Project Title:

LifeExpectancy DataAnalysis

Students:
Mohammadmahdi Zeynali
Alireza Soori
Alireza Jahanbakhsh
Professor:
Dr. Maryam Babaei
Teaching assistant:
Mr. Matin Jaberi
Mr. Sepehr Rezaei

Students ID: **402222079** **400222053** **400222068**	***Recipient organization name:*** **Shahid Beheshti University**
Project link:	***E-mail:*** m.zeynal.gh@gmail.com alirezasr8228@gmail.com a.jahanbakhsh@mail.sbu.ac.ir

Contents

1. Description of the dataset

1.1. About Dataset

Data contains life expectancy, health, immunization, and economic and demographic information about **179 countries** from **2000-2015 years**. The adjusted dataset has **21 variables** and **2.864 rows**.
Data were initially collected from Kaggle Source.
The dataset had inaccurate data and a lot of values were missing.

The dataset is completely updated.

Data about Population, GDP, and Life Expectancy was updated according to World Bank Data. Information about vaccinations for Measles, Hepatitis B, Polio, and Diphtheria, alcohol consumption, BMI, HIV incidents, mortality rates, and thinness were collected from World Health Organization public datasets. Information about Schooling was collected from the Our World in Data which is a project of the University of Oxford.

The database has one variable that categorizes countries into two groups: **Developed vs Developing** countries. According to World Trade Organization, each country defines itself as "Developed" or "Developing". Therefore, it is challenging to categorize countries. UN has a list dated 2014 that for analytical purposes classifies countries as developed, in transition, and developing economies. Countries that have economies in transition have similar characteristics to the countries that are categorized as developed or developing countries. Countries have been grouped according to their Gross National Income per capita. As a result, nations were divided into four income groups: high-income, higher-middle-income, lower-middle-income, and low-income. The levels of Gross Domestic Income are set by the World Bank to ensure comparability.

1.2. Introducing features

1.2.1. Target feature

Life_expectancy: Average life expectancy of both sexes in different years from 2010 to 2015 in countries

1.2.2. Auxiliary feature

Country: List of the 179 countries

Region: 179 countries are distributed in 9 regions. E.g. Africa, Asia, Oceania, European Union, Rest of Europe

Year: Years observed from 2000 to 2015

Infant_deaths: Represents infant deaths per 1000 population

Under_five_deaths: Represents deaths of children under five years old per 1000 population

Adult_mortality: Represents deaths of adults per 1000 population

Alcohol_consumption: Represents alcohol consumption that is recorded in liters of pure alcohol per capita with 15+ years old

Hepatitis_B: Represents % of coverage of Hepatitis B (HepB3) immunization among 1-year-olds.

Measles: Represents % of coverage of Measles containing vaccine first dose (MCV1) immunization among 1-year-olds

BMI: BMI is a measure of nutritional status in adults. It is defined as a person's weight in kilograms divided by the square of that person's height in meters (kg/m2)

Polio: Represents % of coverage of Polio (Pol3) immunization among 1-year-olds.

Diphtheria: Represents % of coverage of Diphtheria tetanus toxoid and pertussis (DTP3) immunization among 1-year-olds.

Incidents_HIV: Incidents of HIV per 1000 population aged 15-49

GDP_per_capita: GDP per capita in current USD

Population_mln: Total population in millions

Thinness_ten_nineteen_years: Prevalence of thinness among adolescents aged 10-19 years. BMI < -2 standard deviations below the median.

Thinness_five_nine_years: Prevalence of thinness among children aged 5-9 years. BMI < -2 standard deviations below the median.

Schooling: Average years that people aged 25+ spent in formal education

Economy_status_Developed: Developed country

Economy_status_Developing: Developing county

2. Implementation of Descriptive Data Analysis

2.1. Visualization

In the following command, the dataset and required libraries have been called in this section.

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

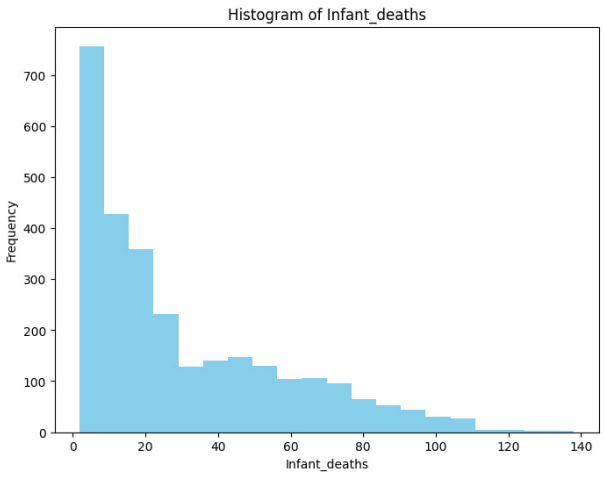
```
file_path = "E:\\LifeExpectancy_DataAnalysis\\primary_dataset.csv"

data = pd.read_csv(file_path)

print(df.head())
```

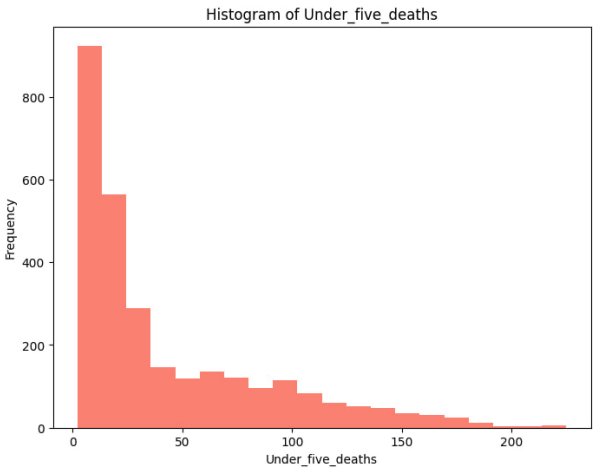
2.1.1. Histogram

In this section, the normality of the data for each feature is examined using a histogram.

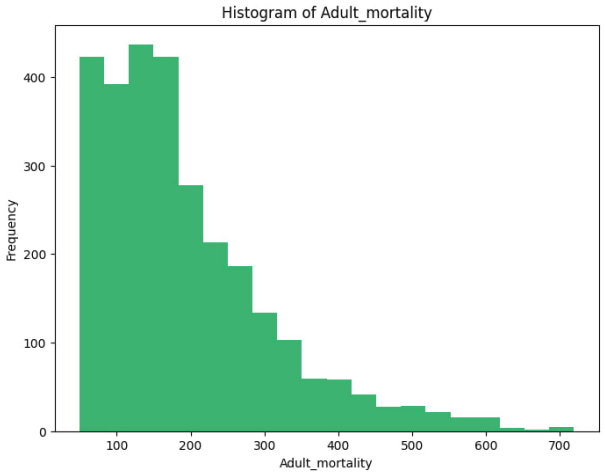
```
#hist
def plot_histogram(data, variables):
    colors = ['skyblue', 'salmon', 'mediumseagreen', 'gold', 'tomato', 'teal', 'violet',
'lightcoral', 'royalblue', 'orange', 'slategray', 'darkorchid', 'indianred', 'forestgreen']

    for i, var in enumerate(variables):
        plt.figure(figsize=(8, 6))
        plt.hist(data[var].dropna(), bins=20, color=colors[i % len(colors)])
        plt.title(f'Histogram of {var}')
        plt.xlabel(var)
        plt.ylabel('Frequency')
        plt.show()
variables = ['Infant_deaths', 'Under_five_deaths', 'Adult_mortality', 'Alcohol_consumption',
'Hepatitis_B', 'Measles',
             'BMI', 'Diphtheria', 'GDP_per_capita', 'Population_mln',
'Thinness_ten_nineteen_years', 'Thinness_five_nine_years', 'Schooling', 'Life_expectancy']

plot_histogram(data, variables)
```

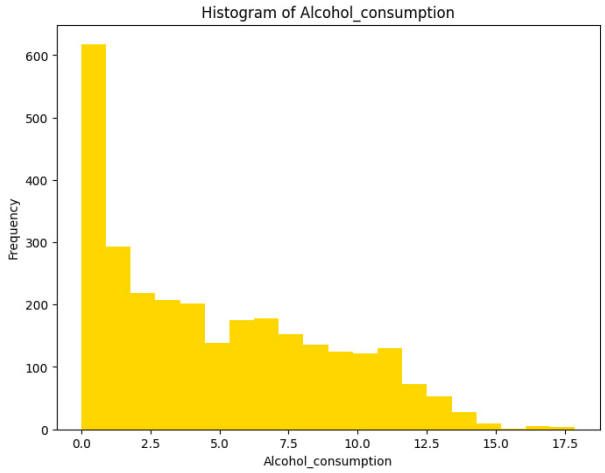
The output of the above code is as follows:

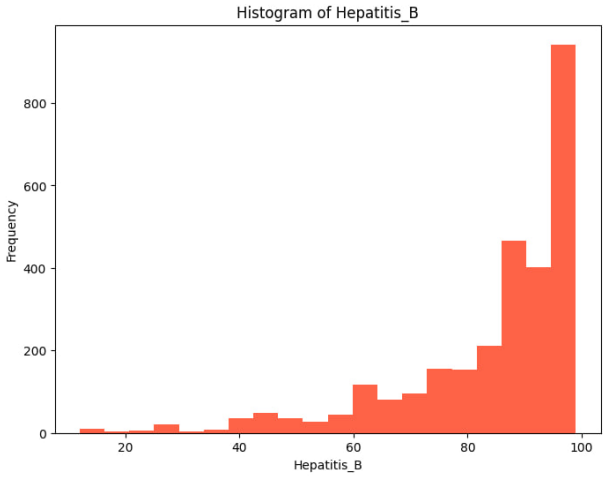


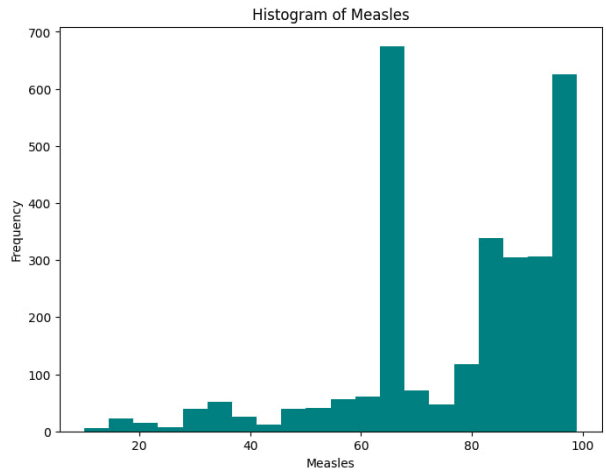


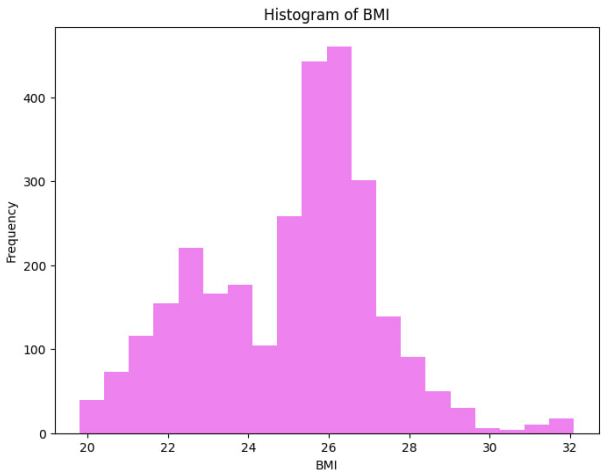
Advanced Programming the Final Project

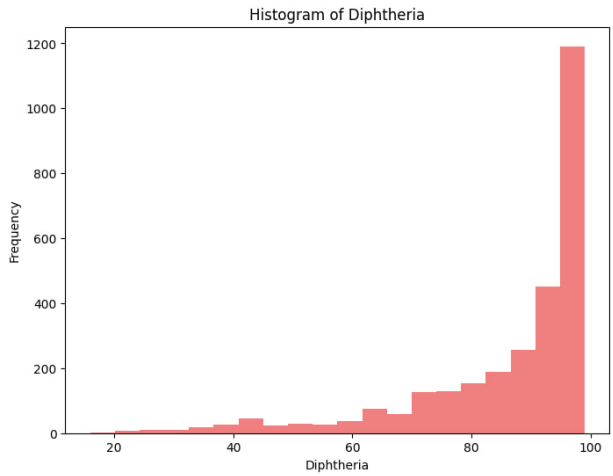




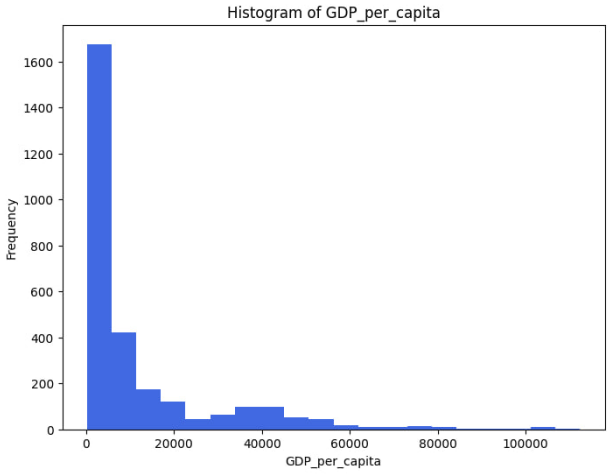


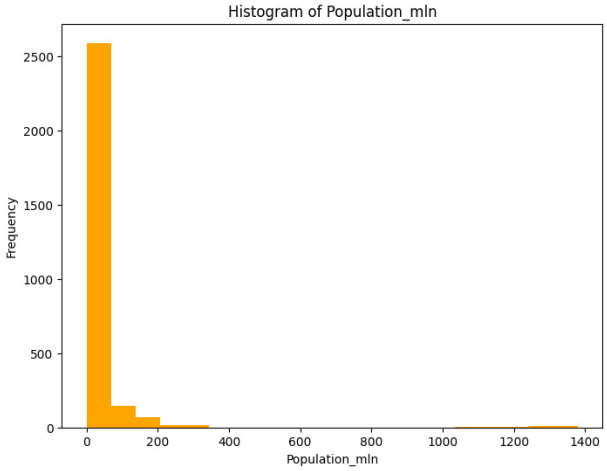


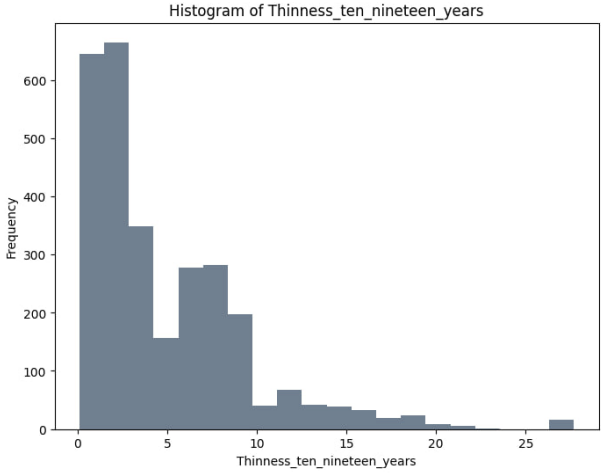


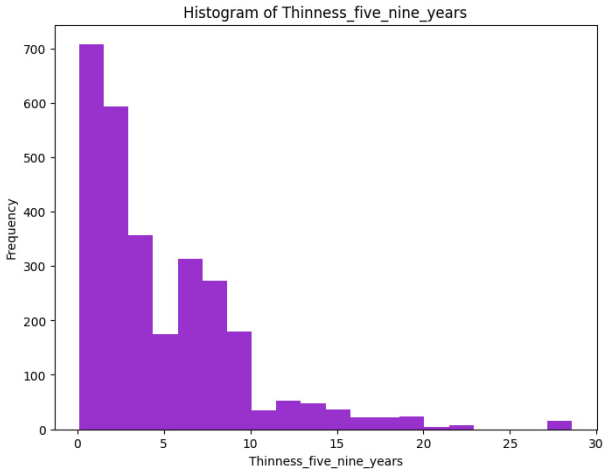


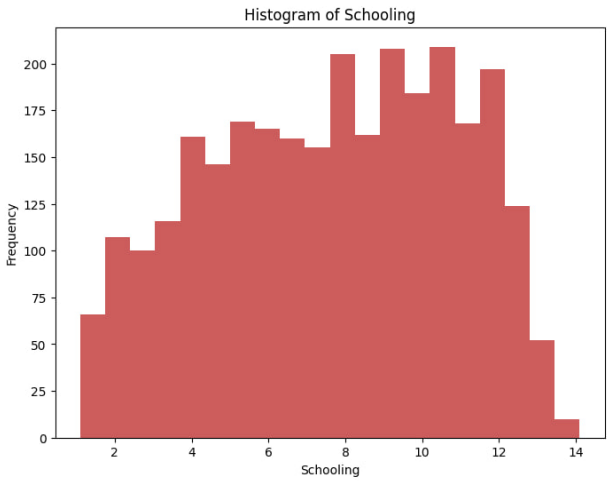
Advanced Programming the Final Project











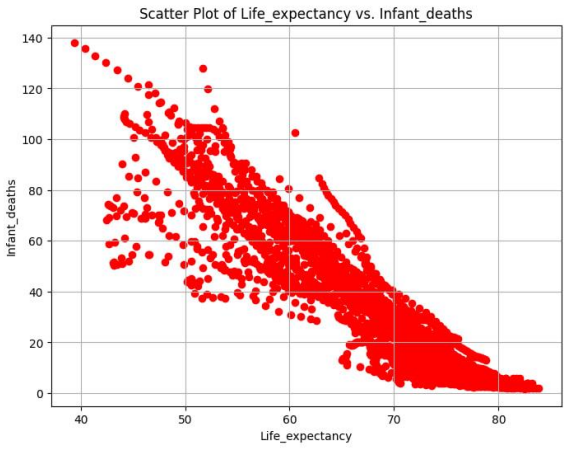
Based on the charts, it is clear that none of these variables follow a normal distribution.

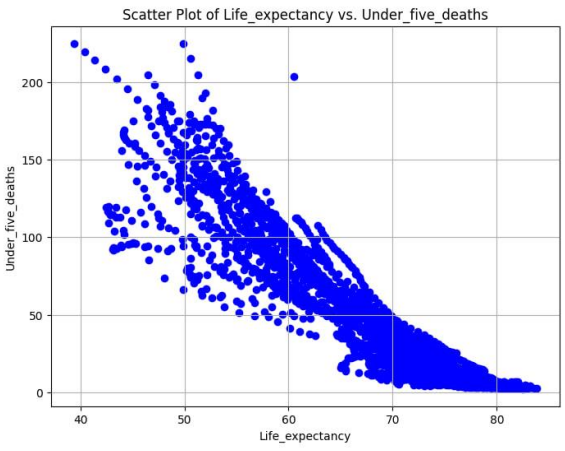
2.1.2. Scatter Plot

In this section, the relationship between key quantitative auxiliary variables and the response variable is observed using a scatter plot.

```
#scatter

def scatter_plot(data, x_variable):
    variables = ['Infant_deaths', 'Under_five_deaths', 'Alcohol_consumption',
'GDP_per_capita', 'Population_mln','Adult_mortality']
    colors = ['red', 'blue', 'green', 'orange', 'purple', 'cyan']

    for i, variable in enumerate(variables):
        plt.figure(figsize=(8, 6))
        plt.scatter(data[x_variable], data[variable], color=colors[i])
        plt.title(f'Scatter Plot of {x_variable} vs. {variable}')
        plt.xlabel(x_variable)
        plt.ylabel(variable)
