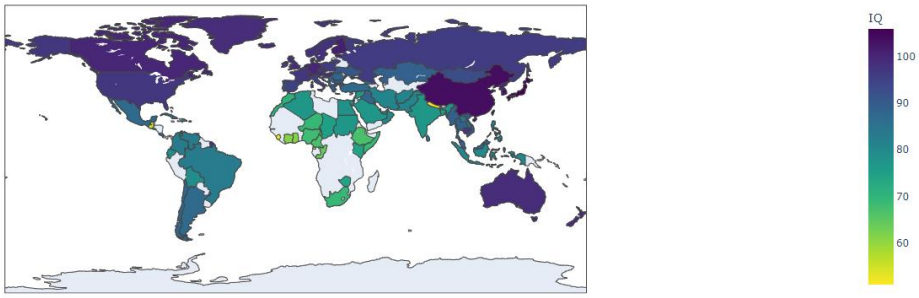

avg_temp
avg_temp
IQ
0
5
10
15
20
25
30
35
70
75
80
85
90
95
100
105

```
cols = ['rank','IQ', 'education_expenditure', 'avg_income','avg_temp']
# Choropleth by Country

for i in cols:

    if i == 'rank':
        fig = px.choropleth(data,
                            locations='country', locationmode='country names',
                            color = i,hover_name="country",
                            title = f'{i} Choropleth',
                            color_continuous_scale='Viridis_r')
        fig.show()
        fig.write_html(f"geo-{i}.html")
    else:
        fig = px.choropleth(data,
                            locations='country', locationmode='country names',
                            color = i,hover_name="country",
                            title = f'{i} Choropleth',
                            color_continuous_scale='Viridis_r')
        fig.show()
        fig.write_html(f"geo-{i}.html")
```

