Global Quality of Life Analysis

Project submitted to the

APSSDC

Bachelor of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted By

Satyam Sree Bhargavi - S200471

Atmakuri Hymavathi - S200422

Pallanti Mamatha - S200541

Pangi Jahnavi - S200987

Kongara Harshini - S200998

Under the guidance of

K.Meenakshi

M. Ruthuamma

June 2025

1 ABSTRACT

This project is about analyzing the quality of life in different countries around the world using Python. The goal of this project is to understand how factors like safety, health care, climate, pollution, cost of living, and purchasing power vary from one country to another.

We collected data that shows different quality of life indicators and performed step-by-step analysis using Python libraries like Pandas, Matplotlib, and Seaborn. First, we explored the data to understand its structure and cleaned it to remove missing and unwanted values. After that, we grouped and compared the countries based on different factors to see which countries are performing well and which are not.

We used different types of graphs and charts to clearly visualize the patterns and relationships in the data. This project helped us learn how to handle real-world datasets, clean them, analyze them, and present them in an understandable way using visualizations.

The main aim of this project is to provide useful insights about the quality of life in different countries and to help understand which factors are most important in improving living conditions.

2 INTRODUCTION

In today's fast-changing and globally connected world, understanding the quality of life in different countries has become an important area of study. The quality of life is influenced by many factors, including safety, health care, pollution, climate, cost of living, and purchasing power. Analyzing these factors helps us compare countries and understand which regions offer better living conditions for people.

This study focuses on a detailed analysis of quality of life data using Python's popular data analysis libraries such as Pandas, Matplotlib, and Seaborn. By using these tools, we aim to uncover patterns and trends that can provide useful insights into how different countries perform in key life quality indicators.

The dataset used in this project contains valuable information about various life quality factors for different countries. Using Python's Pandas library, we first performed data preprocessing and cleaning to ensure the dataset was accurate and reliable. We explored the data to identify any missing values, corrected them where necessary, and filtered the dataset for meaningful analysis.

With the help of Matplotlib and Seaborn, we created clear and informative visualizations like bar charts, heatmaps, and boxplots to display our findings in a simple and understandable way. These visual tools made it easier to communicate the key insights and comparisons across countries.

By analyzing the global quality of life data using Python, this project helps to highlight which countries are leading in safety, health, and affordability, and which countries need improvements in these areas. The study also shows how different life quality indicators are connected and how they can influence each other.

This project demonstrates how data analysis and visualization can provide valuable information to researchers, policymakers, and people who want to better understand global living standards. Through this study, we gained hands-on experience working with real-world datasets and learned how to present data-driven results that can support decision-making and international comparisons.

3 System Requirements

3.1 Software Requirements

- **Operating System:** Linux
- **Programming Language:** Python 3.x
- **Python Libraries:**
  - **Pandas** – For data cleaning, preprocessing, grouping, and analysis.
  - **Matplotlib** – For creating various data visualizations like bar charts and boxplots.
  - **Seaborn** – For advanced statistical visualizations such as heatmaps.
- **Development Environment:** Jupyter Notebook

3.2 Hardware Requirements

- **Processor:** Minimum 1 GHz or faster processor
- **RAM:** Minimum 4 GB RAM
- **Storage:** Minimum 2 GB free space for storing datasets and project files
- **Display:** 1024 x 768 or higher screen resolution

4 Architecture

The architecture of the Global Quality of Life Analysis using Python, Matplotlib, and Pandas involves several key steps that form a structured and cohesive workflow. The process includes data acquisition, data preprocessing, exploratory data analysis (EDA), data visualization, and insight generation.

4.1 Data Acquisition

The analysis begins with obtaining the Quality of Life dataset. The data is collected from publicly available sources and contains various indicators such as safety, health care, pollution levels, cost of living, climate, and purchasing power. The dataset is loaded into Python using the Pandas library for further analysis.

4.2 Data Preprocessing

Data preprocessing is an essential step to ensure that the dataset is clean, accurate, and ready for analysis. Using Python's Pandas library, the dataset is converted into a DataFrame for easy manipulation. During this step, missing values are handled, duplicate entries are removed, and data types are corrected to maintain consistency and data quality throughout the project.

4.3 Exploratory Data Analysis (EDA)

EDA is used to explore the dataset and gain valuable insights into its structure, distributions, and relationships between variables. Pandas functions are applied to perform summary statistics, groupings, and aggregations to understand key trends across countries. Visualizations created using Matplotlib and Seaborn help in identifying patterns and correlations among the quality of life indicators.

4.4 Data Visualization

Matplotlib and Seaborn are powerful libraries used to create a wide range of visualizations, allowing the presentation of complex global data in an easy-to-understand manner.

Visualizations such as bar charts, boxplots, and heatmaps are used to highlight differences between countries, display top-performing regions, and visually compare quality of life factors.

4.5 Insights and Decision Making

The final step is to draw meaningful insights from the analysis and use these results to understand which countries offer the best quality of life and which areas need improvement. The analysis also helps in understanding how different factors like safety, health care, and cost of living are connected. The results can be presented in clear reports and visual dashboards to help stakeholders easily understand the key findings.

5 Uses of Data Analysis Libraries

5.1 Pandas

- **Data Manipulation:** Enables easy loading, cleaning, and preprocessing of the Global Quality of Life dataset. It allows filtering, grouping, and aggregating data to derive meaningful insights about different countries.

- **Data Exploration:** Facilitates the exploration of country-wise quality indicators, helping to understand patterns in safety, health care, pollution, and other life quality factors.

- **Handling Missing Data:** Functions handle missing or incorrect data points effectively, ensuring the quality of the dataset and preventing errors in the analysis.

- **Data Transformation:** Supports transforming the dataset into the required format by converting data types, applying mathematical operations, and generating new variables when needed.

- **Joining and Merging:** Combines datasets if additional quality-of-life information from other sources is integrated in the future.

5.2 Matplotlib

- **Data Visualization:** Creates visualizations like bar charts, boxplots, and scatter plots to display quality of life comparisons across countries and to highlight major trends.
- **Time Series Analysis:** Can be used to plot time-based trends if the dataset includes multiple years of quality-of-life data.
- **Geospatial Analysis:** Can generate world maps to visualize country-wise variations in safety, health care, and pollution, providing a global perspective.

5.3 Seaborn

- **Enhanced Data Visualization:** Offers more visually appealing and informative visualizations, simplifying the creation of heatmaps, pair plots, and boxplots.
- **Statistical Insights:** Built-in statistical functions help in visualizing correlations between different quality-of-life indicators, such as the relationship between safety and health care.
- **Categorical Data Visualization:** Useful for visualizing country groupings or category-wise comparisons using bar plots and box plots to better understand the differences between countries and regions.