        plt.grid(True)
        plt.show()

scatter_plot(data, 'Life_expectancy')
```

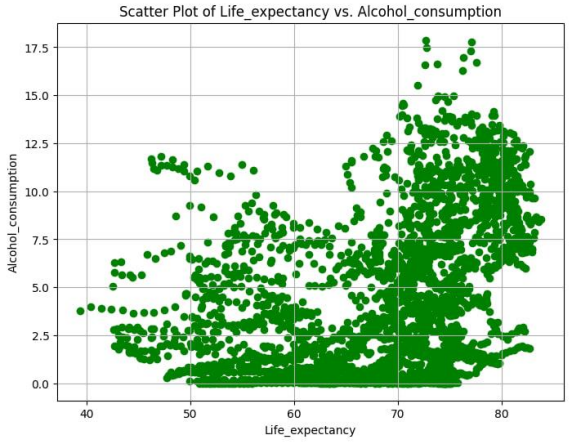
The output of the above code is as follows:

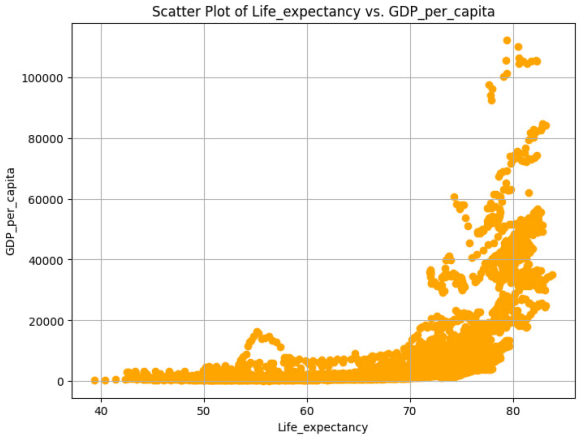




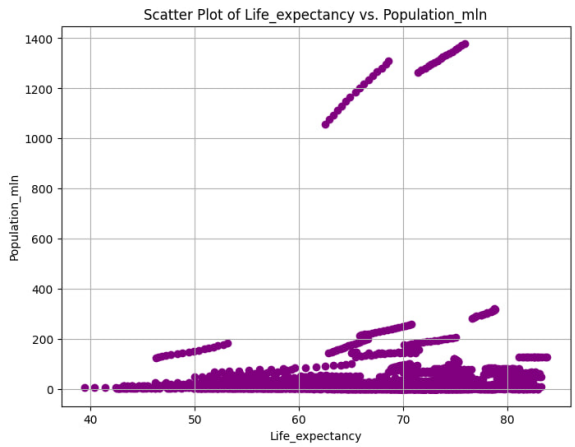
It is evident from the chart that there is an inverse linear relationship between the variable Infant_deaths and the variable Life_expectancy.

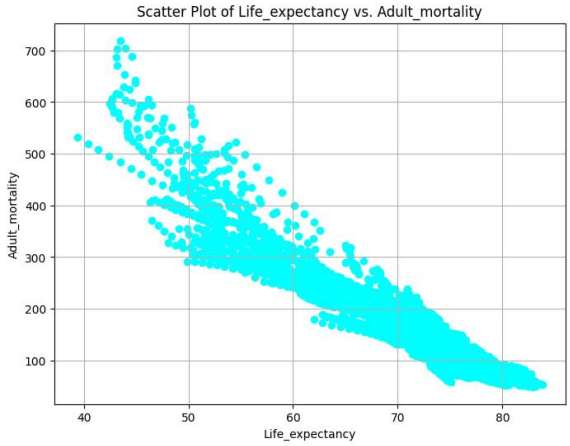
Additionally, there is also an inverse linear relationship between the variable Under_five_deaths and Life_expectancy.





The scatter plots of the variable Alcohol_consumption with Life_expectancy and GDP_per_capita with Life_expectancy exhibit a greater dispersion of data compared to the previous two plots. These variables are related to Life_expectancy, and the correlation coefficient will be used in the upcoming tasks to further investigate this relationship.





There is a pattern between the variable Population_mln and Life_expectancy in the scatter plot, but due to the high population of some countries, there are several data points at the top of this plot that are further away from the rest of the data.

Based on the scatter plot on the right, it is evident that there is an inverse linear relationship between the variables Adult_mortality and Life_expectancy.