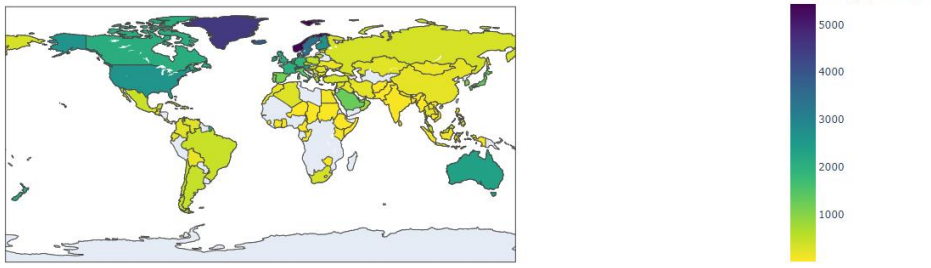
rank Choropleth



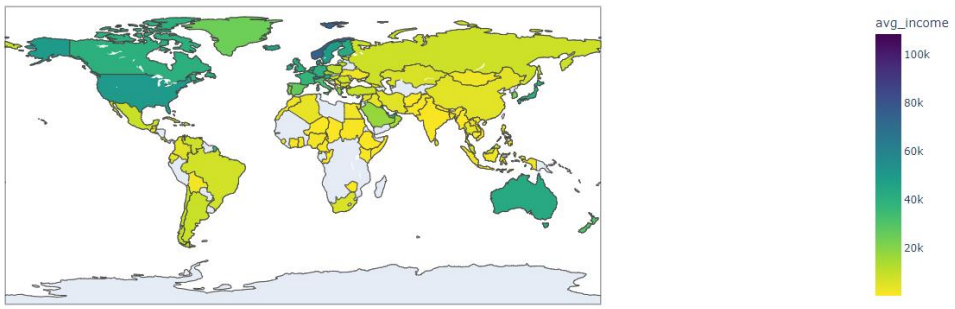
IQ Choropleth



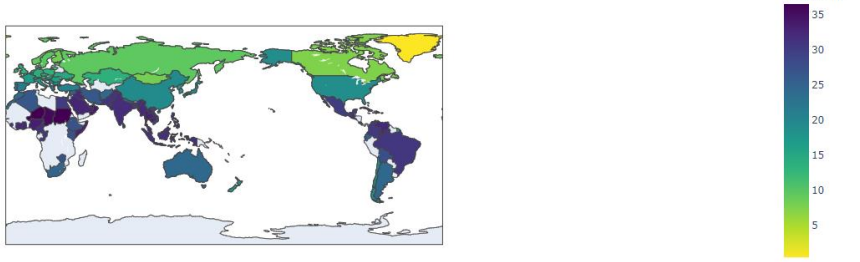
education_expenditure Choropleth



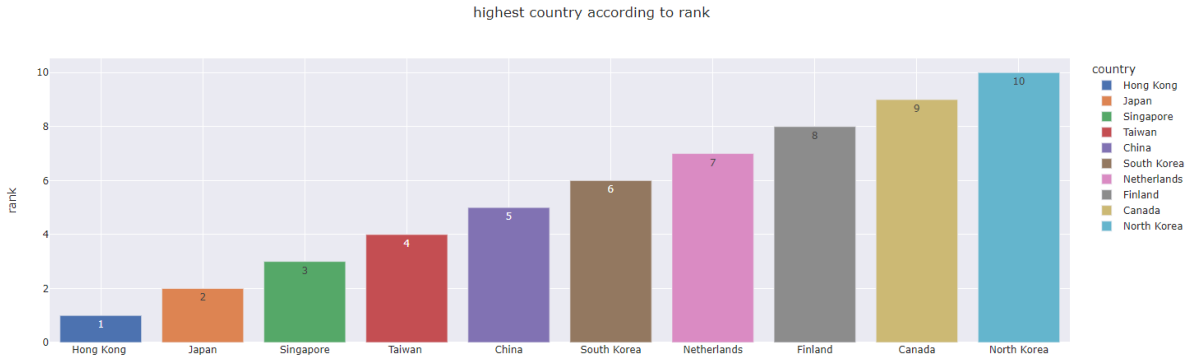
avg_income Choropleth

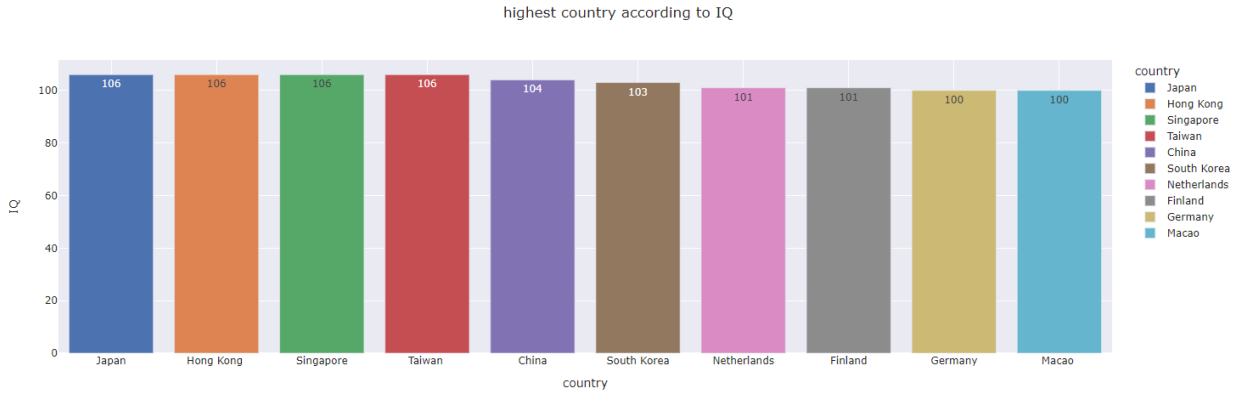


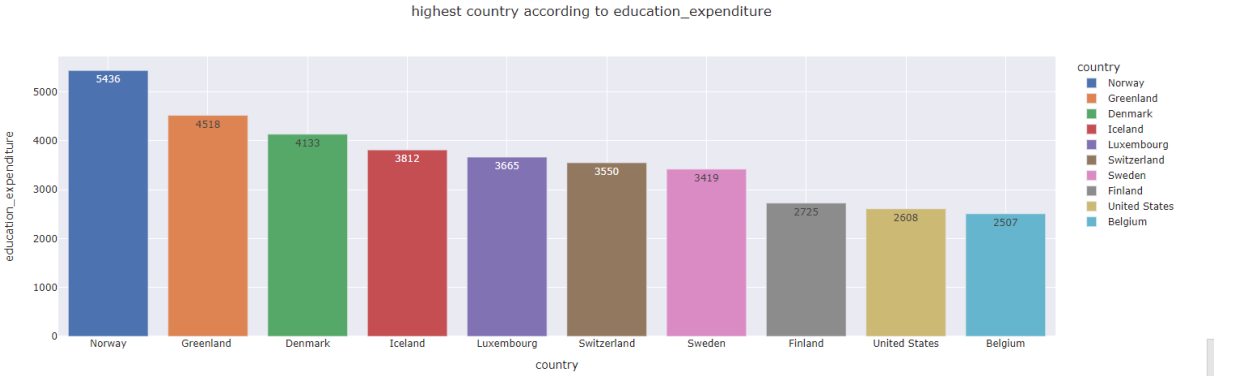
avg_temp Choropleth

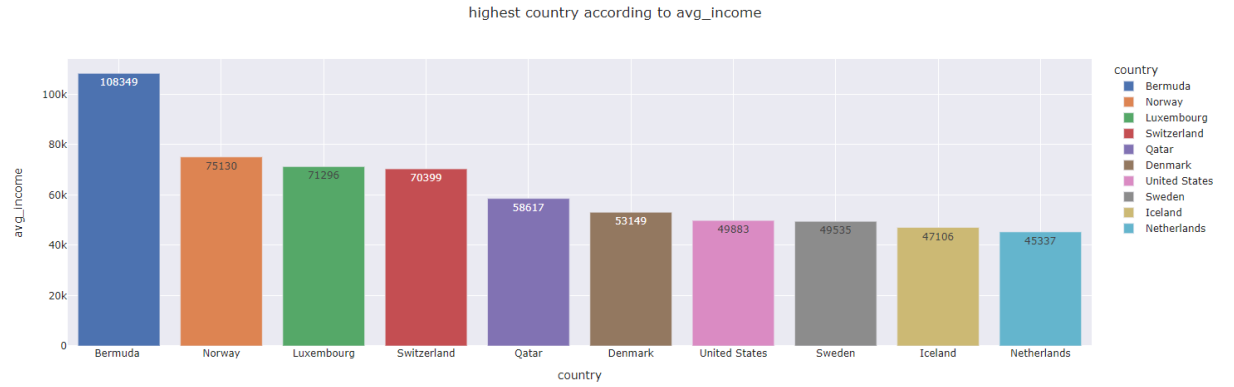


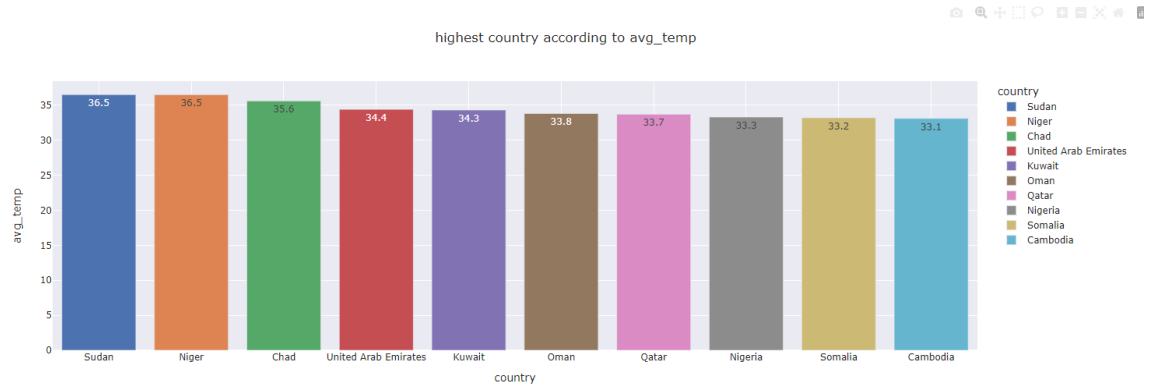

```
# highest country according to Cost / Income

for i in cols:

    if i == 'rank':
        df_country = pd.DataFrame(data.groupby('country')[['country','rank']].sum().sort_values('rank', ascending=True).round(2).head(10))
        fig = px.bar(df_country, x = df_country.country, y = 'rank',
            title = 'highest country according to rank', template = 'seaborn', color = df_country.country, text = 'rank')
        fig.show()
    else:
        df_country = pd.DataFrame(data.groupby('country')[['country',i]].sum().sort_values(i, ascending=False).round(2).head(10))
        fig = px.bar(df_country, x = df_country.country, y = i,
            title = 'highest country according to ' + i, template = 'seaborn', color = df_country.country, text = i)
        fig.show()
```




highest country according to IQ
IQ
country
Japan
Hong Kong
Singapore
Taiwan
China
South Korea
Netherlands
Finland
Germany
Macao
106
106
106
106
104
103
101
101
100
100


highest country according to education_expenditure
education_expenditure
country
Norway
Greenland
Denmark
Iceland
Luxembourg
Switzerland
Sweden
Finland
United States
Belgium
5436
4518
4133
3812
3665
3550
3419
2725
2608
2507


highest country according to avg_income
avg_income
country
Bermuda
Norway
Luxembourg
Switzerland
Qatar
Denmark
United States
Sweden
Iceland
Netherlands
108349
75130
71296
70399
58617
53149
49883
49535
47106
45337


highest country according to avg_temp
avg_temp
country
Sudan
Niger
Chad
United Arab Emirates
Kuwait
Oman
Qatar
Nigeria
Somalia
Cambodia
36.5
36.5
35.6
34.4
34.3
33.8
33.7
33.3
33.2
33.1