6 Dataset Description

The dataset used in this project is the **Global Quality of Life Dataset**, which provides detailed information about various factors influencing the quality of life across different countries and major cities worldwide.

6.1 Dataset Source

The data was collected from publicly available global repositories such as:

- **Numbeo:** Global quality of life and cost of living indices.
- **World Bank:** Country-level socio-economic data.

- **OECD:** International well-being statistics.
- **Kaggle:** Quality of life datasets and global ranking indices.

6.2 Dataset Features

The dataset consists of the following key attributes:

- **Country:** Name of the country or city.
- **Quality of Life Index:** Composite score representing the overall quality of life.
- **Purchasing Power Index:** Indicates the relative purchasing power of residents.
- **Safety Index:** Measures the safety level of each country (higher values indicate safer countries).
- **Health Care Index:** Reflects the accessibility and quality of healthcare services.
- **Cost of Living Index:** Represents the average cost of living, including expenses like food, housing, and transportation.
- **Property Price to Income Ratio:** Measures housing affordability based on local income levels.
- **Traffic Commute Time:** Average daily commuting time in minutes.
- **Pollution Index:** Indicates environmental pollution levels.
- **Climate Index:** Reflects the favorability of the climate for comfortable living.

6.3 Dataset Size

The dataset contains approximately **250 to 300 entries**, each representing a different country or major city. Each entry consists of **9 to 10 attributes**.

6.4 Dataset Type

- **Format:** Tabular
- **Data Types:** Mixture of numerical and categorical data

6.5 Data Collection Period

The dataset includes data collected from the period of **2021 to 2024**, ensuring that the analysis is based on the most recent and relevant information.

6.6 Purpose of Dataset

The dataset is used to:

- Analyze and compare the quality of life across various countries and regions.
- Identify countries with the highest and lowest quality of life indicators.
- Visualize global trends related to safety, healthcare, pollution, and cost of living.
- Support data-driven decision-making and provide valuable insights for improving living conditions worldwide.

6.7 Significance of Dataset

This dataset serves as a crucial foundation for conducting a comprehensive **Global Quality of Life Analysis**. It enables effective visualization, pattern recognition, and model building for predicting or ranking countries based on multiple life quality indicators.

7 Implementation

The methodology of this project focuses on analyzing and interpreting the Global Quality of Life dataset using Python-based data analysis techniques. The entire process is implemented step-by-step to ensure a clear understanding of the dataset and accurate visualization of key insights.

The methodology is divided into the following phases:

- Importing necessary Python libraries for data analysis and visualization.
- Loading the dataset into the working environment for further processing.
- Performing data analysis and cleaning to ensure the dataset is free from missing values, duplicates, and inconsistencies.
- Filtering the data to select relevant records and attributes that contribute significantly to the quality of life indicators.
- Applying grouping and aggregation techniques to summarize the data and identify meaningful patterns.
- Visualizing the results using appropriate graphs and plots to interpret the global quality of life distribution effectively.

Each step is carefully implemented, and the respective code snippets along with their outputs are provided in the following subsections for better clarity and traceability.

7.1 Importing Libraries

The first step involves importing all the required Python libraries necessary for data analysis and visualization.

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

7.2 Loading the Dataset

The dataset is loaded using Pandas' `read_csv` function to bring the data into the working environment.

```
df = pd.read_csv('Quality_of_Life.csv')
df
```

7.3 Data Analysis and Cleaning

This step involves analyzing the dataset structure and cleaning missing values and duplicates.

```
df.head()
```

Output:

	country	Purchasing Power Value	Purchasing Power Category	Safety Value	Safety Category	Health Care Value	Health Care Category	Climate Value	Climate Category	Cost of Living Value	Cost of Living Category	Property Price to Income Value	Property Price to Income Category	Traffic Commute Time Value
0	Afghanistan	32.15	'Very Low'	25.33	'Low'	24.24	'Low'	0.00	NaN	21.08	'Very Low'	7.8	'Low'	56.17
1	Aland Islands	125.01	'Very High'	71.81	'High'	79.72	'High'	0.00	NaN	53.44	'Low'	5.33	'Low'	19.05
2	Albania	42.82	'Low'	55.52	'Moderate'	48.21	'Moderate'	86.43	'Very High'	40.85	'Low'	14.88	'High'	36.74
3	Alderney	0.00	NaN	83.79	'Very High'	100.00	'Very High'	0.00	NaN	0.00	NaN	0.0	NaN	5.00
4	Algeria	27.60	'Very Low'	47.54	'Moderate'	54.43	'Moderate'	94.82	'Very High'	25.31	'Very Low'	21.7	'Very High'	45.09

```
df.tail()
```

Output:

	country	Purchasing Power Value	Purchasing Power Category	Safety Value	Safety Category	Health Care Value	Health Care Category	Climate Value	Climate Category	Cost of Living Value	Cost of Living Category	Property Price to Income Value	Property Price to Income Category	Traffic Commute Time Value
231	Wallis And Futuna	0.00	NaN	0.00	NaN	0.00	NaN	0.00	NaN	0.00	NaN	0.0	NaN	0.00
232	Western Sahara	0.00	NaN	62.87	'High'	0.00	NaN	0.00	NaN	0.00	NaN	12.75	'High'	0.00
233	Yemen	20.74	'Very Low'	34.07	'Low'	25.31	'Low'	0.00	NaN	48.66	'Low'	15.98	'High'	15.00
234	Zambia	22.32	'Very Low'	54.39	'Moderate'	54.44	'Moderate'	0.00	NaN	36.74	'Very Low'	72.42	'Very High'	38.86
235	Zimbabwe	28.76	'Very Low'	39.31	'Low'	44.80	'Moderate'	96.76	'Very High'	35.36	'Very Low'	17.35	'Very High'	27.79

```
df.size
```

Output:

```
(236, 19)
```

```
df.columns
```

Output:

```
Index(['country', 'Purchasing Power Value', 'Purchasing Power Category',
       'Safety Value', 'Safety Category', 'Health Care Value',
       'Health Care Category', 'Climate Value', 'Climate Category',
       'Cost of Living Value', 'Cost of Living Category',
       'Property Price to Income Value', 'Property Price to Income Category',
       'Traffic Commute Time Value', 'Traffic Commute Time Category',
       'Pollution Value', 'Pollution Category', 'Quality of Life Value',
       'Quality of Life Category'],
      dtype='object')
```

```
df.info()
```

Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 236 entries, 0 to 235
Data columns (total 19 columns):
 #   Column                             Non-Null Count  Dtype
---  ------                             --------------  -----
 0   country                            236 non-null    object
 1   Purchasing Power Value             236 non-null    float64
 2   Purchasing Power Category          190 non-null    object
 3   Safety Value                       236 non-null    float64
 4   Safety Category                    234 non-null    object
 5   Health Care Value                  236 non-null    float64
 6   Health Care Category               221 non-null    object
 7   Climate Value                      236 non-null    float64
 8   Climate Category                   114 non-null    object
 9   Cost of Living Value               236 non-null    float64
 10  Cost of Living Category            191 non-null    object
 11  Property Price to Income Value     236 non-null    object
 12  Property Price to Income Category  215 non-null    object
 13  Traffic Commute Time Value         236 non-null    float64
 14  Traffic Commute Time Category      207 non-null    object
 15  Pollution Value                    236 non-null    float64
 16  Pollution Category                 226 non-null    object
 17  Quality of Life Value              236 non-null    object
 18  Quality of Life Category           114 non-null    object
dtypes: float64(7), object(12)
memory usage: 35.2+ KB
```

```python
df.dtypes
```

Output:

```
country                                 object
Purchasing Power Value                 float64
Purchasing Power Category               object
Safety Value                           float64
Safety Category                         object
Health Care Value                      float64
Health Care Category                    object
Climate Value                          float64
Climate Category                        object
Cost of Living Value                   float64
Cost of Living Category                 object
Property Price to Income Value          object
Property Price to Income Category       object
Traffic Commute Time Value             float64
Traffic Commute Time Category           object
Pollution Value                        float64
Pollution Category                      object
Quality of Life Value                   object
Quality of Life Category                object
dtype: object
```

```python
df.describe()
```

Output:

	Purchasing Power Value	Safety Value	Health Care Value	Climate Value	Cost of Living Value	Traffic Commute Time Value	Pollution Value
count	236.000000	236.000000	236.000000	236.000000	236.000000	236.000000	236.000000
mean	55.573305	55.274449	54.731568	37.598178	37.526314	28.492966	54.266186
std	52.008245	16.914298	20.607381	40.851542	26.026565	17.347242	25.853695
min	0.000000	0.000000	0.000000	-3.540000	0.000000	0.000000	0.000000
25%	16.340000	43.857500	45.807500	0.000000	24.550000	17.100000	35.700000
50%	42.930000	54.635000	57.150000	0.000000	36.895000	29.845000	59.765000
75%	85.940000	68.132500	68.447500	79.332500	51.090000	38.870000	73.740000
max	281.830000	100.000000	100.000000	99.890000	137.370000	100.000000	106.900000

```python
df.isnull().sum()
```

Output:

```
country                                0
Purchasing Power Value                 0
Purchasing Power Category             46
Safety Value                           0
Safety Category                        2
Health Care Value                      0
Health Care Category                  15
Climate Value                          0
Climate Category                     122
Cost of Living Value                   0
Cost of Living Category               45
Property Price to Income Value         0
Property Price to Income Category     21
Traffic Commute Time Value             0
Traffic Commute Time Category         29
Pollution Value                        0
Pollution Category                    10
Quality of Life Value                  0
Quality of Life Category             122
dtype: int64
```

```
df.dropna()
```

Output:

	country	Purchasing Power Value	Purchasing Power Category	Safety Value	Safety Category	Health Care Value	Health Care Category	Climate Value	Climate Category	Cost of Living Value	Cost of Living Category	Property Price to Income Value	Property Price to Income Category	Traffic Commute Time Value	Tra Com Cate
2	Albania	42.82	'Low'	55.52	'Moderate'	48.21	'Moderate'	86.43	'Very High'	40.85	'Low'	14.88	'High'	36.74	'Mod
4	Algeria	27.60	'Very Low'	47.54	'Moderate'	54.43	'Moderate'	94.82	'Very High'	25.31	'Very Low'	21.7	'Very High'	45.09	'
10	Argentina	40.36	'Low'	36.36	'Low'	68.00	'High'	98.28	'Very High'	32.65	'Very Low'	20.05	'Very High'	44.07	'
11	Armenia	36.91	'Very Low'	77.81	'High'	58.07	'Moderate'	63.42	'High'	41.84	'Low'	20.95	'Very High'	29.31	
13	Australia	137.58	'Very High'	52.71	'Moderate'	73.35	'High'	93.80	'Very High'	64.50	'Moderate'	8.25	'Moderate'	37.48	'Mod
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
223	United States	153.02	'Very High'	50.74	'Moderate'	67.86	'High'	79.14	'High'	65.69	'Moderate'	3.54	'Very Low'	32.91	
224	Uruguay	57.92	'Low'	48.20	'Moderate'	68.57	'High'	98.04	'Very High'	47.39	'Low'	12.28	'High'	39.52	'Mod
226	Uzbekistan	44.66	'Low'	71.97	'High'	61.82	'High'	69.79	'High'	24.40	'Very Low'	13.08	'High'	34.30	
230	Vietnam	44.49	'Low'	59.00	'Moderate'	61.32	'High'	71.24	'High'	26.85	'Very Low'	24.77	'Very High'	29.62	
235	Zimbabwe	28.76	'Very Low'	39.31	'Low'	44.80	'Moderate'	96.76	'Very High'	35.36	'Very Low'	17.35	'Very High'	27.79	

```
df.fillna(value=0)
```

Output:

	country	Purchasing Power Value	Purchasing Power Category	Safety Value	Safety Category	Health Care Value	Health Care Category	Climate Value	Climate Category	Cost of Living Value	Cost of Living Category	Property Price to Income Value	Property Price to Income Category	Traffic Commute Time Value	Tra Com Cate
0	Afghanistan	32.15	'Very Low'	25.33	'Low'	24.24	'Low'	0.00	0	21.08	'Very Low'	7.8	'Low'	56.17	'Very
1	Aland Islands	125.01	'Very High'	71.81	'High'	79.72	'High'	0.00	0	53.44	'Low'	5.33	'Low'	19.05	'Very
2	Albania	42.82	'Low'	55.52	'Moderate'	48.21	'Moderate'	86.43	'Very High'	40.85	'Low'	14.88	'High'	36.74	'Mod
3	Alderney	0.00	0	83.79	'Very High'	100.00	'Very High'	0.00	0	0.00	0	0.0	0	5.00	'Very
4	Algeria	27.60	'Very Low'	47.54	'Moderate'	54.43	'Moderate'	94.82	'Very High'	25.31	'Very Low'	21.7	'Very High'	45.09	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
231	Wallis And Futuna	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.0	0	0.00	
232	Western Sahara	0.00	0	62.87	'High'	0.00	0	0.00	0	0.00	0	12.75	'High'	0.00	
233	Yemen	20.74	'Very Low'	34.07	'Low'	25.31	'Low'	0.00	0	48.66	'Low'	15.98	'High'	15.00	'Very
234	Zambia	22.32	'Very Low'	54.39	'Moderate'	54.44	'Moderate'	0.00	0	36.74	'Very Low'	72.42	'Very High'	38.86	'Mod