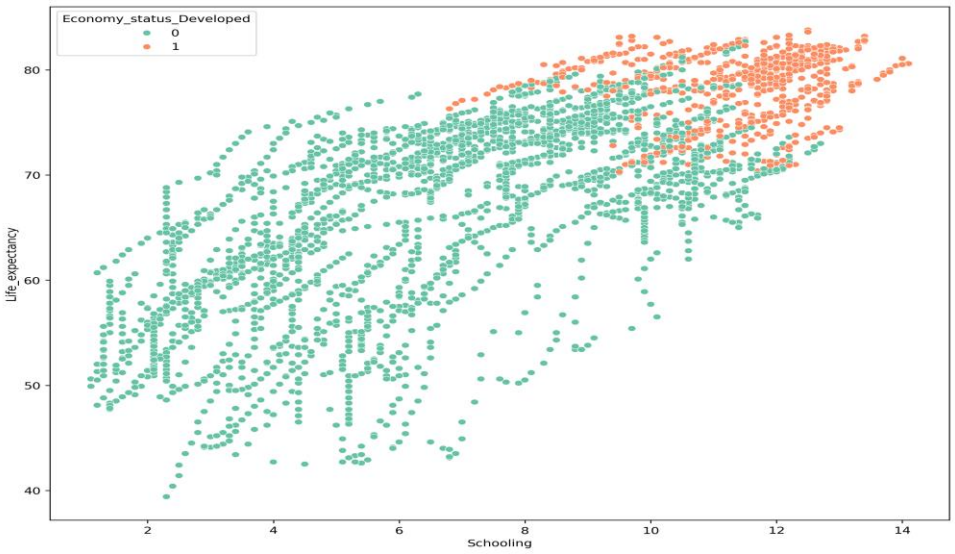
2.1.3. Scatter Plot between Response Variable and Dummy and Nominal Variables

In this section, a scatter plot with three variables has been drawn:
- The variable Life_expectancy is on the vertical axis.
- The variable Schooling is on the horizontal axis.
- The variable Economy_status_Developed is represented by the circles:
  - Green circles with a value of 0 indicate this variable.
  - Orange circles with a value of 1 indicate this variable.

In this section, the relationship between key quantitative auxiliary variables and the response variable is observed using a scatter plot.

```
def scater_3feature(data):
    plt.figure(figsize=(12, 10), dpi=250)
    sns.scatterplot(x=data['Schooling'], y=data['Life_expectancy'],
hue=data['Economy_status_Developed'], palette='Set2')
scater_3feature(data)
```

The output of the above code is as follows:



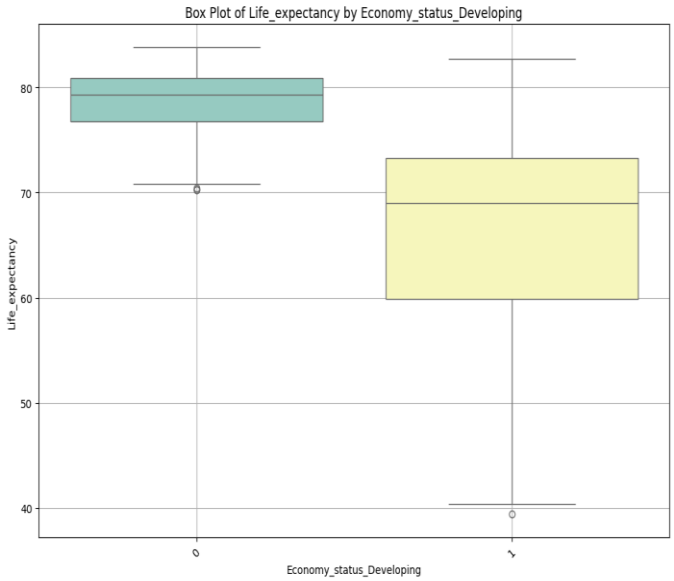
Based on the plot, it can be observed that there is a positive correlation between the variables Life_expectancy (Y) and Schooling (X1). Additionally, there is a positive correlation between the variable Economy_status_Developed(X2) and the other two variables because as the values of Life_expectancy (Y) and Schooling (X1) increase, the circles transition from green to orange.

2.1.4. Box Plot

In this section, the relationship between key qualitative auxiliary variables and the response variable is observed using a box plot.

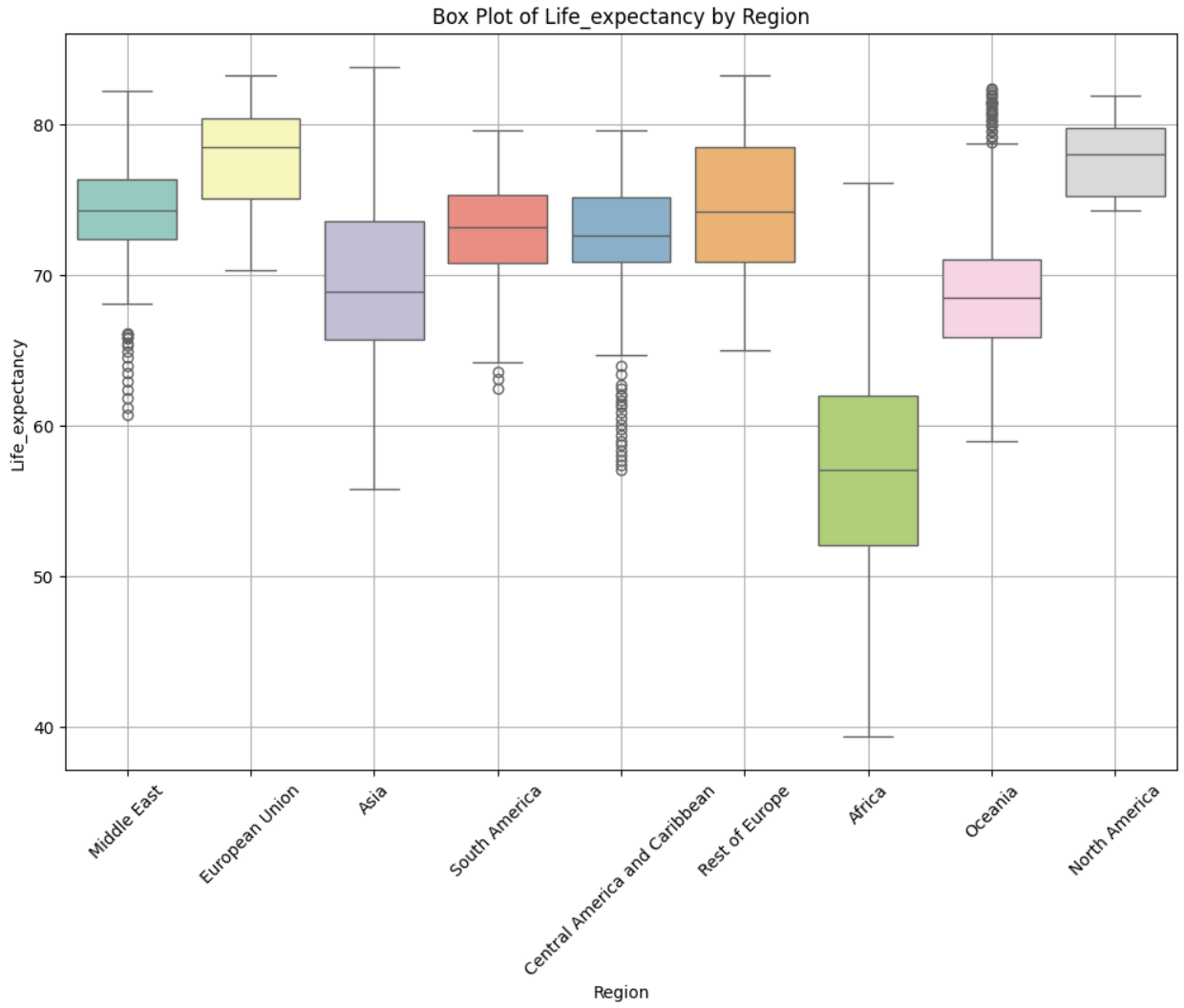
```
#box_plot
def box_plot(data, x_variable, y_variable):
    plt.figure(figsize=(12, 8))
    sns.boxplot(x=x_variable, y=y_variable, data=data, palette='Set3')
    plt.title(f'Box Plot of {y_variable} by {x_variable}')
    plt.xlabel(x_variable)
    plt.ylabel(y_variable)
    plt.xticks(rotation=45)
    plt.grid(True)
    plt.show()


X_variables = ['Economy_status_Developing', 'Region']

for var in X_variables:
    box_plot(data, var, 'Life_expectancy')
```

The output of the above code is as follows:



Based on the plot, it is evident that there is an inverse relationship between the variable Economy_status_Developing. Developed countries have higher life expectancy rates.Furthermore, there are several countries that do not follow this interpretation. (Outliers) As shown in the plot, there is a developed country with a very low life expectancy rate and a developing country with a very low life expectancy as well.

Another point to note is that in developing countries, there is much more data dispersion shown in the plot.



As observable, the purpose of this plot is to examine the relationship between the variable Region and the variable Life_expectancy.