```python
df.duplicated().sum()
```

Output:

```
country                                 236
Purchasing Power Value                  189
Purchasing Power Category                 5
Safety Value                            231
Safety Category                           5
Health Care Value                       204
Health Care Category                      5
Climate Value                           115
Climate Category                          4
Cost of Living Value                    190
Cost of Living Category                   5
Property Price to Income Value          204
Property Price to Income Category         5
Traffic Commute Time Value              184
Traffic Commute Time Category             5
Pollution Value                         222
Pollution Category                        5
Quality of Life Value                   113
Quality of Life Category                  5
dtype: int64
```

```python
df.drop_duplicates()
```

Output:

	country	Purchasing Power Value	Purchasing Power Category	Safety Value	Safety Category	Health Care Value	Health Care Category	Climate Value	Climate Category	Cost of Living Value	Cost of Living Category	Property Price to Income Value	Property Price to Income Category	Traffic Commute Time Value	C
0	Afghanistan	32.15	'Very Low'	25.33	'Low'	24.24	'Low'	0.00	NaN	21.08	'Very Low'	7.8	'Low'	56.17	'
1	Aland Islands	125.01	'Very High'	71.81	'High'	79.72	'High'	0.00	NaN	53.44	'Low'	5.33	'Low'	19.05	'
2	Albania	42.82	'Low'	55.52	'Moderate'	48.21	'Moderate'	86.43	'Very High'	40.85	'Low'	14.88	'High'	36.74	'
3	Alderney	0.00	NaN	83.79	'Very High'	100.00	'Very High'	0.00	NaN	0.00	NaN	0.0	NaN	5.00	'
4	Algeria	27.60	'Very Low'	47.54	'Moderate'	54.43	'Moderate'	94.82	'Very High'	25.31	'Very Low'	21.7	'Very High'	45.09	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
231	Wallis And Futuna	0.00	NaN	0.00	NaN	0.00	NaN	0.00	NaN	0.00	NaN	0.0	NaN	0.00	
232	Western Sahara	0.00	NaN	62.87	'High'	0.00	NaN	0.00	NaN	0.00	NaN	12.75	'High'	0.00	
233	Yemen	20.74	'Very Low'	34.07	'Low'	25.31	'Low'	0.00	NaN	48.66	'Low'	15.98	'High'	15.00	'
234	Zambia	22.32	'Very Low'	54.39	'Moderate'	54.44	'Moderate'	0.00	NaN	36.74	'Very Low'	72.42	'Very High'	38.86	'

```python
df.nunique()
```

Output:

```
country                              236
Purchasing Power Value               189
Purchasing Power Category              5
Safety Value                         231
Safety Category                        5
Health Care Value                    204
Health Care Category                   5
Climate Value                        115
Climate Category                       4
Cost of Living Value                 190
Cost of Living Category                5
Property Price to Income Value       204
Property Price to Income Category      5
Traffic Commute Time Value           184
Traffic Commute Time Category          5
Pollution Value                      222
Pollution Category                     5
Quality of Life Value                113
Quality of Life Category               5
dtype: int64
```

```python
df['Quality of Life Category'].value_counts()
```

7.4 Data Filtering

Filtering the dataset to select records relevant to the analysis.

```
# Top 10 Countries by Quality of Life Score
df['Quality of Life Value'] = pd.to_numeric(df['Quality of Life Value'], errors='coerce')

top_qol = df.sort_values(by='Quality of Life Value', ascending=False).head(10)
print(top_qol[['country', 'Quality of Life Value']])
```

Output:

```
                country  Quality of Life Value
0           Afghanistan                    0.0
1         Aland Islands                    0.0
3              Alderney                    0.0
5        American Samoa                    0.0
6               Andorra                    0.0
7                Angola                    0.0
8              Anguilla                    0.0
9   Antigua And Barbuda                    0.0
12                Aruba                    0.0
16              Bahamas                    0.0
```

```
#Average Score for Each Metric
df[['Purchasing Power Value', 'Safety Value', 'Health Care Value',
    'Climate Value', 'Cost of Living Value', 'Traffic Commute Time Value',
    'Pollution Value', 'Quality of Life Value']].mean()
```

Output:

	0
Purchasing Power Value	55.573305
Safety Value	55.274449
Health Care Value	54.731568
Climate Value	37.598178
Cost of Living Value	37.526314
Traffic Commute Time Value	28.492966
Pollution Value	54.266186
Quality of Life Value	0.000000

dtype: float64

```
#Find Countries with Very High Safety
df[df['Safety Category'] == "'Very High'"][['country', 'Safety
    Value']]
```

Output:

	country	Safety Value
3	Alderney	83.79
6	Andorra	84.71
68	Faroe Islands	83.57
111	Kiribati	81.86
121	Liechtenstein	87.84
141	Montserrat	93.38
159	Oman	81.94
171	Qatar	84.05
177	Saint Helena	92.65
181	Saint-Pierre And Miquelon	100.00
183	San Marino	80.65
184	Sao Tome And Principe	82.56
206	Taiwan	82.89
218	Tuvalu	81.52

```python
# Best Countries with High Safety and Low Pollution
best_countries = df[(df['Safety Value'] > 70) & (df['Pollution
    Value'] < 30)]
print(best_countries[['country', 'Safety Value', 'Pollution Value
    ']])
```

Output:

```
          country  Safety Value  Pollution Value
1   Aland Islands         71.81            18.05
3        Alderney         83.79             1.72
6         Andorra         84.71            22.98
8        Anguilla         75.44             0.00
14        Austria         70.38            20.73
55        Denmark         73.81            20.75
63        Eritrea         72.86            15.52
64        Estonia         76.37            17.02
68  Faroe Islands         83.57             4.91
70        Finland         73.57            11.83
87       Guernsey         76.14            11.49
95        Iceland         74.26            15.84
101   Isle Of Man         79.57            18.12
107        Jersey         74.32            26.72
121 Liechtenstein         87.84             6.47
141    Montserrat         93.38             0.00
148   Netherlands         73.36            21.14
161         Palau         73.24            14.37
177  Saint Helena         92.65            15.52
183    San Marino         80.65            15.36
193      Slovenia         76.17            22.25
204   Switzerland         73.69            23.06
218        Tuvalu         81.52             0.00
```

```python
df.describe(include='all')
```

Output:

	country	Purchasing Power Value	Purchasing Power Category	Safety Value	Safety Category	Health Care Value	Health Care Category	Climate Value	Climate Category	Cost of Living Value	Cost of Living Category	Property Price to Income Value	Property Price to Income Category
count	236	236.000000	190	236.000000	234	236.000000	221	236.000000	114	236.000000	191	236	215
unique	236	NaN	5	NaN	5	NaN	5	NaN	4	NaN	5	204	5
top	Afghanistan	NaN	'Very Low'	NaN	'Moderate'	NaN	'High'	NaN	'Very High'	NaN	'Very Low'	0.0	'Very High'
freq	1	NaN	63	NaN	100	NaN	93	NaN	58	NaN	88	22	74
mean	NaN	55.573305	NaN	55.274449	NaN	54.731568	NaN	37.598178	NaN	37.526314	NaN	NaN	NaN
std	NaN	52.008245	NaN	16.914298	NaN	20.607381	NaN	40.851542	NaN	26.026565	NaN	NaN	NaN
min	NaN	0.000000	NaN	0.000000	NaN	0.000000	NaN	-3.540000	NaN	0.000000	NaN	NaN	NaN
25%	NaN	16.340000	NaN	43.857500	NaN	45.807500	NaN	0.000000	NaN	24.550000	NaN	NaN	NaN
50%	NaN	42.930000	NaN	54.635000	NaN	57.150000	NaN	0.000000	NaN	36.895000	NaN	NaN	NaN
75%	NaN	85.940000	NaN	68.132500	NaN	68.447500	NaN	79.332500	NaN	51.090000	NaN	NaN	NaN
max	NaN	281.830000	NaN	100.000000	NaN	100.000000	NaN	99.890000	NaN	137.370000	NaN	NaN	NaN

```python
# Remove colon (:) and convert to float
df['Quality of Life Value'] = df['Quality of Life Value'].replace
    (r":\s*", "", regex=True).astype(float)
```

```python
df.replace(0, np.nan, inplace=True)
```

```python
#Drop rows with >50% missing values
df = df[df.isnull().mean(axis=1) < 0.5]
```

```python
#Encode Ordinal Categories
category_map = {
    'Very Low': 1,
    'Low': 2,
    'Moderate': 3,
    'High': 4,
    'Very High': 5,
    0: np.nan  # Treat 0 as missing
}

category_cols = [col for col in df.columns if 'Category' in col]

for col in category_cols:
    df[col] = df[col].replace(category_map)
```

```python
# Filter countries where pollution is very high and safety is low
high_risk_countries = df[(df['Pollution Value'] > 75) & (df['Safety Value'] < 40)]
high_risk_countries
```

Output:

	country	Purchasing Power Value	Purchasing Power Category	Safety Value	Safety Category	Health Care Value	Health Care Category	Climate Value	Climate Category	Cost of Living Value	Cost of Living Category	Property Price to Income Value	Property Price to Income Category	Traffic Commute Time Value	Tra Co Cat
0	Afghanistan	32.15	'Very Low'	25.33	'Low'	24.24	'Low'	NaN	NaN	21.08	'Very Low'	7.8	'Low'	56.17	'Very
7	Angola	224.46	'Very High'	33.71	'Low'	36.58	'Low'	NaN	NaN	42.57	'Low'	8.01	'Moderate'	65.43	'Very
18	Bangladesh	36.16	'Very Low'	38.79	'Low'	42.20	'Moderate'	71.29	'High'	20.76	'Very Low'	12.26	'High'	57.57	'Very
37	Cameroon	10.02	'Very Low'	34.53	'Low'	45.53	'Moderate'	NaN	NaN	37.61	'Very Low'	46.06	'Very High'	46.00	
41	Central African Republic	NaN	NaN	15.89	'Very Low'	41.67	'Moderate'	NaN	NaN	NaN	NaN	8.5	'Moderate'	5.00	'Ver
43	Chile	51.47	'Low'	39.55	'Low'	63.52	'High'	90.21	'Very High'	36.29	'Very Low'	16.03	'Very High'	36.04	'Mo
54	Democratic Republic of the Congo	25.51	'Very Low'	36.16	'Low'	25.69	'Low'	NaN	NaN	50.85	'Low'	37.22	'Very High'	75.00	'Very
75	Gabon	20.07	'Very Low'	39.86	'Low'	31.48	'Low'	NaN	NaN	48.70	'Low'	69.71	'Very High'	16.35	'Ver
91	Haiti	77.70	'Moderate'	21.41	'Low'	32.72	'Low'	NaN	NaN	51.39	'Low'	5.5	'Low'	80.67	'Very
92	Honduras	43.68	'Low'	27.75	'Low'	37.83	'Low'	NaN	NaN	34.39	'Very Low'	6.36	'Low'	19.70	'Ver

```python
# Filter countries where people earn less but living costs are high
costly_countries = df[(df['Purchasing Power Value'] < 50) & (df['Cost of Living Value'] > 50)]

costly_countries
```

Output:

	country	Purchasing Power Value	Purchasing Power Category	Safety Value	Safety Category	Health Care Value	Health Care Category	Climate Value	Climate Category	Cost of Living Value	Cost of Living Category	Property Price to Income Value	Property Price to Income Category	Traffic Commute Time Value	Tra Com Cate
19	Barbados	46.93	'Low'	55.14	'Moderate'	71.88	'High'	NaN	NaN	70.60	'Moderate'	14.67	'High'	36.67	'Mod
54	Democratic Republic of the Congo	25.51	'Very Low'	36.16	'Low'	25.69	'Low'	NaN	NaN	50.85	'Low'	37.22	'Very High'	75.00	'Very
57	Dominica	32.92	'Very Low'	46.42	'Moderate'	36.57	'Low'	NaN	NaN	50.36	'Low'	0.64	'Very Low'	NaN	
62	Equatorial Guinea	30.87	'Very Low'	52.16	'Moderate'	42.59	'Moderate'	NaN	NaN	51.87	'Low'	7.15	'Low'	20.50	'Very
83	Grenada	31.88	'Very Low'	75.07	'High'	65.28	'High'	NaN	NaN	75.44	'High'	175.23	'Very High'	40.00	'Mod
119	Liberia	33.34	'Very Low'	19.13	'Very Low'	38.89	'Low'	NaN	NaN	64.10	'Moderate'	25.57	'Very High'	30.00	
136	Micronesia	14.28	'Very Low'	55.87	'Moderate'	53.47	'Moderate'	NaN	NaN	56.24	'Low'	14.77	'High'	17.17	'Very
152	Niger	7.04	'Very Low'	33.38	'Low'	30.56	'Low'	NaN	NaN	132.99	'Very High'	'2,746.00'	'Very High'	20.00	'Very
164	Papua New Guinea	13.71	'Very Low'	19.89	'Very Low'	23.86	'Low'	NaN	NaN	69.30	'Moderate'	450.4	'Very High'	44.56	
172	Republic of the Congo	20.45	'Very Low'	45.71	'Moderate'	68.52	'High'	NaN	NaN	54.46	'Low'	35.01	'Very High'	NaN	

```python
# Filter for countries offering good quality of life and low commute burden
ideal_living = df[(df['Quality of Life Value'] > 80) & (df['Traffic Commute Time Value'] < 25)]

ideal_living
```

Output:

country	Purchasing Power Value	Purchasing Power Category	Safety Value	Safety Category	Health Care Value	Health Care Category	Climate Value	Climate Category	Cost of Living Value	Cost of Living Category	Property Price to Income Value	Property Price to Income Category	Traffic Commute Time Value	Tra Co Cate

7.5 Data Grouping and Aggregations

Grouping the dataset and applying aggregation functions to summarize the data.