It can be seen that countries in the European Union have the highest life expectancy rates. The lowest life expectancy rates are found in the African region.

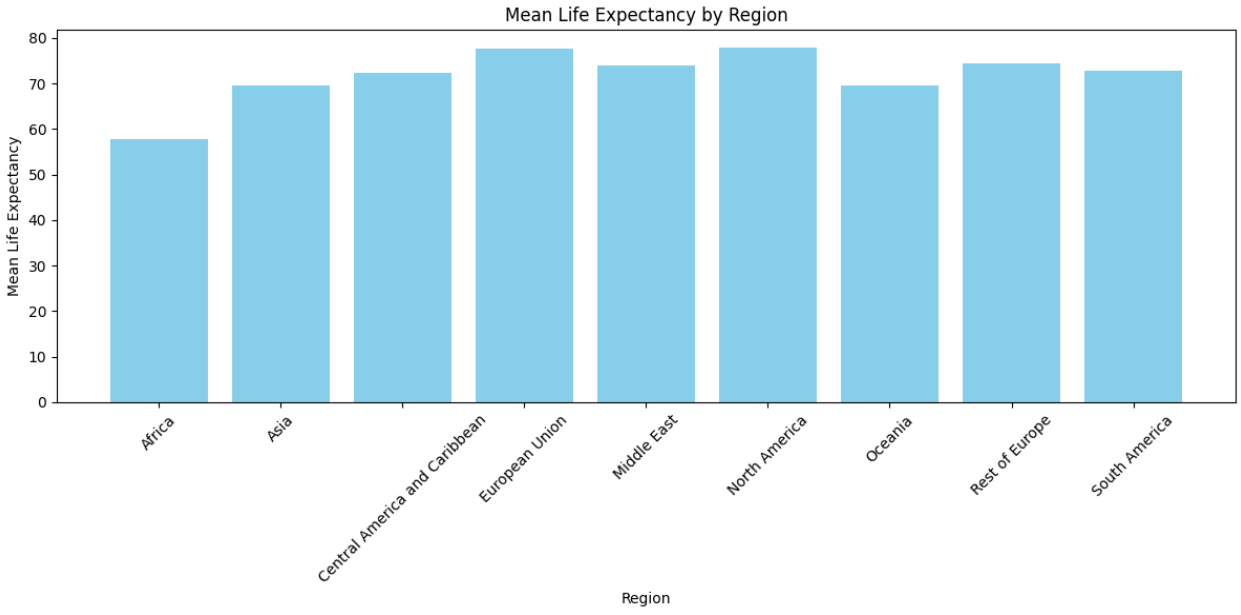
In Central America, there is the highest amount of outlier data points.

2.1.5. Bar Chart

In this section, the average life expectancy in each region is compared with other regions.

```
data_grouped_by_region =
data.groupby('Region')['Life_expectancy'].mean().reset_index()
data_grouped_by_region = data_grouped_by_region.rename(columns={'Life_Expectancy':
'mean_life_expectancy'})
data_grouped_by_region.head()
def plot_bar_chart(data):
    plt.figure(figsize=(12, 6))
    plt.bar(data['Region'], data['Life_expectancy'], color='skyblue')
    plt.xlabel('Region')
    plt.ylabel('Mean Life Expectancy')
    plt.title('Mean Life Expectancy by Region')
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()

plot_bar_chart(data_grouped_by_region)
```

The output of the above code is as follows:



Based on the observable output, it is clear that regions such as the European Union and North America have the highest average life expectancy rates over these 16 years.

Furthermore, the average life expectancy in Africa remains the lowest as indicated by the data.

2.2. Numerical Analysis

2.2.1. describe data (mean,min,...)

In this section, key statistical indicators such as mean, standard deviation, median, etc. are calculated for each feature.

```
#describe_data
def describe_data(data):
    return data.describe()

describe_data(data)
```

The output of the above code is as follows:

	Year	Infant_deaths	Under_five_deaths	Adult_mortality	Alcohol_consumption	Hepatitis_B	Measles	BMI	Polio	Diphtheria
count	2864.000000	2864.000000	2864.000000	2864.000000	2864.000000	2864.000000	2864.000000	2864.000000	2864.000000	2864.000000
mean	2007.500000	30.363792	42.938268	192.251775	4.820882	84.292598	77.344972	25.032926	86.499651	86.271648
std	4.610577	27.538117	44.569974	114.910281	3.981949	15.995511	18.659693	2.193905	15.080365	15.534225
min	2000.000000	1.800000	2.300000	49.384000	0.000000	12.000000	10.000000	19.800000	8.000000	16.000000
25%	2003.750000	8.100000	9.675000	106.910250	1.200000	78.000000	64.000000	23.200000	81.000000	81.000000
50%	2007.500000	19.600000	23.100000	163.841500	4.020000	89.000000	83.000000	25.500000	93.000000	93.000000
75%	2011.250000	47.350000	66.000000	246.791375	7.777500	96.000000	93.000000	26.400000	97.000000	97.000000
max	2015.000000	138.100000	224.900000	719.360500	17.870000	99.000000	99.000000	32.100000	99.000000	99.000000

Incidents_HIV	GDP_per_capita	Population_mln	Thinness_ten_nineteen_years	Thinness_five_nine_years	Schooling	Economy_status_Developed	Economy_status_Developing	Life_expectancy
2864.000000	2864.000000	2864.000000	2864.000000	2864.000000	2864.000000	2864.000000	2864.000000	2864.000000
0.894288	11540.924930	36.675915	4.865852	4.899825	7.632123	0.206704	0.793296	68.856075
2.381389	16934.788931	136.485867	4.438234	4.525217	3.171556	0.405012	0.405012	9.405608
0.010000	148.000000	0.080000	0.100000	0.100000	1.100000	0.000000	0.000000	39.400000
0.080000	1415.750000	2.097500	1.600000	1.600000	5.100000	0.000000	1.000000	62.700000
0.150000	4217.000000	7.850000	3.300000	3.400000	7.800000	0.000000	1.000000	71.400000
0.460000	12557.000000	23.687500	7.200000	7.300000	10.300000	0.000000	1.000000	75.400000
21.680000	112418.000000	1379.860000	27.700000	28.600000	14.100000	1.000000	1.000000	83.800000

2.2.2. Check normality

In this section, the `shapiro` module from the `scipy` library is called to perform the Shapiro-Wilk test.

Then, if the P-Value after conducting the test is less than 0.05, it indicates that the data for that feature is not normally distributed.

```
#check_normality
from scipy.stats import Shapiro
```

```
def check_normality(data):
    variables = ['Infant_deaths','Year', 'Under_five_deaths', 'Adult_mortality',
'Alcohol_consumption', 'Hepatitis_B',
                 'Measles', 'BMI', 'Polio', 'Diphtheria', 'Incidents_HIV',
'GDP_per_capita', 'Population_mln',
                 'Thinness_ten_nineteen_years', 'Thinness_five_nine_years',
'Schooling', 'Life_expectancy']
    for var in variables:
        stat, p = shapiro(data[var])
        print(f'Variable: {var}')
        print(f'Statistic: {stat}, p-value: {p}')

        if p > 0.05:
            print('Normal distribution: Yes\n')
        else:
            print('Normal distribution: No\n')
check_normality(data)
```

The output of the above code is as follows:

```
Variable: Infant_deaths
Statistic: 0.8646653588385218, p-value: 2.289380707152347e-44
Normal distribution: No

Variable: Year
Statistic: 0.9465555306785974, p-value: 4.687800820545888e-31
Normal distribution: No

Variable: Under_five_deaths
Statistic: 0.8157073852291188, p-value: 2.443337410169196e-49
Normal distribution: No

Variable: Adult_mortality
Statistic: 0.8846410706629056, p-value: 6.665827909514486e-42
Normal distribution: No

Variable: Alcohol_consumption
Statistic: 0.9235805994966695, p-value: 6.926800062767792e-36
Normal distribution: No

Variable: Hepatitis_B
Statistic: 0.8141723470446879, p-value: 1.7809443058236373e-49
Normal distribution: No

Variable: Measles
...
Variable: Life_expectancy
Statistic: 0.9349650108561698, p-value: 1.1611718329512556e-33
Normal distribution: No
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

Based on the Shapiro-Wilk test results, the data for the following features are not normally distributed:

- Infant_deaths – Year - Under_five_deaths - Adult_mortality - Alcohol_consumption

- Hepatitis_B- Measles- BMI- Polio- Diphtheria- Incidents_HIV- GDP_per_capita

- Population_mln- Thinness_ten_nineteen_years- Thinness_five_nine_years- Schooling

- Life_expectancy

3. Data Preprocessing

3.1. Cleaning Data & Handling Missing Values

The steps for this task have previously been performed by a data analyst, and the link to this updated dataset has been uploaded on Kaggle. This updated dataset has also been used in this project.

The data had some missing values. A few **strategies for filling in missing** values were applied.

1. Filling data with the **closest three-year average**. If a specific country had a missing value in any year, the data was filled with the closest three-year average.
2. Filling data with the **average of the Region**. If a specific country was missing values for all years, the data was filled with the average of the Region (e.g. Asia, Africa, European Union, etc.)

Data is adjusted and the missing values are filled. Countries that were missing more than 4 data columns were omitted from the database. Examples of these countries are Sudan, South Sudan, and North Korea.