```python
# Group by Safety Category (Mean)
safety_group = df.groupby('Safety Category')['Quality of Life Value'].mean()
print("Grouped by Safety Category (Mean Quality of Life Value):\n")
print(safety_group)
```

Output:

```
Grouped by Safety Category (Mean Quality of Life Value):

Safety Category
'High'         NaN
'Low'          NaN
'Moderate'     NaN
'Very High'    NaN
'Very Low'     NaN
Name: Quality of Life Value, dtype: float64
```

```python
# Group by Purchasing Power Category (Mean)
purchasing_power_group = df.groupby('Purchasing Power Category')['Purchasing Power Value'].mean()
print("\nGrouped by Purchasing Power Category (Mean Purchasing Power Value):\n")
print(purchasing_power_group)
```

Output:

```
Grouped by Purchasing Power Category (Mean Purchasing Power Value):

Purchasing Power Category
'High'          98.842381
'Low'           49.525556
'Moderate'      74.265238
'Very High'    145.565128
'Very Low'      24.323651
Name: Purchasing Power Value, dtype: float64
```

```
# Group by Health Care Category (Mean)
health_care_group = df.groupby('Health Care Category')['Health
    Care Value'].mean()
print("\nGrouped by Health Care Category (Mean Health Care Value)
    :\n")
print(health_care_group)
```

Output:

```
Grouped by Health Care Category (Mean Health Care Value):

Health Care Category
'High'            69.276778
'Low'             33.508800
'Moderate'        51.239080
'Very High'       90.328333
Name: Health Care Value, dtype: float64
```

```
# Group by Climate Category (Mean)
climate_group = df.groupby('Climate Category')['Climate Value'].
    mean()
print("\nGrouped by Climate Category (Mean Climate Value):\n")
print(climate_group)
```

Output:

```
Grouped by Climate Category (Mean Climate Value):

Climate Category
'High'            71.001463
'Low'             -3.540000
'Moderate'        48.811429
'Very High'       91.073966
Name: Climate Value, dtype: float64
```

```
# Group by Pollution Category (Mean)
pollution_group = df.groupby('Pollution Category')['Pollution
    Value'].mean()
print("\nGrouped by Pollution Category (Mean Pollution Value):\n"
    )
print(pollution_group)
```

Output:

```
Grouped by Pollution Category (Mean Pollution Value):

Pollution Category
'High'          69.712222
'Low'           30.380000
'Moderate'      51.535000
'Very High'     88.064643
'Very Low'      14.209333
Name: Pollution Value, dtype: float64
```

```
# Aggregation for Quality of Life Value
safety_agg = df.groupby('Safety Category')['Quality of Life Value
    '].agg(['mean', 'sum', 'max', 'min', 'count'])
print("\nAggregated Quality of Life Value by Safety Category:\n")
print(safety_agg)
```

Output:

```
Aggregated Quality of Life Value by Safety Category:

                 mean  sum  max  min  count
Safety Category
'High'            NaN  0.0  NaN  NaN      0
'Low'             NaN  0.0  NaN  NaN      0
'Moderate'        NaN  0.0  NaN  NaN      0
'Very High'       NaN  0.0  NaN  NaN      0
'Very Low'        NaN  0.0  NaN  NaN      0
```

```python
# Aggregation for Purchasing Power Value
purchasing_power_agg = df.groupby('Purchasing Power Category')['Purchasing Power Value'].agg(['mean', 'sum', 'max', 'min', 'count'])
print("\nAggregated Purchasing Power Value by Purchasing Power Category:\n")
print(purchasing_power_agg)
```

Output:

```
Aggregated Purchasing Power Value by Purchasing Power Category:

                                 mean      sum     max     min  count
Purchasing Power Category
'High'                      98.842381  2075.69  109.08   85.89     21
'Low'                       49.525556  2228.65   59.69   40.26     45
'Moderate'                  74.265238  1559.57   84.66   62.54     21
'Very High'                145.565128  5677.04  281.83  111.31     39
'Very Low'                  24.323651  1532.39   39.16    2.55     63
```

```python
# Aggregation for Health Care Value
health_care_agg = df.groupby('Health Care Category')['Health Care Value'].agg(['mean', 'sum', 'max', 'min', 'count'])
print("\nAggregated Health Care Value by Health Care Category:\n")
print(health_care_agg)
```

Output:

```
Aggregated Health Care Value by Health Care Category:

                           mean      sum     max    min  count
Health Care Category
'High'                69.276778  6234.91   79.81  60.65     90
'Low'                 33.508800   837.72   39.52  23.61     25
'Moderate'            51.239080  4457.80   59.38  40.49     87
'Very High'           90.328333   541.97  100.00  82.78      6
```

```python
# Aggregation for Climate Value
climate_agg = df.groupby('Climate Category')['Climate Value'].agg
    (['mean', 'sum', 'max', 'min', 'count'])
print("\nAggregated Climate Value by Climate Category:\n")
print(climate_agg)
```

Output:

```
Aggregated Climate Value by Climate Category:

                       mean      sum    max    min  count
Climate Category
'High'            71.001463  2911.06  79.61  60.75     41
'Low'             -3.540000    -3.54  -3.54  -3.54      1
'Moderate'        48.811429   683.36  59.43  20.22     14
'Very High'       91.073966  5282.29  99.89  80.48     58
```

```python
# Aggregation for Pollution Value
pollution_agg = df.groupby('Pollution Category')['Pollution Value
    '].agg(['mean', 'sum', 'max', 'min', 'count'])
print("\nAggregated Pollution Value by Pollution Category:\n")
print(pollution_agg)
```

Output:

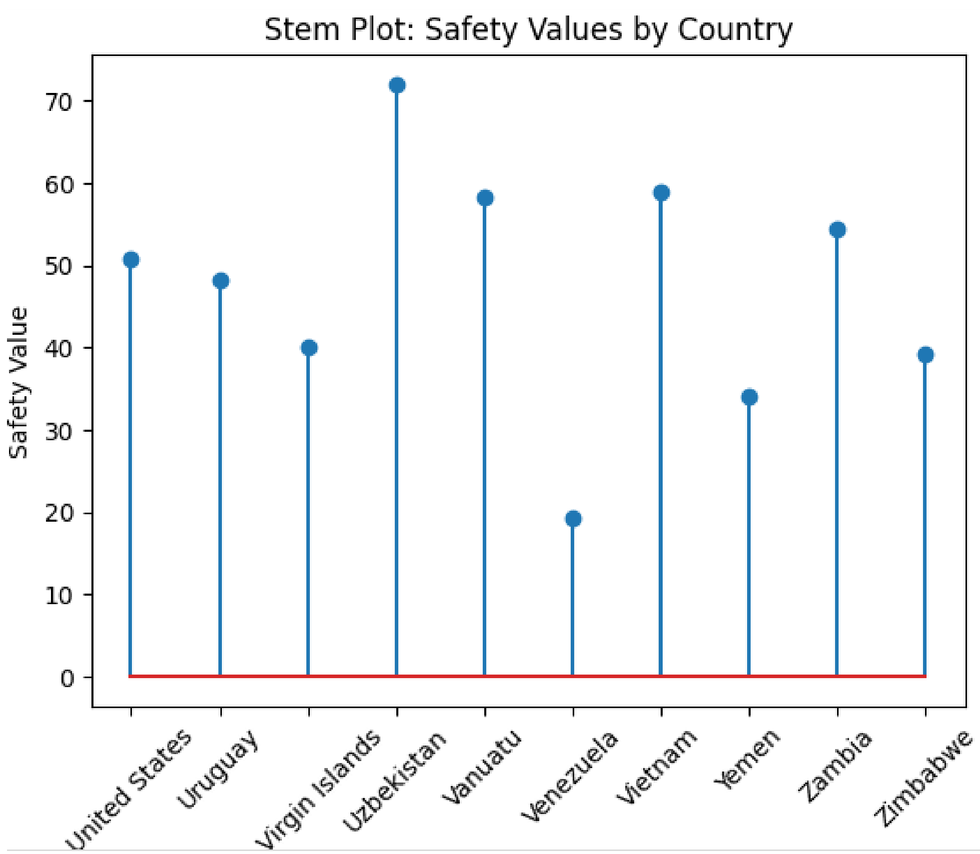
```
Aggregated Pollution Value by Pollution Category:

                         mean      sum     max    min  count
Pollution Category
'High'              69.712222  5646.69   79.59  60.00     81
'Low'               30.380000  1124.06   39.22  20.69     37
'Moderate'          51.535000  2370.61   59.85  40.14     46
'Very High'         88.064643  2465.81  106.90  80.90     28
'Very Low'          14.209333   213.14   18.32   1.72     15
```

7.6 Data Visualization

Visualizing the key trends and relationships in the dataset using various plots.

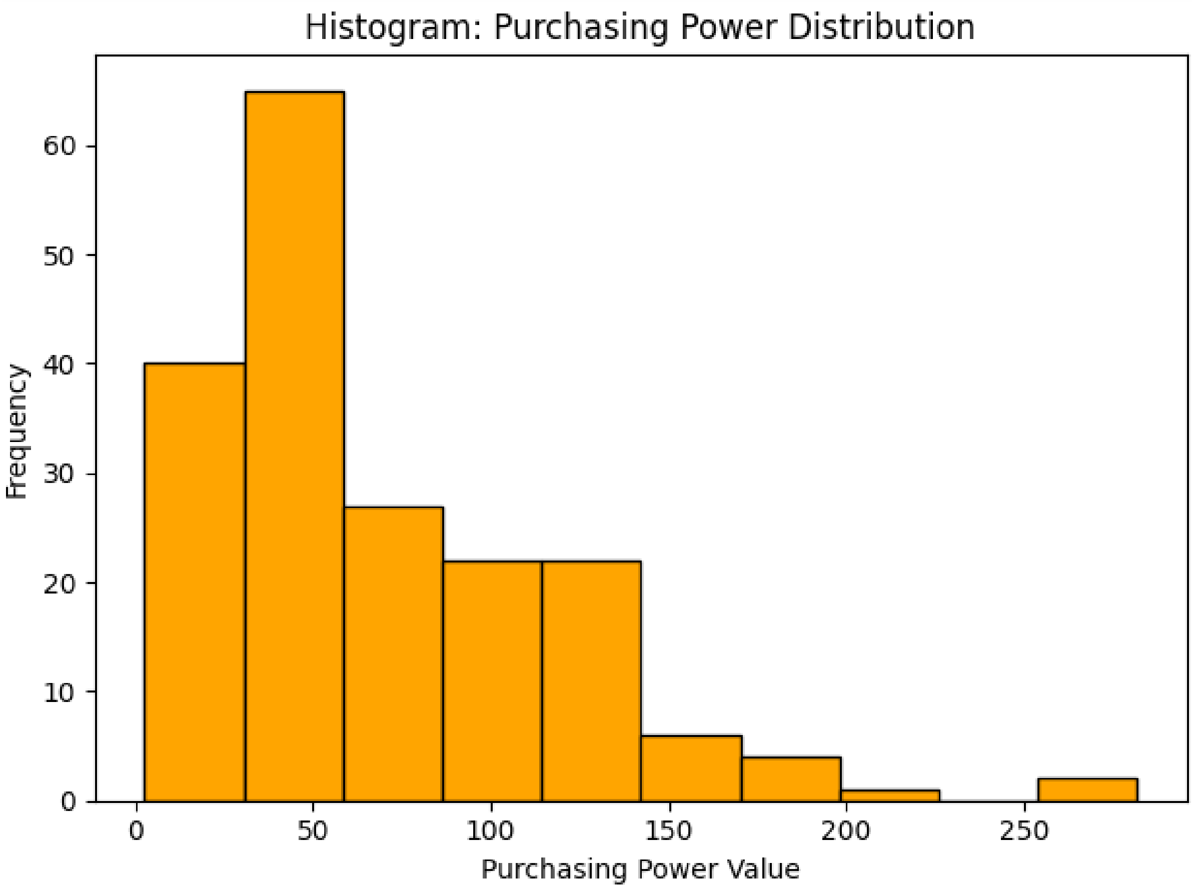
```
#Visualize Safety Values clearly across countries.
df['Safety Value'] = pd.to_numeric(df['Safety Value'], errors='coerce')

df_clean = df[['country', 'Safety Value']].dropna().tail(10)