```
file_path = "D:\\Life Expectancy Data.csv"

df1 = pd.read_csv(file_path)

missing_count = df1.isna().sum()
print(missing_count)
```

✓ 0.0s

```
Country                             0
Year                                0
Status                              0
Life expectancy                    10
Adult Mortality                    10
infant deaths                       0
Alcohol                           194
percentage expenditure              0
Hepatitis B                       553
Measles                             0
 BMI                               34
under-five deaths                   0
Polio                              19
Total expenditure                 226
Diphtheria                         19
 HIV/AIDS                           0
GDP                               448
Population                        652
 thinness  1-19 years              34
 thinness 5-9 years                34
Income composition of resources   167
Schooling                         163
dtype: int64
```

```
file_path = "E:\\LifeExpectancy_DataAnalysis\\primary_dataset.csv"

df2 = pd.read_csv(file_path)

missing_count2 = df2.isna().sum()
print(missing_count2)
```

✓ 0.0s

```
Country                     0
Region                      0
Year                        0
Infant_deaths               0
Under_five_deaths           0
Adult_mortality             0
Alcohol_consumption         0
Hepatitis_B                 0
Measles                     0
BMI                         0
Polio                       0
Diphtheria                  0
Incidents_HIV               0
GDP_per_capita              0
Population_mln              0
Thinness_ten_nineteen_years 0
Thinness_five_nine_years    0
Schooling                   0
Economy_status_Developed    0
Economy_status_Developing   0
Life_expectancy             0
dtype: int64
```

It is clear that there are no missing data in the second dataset, and this issue has been handled well.

```
file_path = "D:\\Life Expectancy Data.csv"

df1 = pd.read_csv(file_path)

print(df1.head())
```

✓ 0.0s

```
      Country  Year  Status  Life expectancy   Adult Mortality  \
0  Afghanistan  2015       0              65.0            263.0
1  Afghanistan  2014       0              59.9            271.0
2  Afghanistan  2013       0              59.9            268.0
3  Afghanistan  2012       0              59.5            272.0
4  Afghanistan  2011       0              59.2            275.0

   infant deaths  Alcohol  percentage expenditure  Hepatitis B  Measles   ...  \
0             62     0.01               71.279624         65.0      1154  ...
1             64     0.01               73.523582         62.0       492  ...
2             66     0.01               73.219243         64.0       430  ...
3             69     0.01               78.184215         67.0      2787  ...
4             71     0.01                7.097109         68.0      3013  ...

   Polio  Total expenditure  Diphtheria    HIV/AIDS         GDP  Population  \
0    6.0               8.16         65.0        0.1  584.259210  33736494.0
1   58.0               8.18         62.0        0.1  612.696514    327582.0
2   62.0               8.13         64.0        0.1  631.744976  31731688.0
3   67.0               8.52         67.0        0.1  669.959000   3696958.0
4   68.0               7.87         68.0        0.1   63.537231   2978599.0

    thinness  1-19 years   thinness 5-9 years  \
0                   17.2                 17.3
1                   17.5                 17.5
2                   17.7                 17.7
...
3                              0.463        9.8
4                              0.454        9.5

[5 rows x 22 columns]
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

```
file_path = "E:\\LifeExpectancy_DataAnalysis\\primary_dataset.csv"

df2 = pd.read_csv(file_path)

print(df2.head())
```

✓ 0.0s

```
   Country          Region  Year  Infant_deaths  Under_five_deaths  \
0  Turkiye     Middle East  2015           11.1               13.0
1    Spain  European Union  2015            2.7                3.3
2    India            Asia  2007           51.5               67.9
3   Guyana   South America  2006           32.8               40.5
4   Israel     Middle East  2012            3.4                4.3

   Adult_mortality  Alcohol_consumption  Hepatitis_B  Measles   BMI  ...  \
0         105.8240                 1.32           97       65  27.8  ...
1          57.9025                10.35           97       94  26.0  ...
2         201.0765                 1.57           60       35  21.2  ...
3         222.1965                 5.68           93       74  25.3  ...
4          57.9510                 2.89           97       89  27.0  ...

   Diphtheria  Incidents_HIV  GDP_per_capita  Population_mln  \
0          97           0.08           11006           78.53
1          97           0.09           25742           46.44
2          64           0.13            1076         1183.21
3          93           0.79            4146            0.75
4          94           0.08           33995            7.91

   Thinness_ten_nineteen_years  Thinness_five_nine_years  Schooling  \
0                          4.9                       4.8        7.8
1                          0.6                       0.5        9.7
2                         27.1                      28.0        5.0
...
3                            0                         1       67.0
4                            1                         0       81.7

[5 rows x 21 columns]
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

As it is evident, a column named Region has been added to the data, indicating the region of each country. It can be observed that some countries have been deleted due to insufficient data availability.

3.2. Transforming Variables

The first dataset had a feature called Status, where 1 represented Developed and 0 represented Developing.

However, in the second dataset, this feature has been transformed into two features: Economy_status_Developed and Economy_status_Developing.

3.3. Normalizing/Standardizing Data

In task 2, considering that 16 quantitative variables were not normally distributed, these features were standardized using the following command:

```
import pandas as pd

file_path = "E:\\LifeExpectancy_DataAnalysis\\primary_dataset.csv"

df2 = pd.read_csv(file_path)

print(df2.head())

from sklearn.preprocessing import StandardScaler

#unnormal feature
non_normal_features = ['Infant_deaths', 'Under_five_deaths', 'Adult_mortality',
'Alcohol_consumption',
                       'Hepatitis_B', 'Measles', 'BMI', 'Polio', 'Diphtheria',
                       'Incidents_HIV', 'GDP_per_capita', 'Population_mln',
                       'Thinness_ten_nineteen_years', 'Thinness_five_nine_years',
                       'Schooling', 'Life_expectancy']

scaler = StandardScaler()

df2[non_normal_features] = scaler.fit_transform(df2[non_normal_features])
print(df2[non_normal_features])
```

The output of the above code is as follows:

```
      Infant_deaths  Under_five_deaths  Adult_mortality  Alcohol_consumption  \
0         -0.699654          -0.671831        -0.752264            -0.879342
1         -1.004739          -0.889504        -1.169371             1.388788
2          0.767660           0.560155         0.076810            -0.816547
3          0.088482          -0.054716         0.260638             0.215791
4         -0.979315          -0.867064        -1.168949            -0.484993
...             ...                ...              ...                  ...
2859       2.420203           4.083321         0.866673            -1.187787
2860      -0.234763          -0.321759         0.374107             0.436827
2861      -0.459944          -0.315026        -0.499231            -0.819059
2862      -0.815877          -0.741397         0.102361             1.552054
2863      -1.026531          -0.905213        -1.233154             0.507156

      Hepatitis_B   Measles       BMI     Polio  Diphtheria  Incidents_HIV  \
0        0.794574 -0.661700  1.261475  0.696414    0.690747      -0.341998
1        0.794574  0.892723  0.440877  0.696414    0.690747      -0.337798
2       -1.518979 -2.269725 -1.747384 -1.293275   -1.433965      -0.320998
3        0.544460 -0.179293  0.121756  0.364800    0.433206      -0.043800
4        0.794574  0.624719  0.896765  0.497445    0.497592      -0.341998
...           ...       ...       ...       ...         ...            ...
2859    -0.768637 -0.715301 -1.929740 -3.017672   -3.365522      -0.169799
2860     0.794574  1.053525  0.121756  0.630091    0.561977      -0.367198
2861    -1.393922  0.946324 -1.428263  0.696414    0.690747      -0.367198
2862     0.606989  0.946324  0.486466  0.696414    0.561977      -0.354598
2863     0.231818  0.678320  0.486466  0.563768    0.561977      -0.354598
...
2862               -0.353597   1.093622         0.313051
2863               -0.884051   1.062086         1.440235

[2864 rows x 16 columns]
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

4.Hidden Patterns and Correlations

4.1. Correlation Matrices

In this section, the correlation between features has been examined using the correlation command.

```
import pandas as pd

def compute_numeric_correlation(file_path):

    df=pd.read_csv(file_path)

    non_numeric_cols = df.select_dtypes(exclude=['number']).columns

    df_numeric = df.drop(non_numeric_cols, axis=1)

    return df_numeric.corr().head()

file_path = 'primary_dataset.csv'
result = compute_numeric_correlation(file_path)
```

The output of the above code is as follows:

index	Year	Infant_deaths	Under_five_deaths	Adult_mortality	Alcohol_consumption	
Year	1.0	-0.17240169695310262	-0.17639262125603364	-0.15865957805369962	-0.0006105222266137491	0.17
Under_five_deaths	-0.17639262125603364	0.9856513455483952	1.0	0.8023611229154944	-0.409367397100711	-0.
Infant_deaths	-0.17240169695310262	1.0	0.9856513455483952	0.7946608587200148	-0.4545261471607	-0.5
Alcohol_consumption	-0.0006105222266137491	-0.4545261471607	-0.409367397100711	-0.2447937555264703	1.0	0.16
Adult_mortality	-0.15865957805369962	0.7946608587200148	0.8023611229154944	1.0	-0.2447937555264703	-0.34

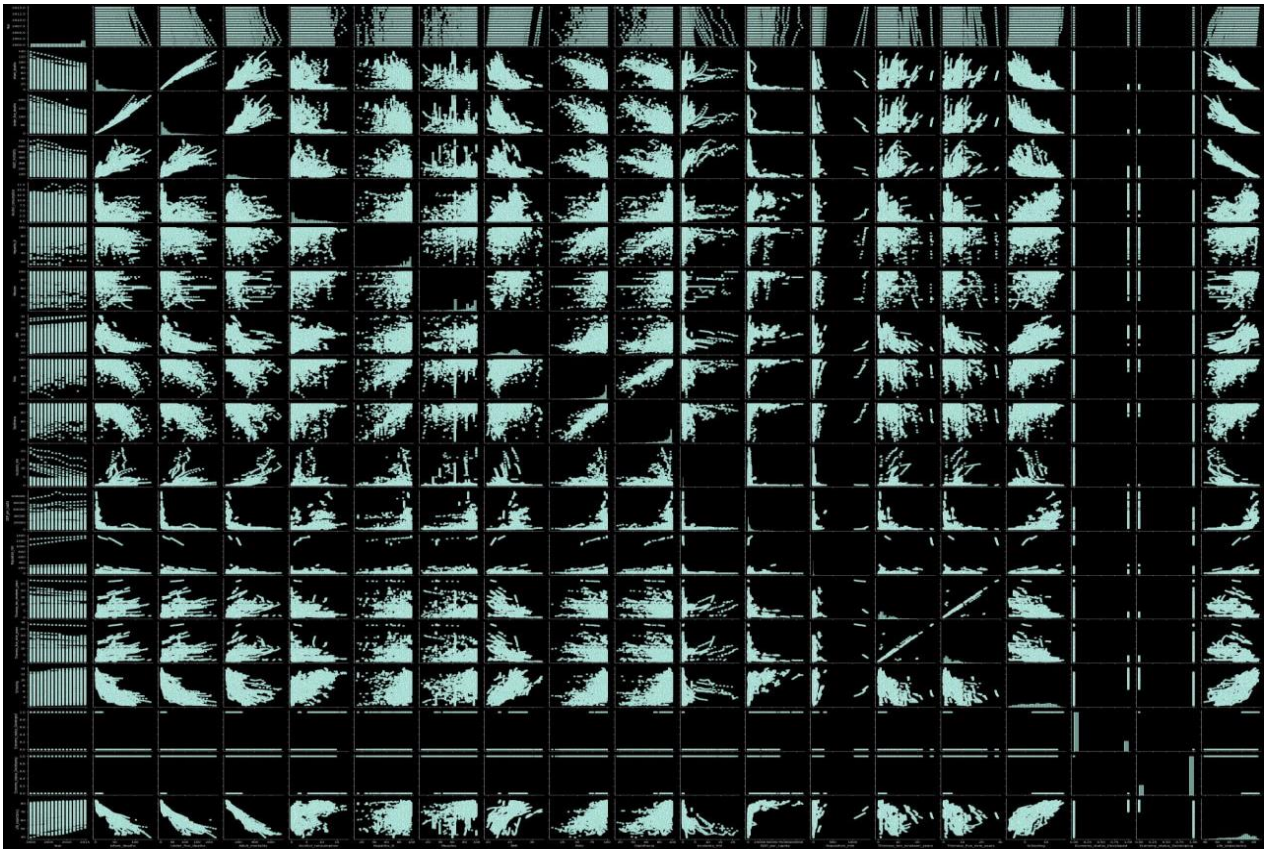
Based on the output, the relationship between features is quite clear. If the correlation coefficient is close to +1, it indicates a very strong and positive correlation between two features. If it is close to -1, it means there is a very strong and negative correlation between two features. If the correlation coefficient is close to 0, it means the two features are not correlated with each other.

4.2. Scatter Plots

To visualize the level of correlation and relationship between features on a plot, a scatter matrix plot is used. If the data shows a linear pattern with a positive slope between two features, there is a strong positive correlation. Conversely, if the data shows a linear pattern with a negative slope, there is a strong negative correlation between the two features.

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

def plot_pairplot(df):
    sns.pairplot(df)
    plt.show()

df = pd.read_csv('/content/primary_dataset.csv')
plot_pairplot(df)
```

The output of the above code is as follows:



Features that follow the definition above exhibit correlation with each other. However, for a better understanding of the correlation between features, it is recommended to use a heatmap plot, which is done in task 5.

4.3. Advanced Statistical Tests

4.3.1. T-test

In this section, a T-test is conducted to compare the means of life satisfaction between developed countries and developing countries. The null hypothesis (H0) is that the means of the two groups are equal, while the alternative hypothesis (H1) is that the means of the two groups are not equal.

```
import pandas as pd
from scipy import stats

def perform_ttest(df, group_col, target_col, group1_value, group2_value, alpha=0.05):

    group1 = df[df[group_col] == group1_value][target_col]
    group2 = df[df[group_col] == group2_value][target_col]

    t_statistic, p_value = stats.ttest_ind(group1, group2)

    print('t-test results:')
    print('----------------------')
    print('Group 1:')
    print('Sample size:', len(group1))
    print('Mean:', group1.mean())
    print('Variance:', group1.var())
    print()
    print('Group 2:')
    print('Sample size:', len(group2))
    print('Mean:', group2.mean())
    print('Variance:', group2.var())
    print()
    print('t-statistic:', t_statistic)
    print('p-value:', p_value)

    print()
    alpha = 0.05
    if p_value < alpha:
        print('Conclusion: The mean life expectancy between the two groups is
statistically significantly different.')
    else:
        print('Conclusion: The mean life expectancy between the two groups is not
statistically significantly different.')

df = pd.read_csv('/content/primary_dataset.csv')
perform_ttest(df, 'Economy_status_Developed', 'Life_expectancy', 1, 0)
```

The output of the above code is as follows:

```
t-test results:
----------------------
Group 1 (Developed countries):
Sample size: 592
Mean: 78.50574324324323
Variance: 9.982843439886588

Group 2 (Developing countries):
Sample size: 2272
Mean: 66.34172535211268
Variance: 78.33049072506373

t-statistic: 32.89510831384994
p-value: 1.4081397446984548e-201

Conclusion: The mean life expectancy between developed and developing countries is statistically significantly different.
```

As observed, the p-value is less than 0.05, so the null hypothesis (H0) of equal means between the two groups (developed countries and developing countries) is rejected.

4.3.2 ANOVA test

In this section, an ANOVA test is conducted to compare the means of life satisfaction across different regions. The null hypothesis (H0) is that the means of all groups are equal, while the alternative hypothesis (H1) is that at least two groups have unequal means.