# Stem Plot
plt.stem(df_clean['country'], df_clean['Safety Value'])
plt.xticks(rotation=45)
plt.title("Stem Plot: Safety Values by Country")
plt.xlabel("Country")
plt.ylabel("Safety Value")
plt.show()
```

Output:



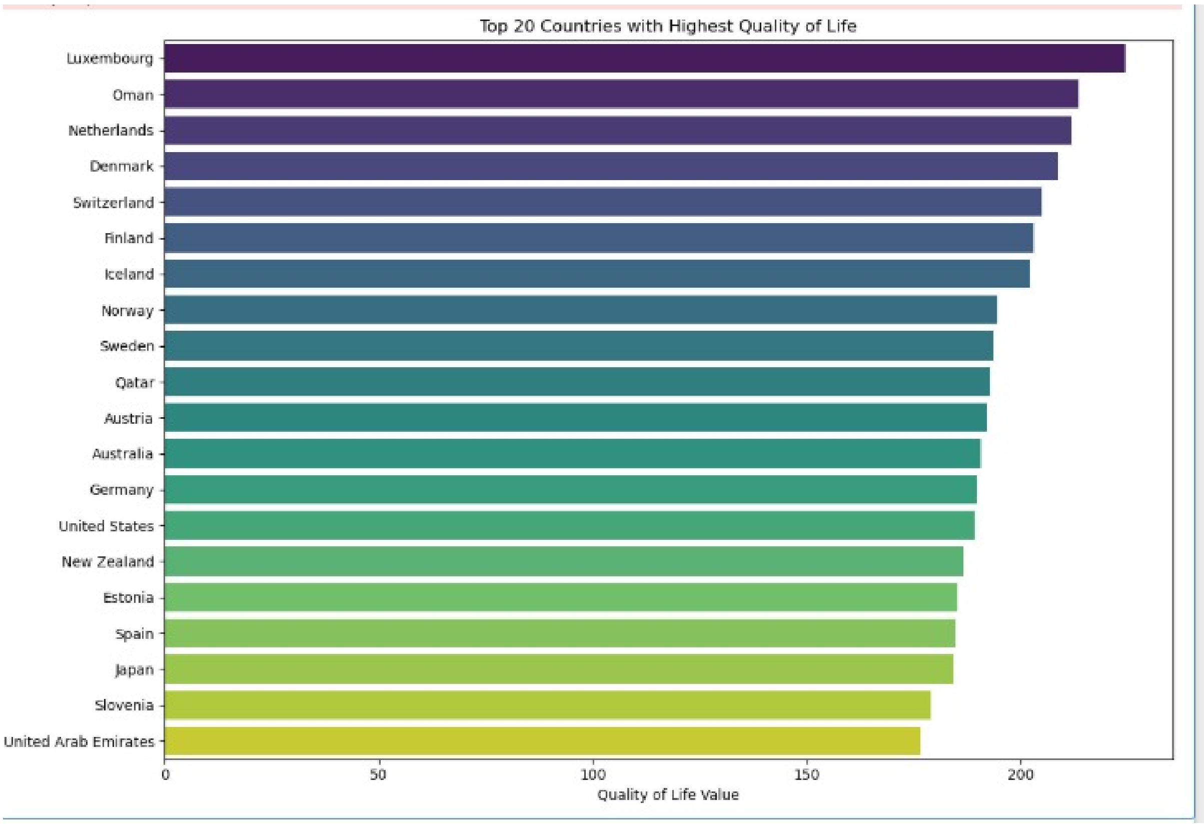
```
#See the distribution of Purchasing Power across countries.
df['Purchasing Power Value'] = pd.to_numeric(df['Purchasing Power
    Value'], errors='coerce')

df_clean = df['Purchasing Power Value'].dropna()

# Histogram
plt.hist(df_clean, bins=10, color='orange', edgecolor='black')
plt.title("Histogram: Purchasing Power Distribution")
plt.xlabel("Purchasing Power Value")
plt.ylabel("Frequency")
plt.tight_layout()
plt.show()
```

Output:



```
#top countries for highest quality of life
# Clean 'Quality of Life Value'
df['Quality of Life Value'] = df['Quality of Life Value'].apply(
    extract_numeric)

# Drop missing and zero/negative values
df_clean = df.dropna(subset=['Quality of Life Value'])
df_clean = df_clean[df_clean['Quality of Life Value'] > 0]

# Select top 20 countries with highest Quality of Life
top_qol_df = df_clean[['country', 'Quality of Life Value']].
    sort_values(
     by='Quality of Life Value', ascending=False).head(20)

# Plotting
plt.figure(figsize=(12, 8))
sns.barplot(
    y='country',
    x='Quality of Life Value',
    data=top_qol_df,
    palette='viridis'
)

plt.title('Top 20 Countries with Highest Quality of Life')
plt.xlabel('Quality of Life Value')
plt.ylabel('Country')
plt.tight_layout()
plt.show()
```

Output:



8 Advantages and Disadvantages

8.1 Advantages

- **Clear Global Comparison:**

  The project helps compare quality of life factors across different countries using simple and clear visualizations.

- **Skill Development:**

  It improves data handling skills, including loading, filtering, grouping, and managing missing values.

- **Python Ecosystem:**

  Python libraries like Pandas, Matplotlib, and Seaborn make data cleaning, analysis, and visualization efficient.

- **Practical and Extendable:**

  The project structure is flexible and can be extended with more datasets, dashboards, or advanced analyses.

- **Data-Driven Decision Support:**

  It supports data-driven decision-making and can help researchers and policymakers understand global living conditions.

8.2 Disadvantages

- **Dataset Accuracy Dependency:**

  The analysis fully depends on the accuracy and completeness of the dataset. Incomplete or incorrect data can mislead results.

- **Dataset Coverage Limitation:**

  The dataset may not include all countries or all quality of life indicators, limiting the comparison scope.

- **No Predictive Analysis:**

  The project is based only on existing data and does not perform predictive analysis or future forecasting.

- **Static Data:**

  The project does not use real-time or regularly updated data, which may reduce the relevance of the findings over time.

- **Basic Visualizations:**

  The visualizations are limited to 2D charts and do not include interactive or advanced visual exploration tools.

9 CONCLUSION

In this project, we successfully analyzed the quality of life across different countries using Python and data visualization tools. We explored how different factors like safety, health care, pollution, cost of living, and purchasing power affect living conditions around the world.

Using libraries like Pandas, Matplotlib, and Seaborn, we were able to clean the dataset, group the countries, and create useful visualizations to easily understand the differences between countries. The project helped us to gain valuable hands-on experience in data analysis, data cleaning, and creating meaningful visualizations.

Through this project, we learned how to handle real-world datasets and present useful insights in a simple and understandable way. The analysis can also help researchers, policymakers, and organizations to compare countries and understand which factors are most important for improving the quality of life.

This project has provided a good foundation for working with large datasets and can be improved further by adding more indicators, real-time data, or building interactive dashboards in the future.

10 REFERENCES

- Analytics Vidhya. (2022). Data Analysis Project for Beginners using Python.
- Make Me Analyst. Python Libraries for Data Analysis.
- Reback, J., et al. Pandas v1.2.4. Zenodo.
- AP Skill Development Corporation. SRM Data Analysis Summer Internship 2023.