```
import pandas as pd
import statsmodels.api as sm
from statsmodels.formula.api import ols
def perform_anova(df, dependent_var, independent_var, alpha=0.05):

    dependent_data = df[dependent_var]
    independent_data = df[independent_var]

    data = pd.DataFrame({dependent_var: dependent_data, independent_var:
independent_data})

    model = ols(f'{dependent_var} ~ C({independent_var})', data=data).fit()

    anova_table = sm.stats.anova_lm(model, typ=2)
    print(anova_table)

    p_value = anova_table["PR(>F)"][0]

    if p_value < alpha:
        print("P-value is less than alpha. Reject the null hypothesis that means are
equal.")
    else:
        print("Fail to reject the null hypothesis. Means are not significantly
different.")
df = pd.read_csv('/content/primary_dataset.csv')
perform_anova(df, 'Life_expectancy', 'Region')
```

The output of the above code is as follows:

```
                   sum_sq      df           F  PR(>F)
C(Region)  157346.125052     8.0  585.349859     0.0
Residual    95930.489235  2855.0         NaN     NaN
P-value is less than alpha. Reject the null hypothesis that means are equal.
```

As observed, the p-value is less than 0.05, so the null hypothesis (H0) of equal means of life satisfaction across different regions is rejected.

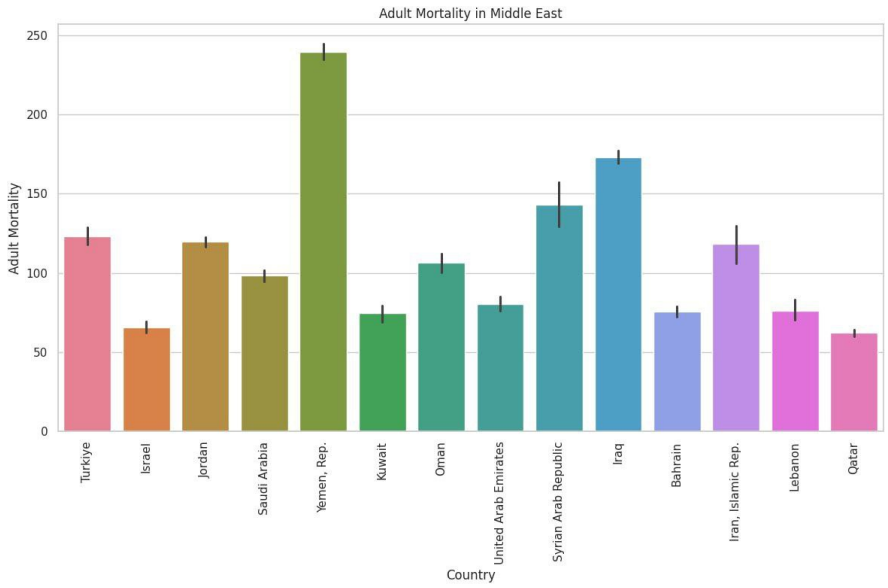
5.Visualization of Findings

5.1. Bar Charts for Middle East

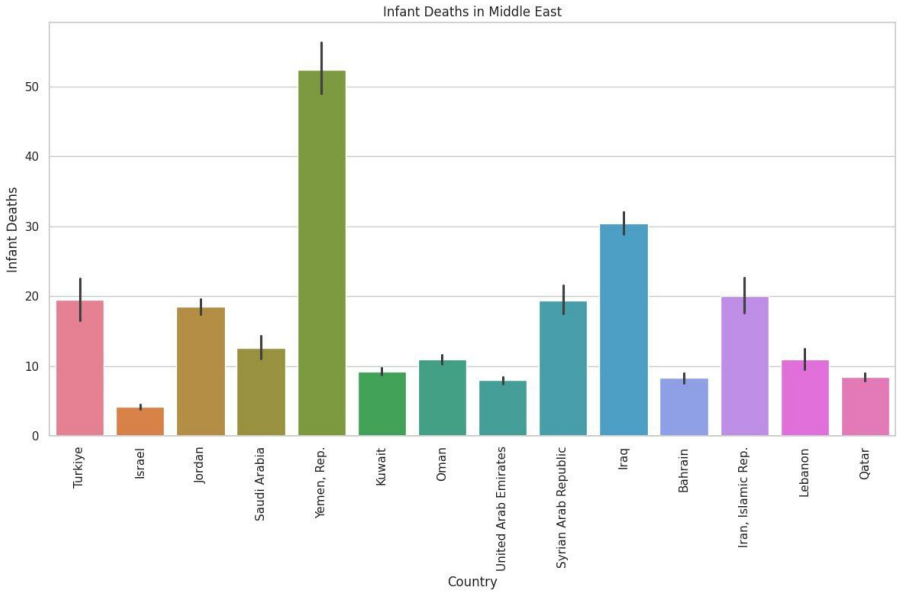
In this section, the research focuses on the Middle East region. Bar charts of 6 features related to Middle Eastern countries are plotted in this section. The vertical axis represents the features, and the horizontal axis represents the Middle Eastern countries. The purpose of these charts is to examine the status of the 6 variables in Middle Eastern countries.

```
def plot_columns_for_region(file_path, region_name='Middle East', region_col='Region',
country_col='Country', columns=None):
    if columns is None:
        columns = ['Adult_mortality', 'Infant_deaths', 'GDP_per_capita',
                   'Population_mln', 'Alcohol_consumption', 'Incidents_HIV']

    data = pd.read_csv(file_path)
    region_data = data[data[region_col] == region_name]
    print(region_data.columns)

    sns.set(style="whitegrid")
    for column in columns:
        if column in region_data.columns:
            plt.figure(figsize=(12, 8))
            sns.barplot(x=country_col, y=column, data=region_data, palette='husl')

            plt.xticks(rotation=90)
            plt.xlabel('Country')
            plt.ylabel(column.replace('_', ' ').title())
            plt.title(f'{column.replace("_", " ").title()} in {region_name}')
            plt.tight_layout()
            plt.show()
        else:
            print(f"Warning: Column '{column}' not found in the dataset for
{region_name}.")

plot_columns_for_region('/content/primary_dataset.csv')
```

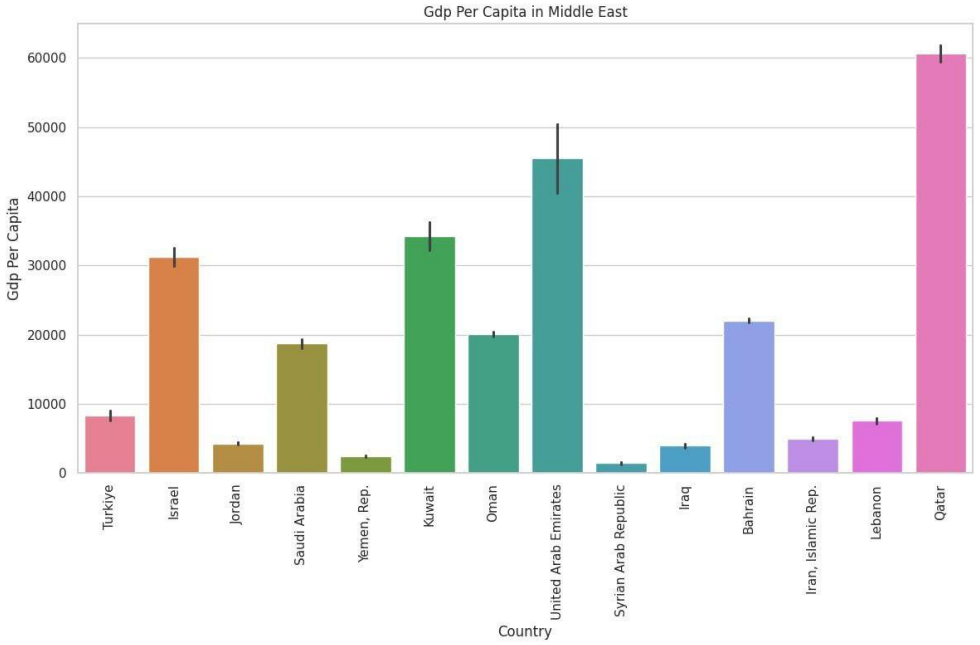
The output of the above code is as follows:



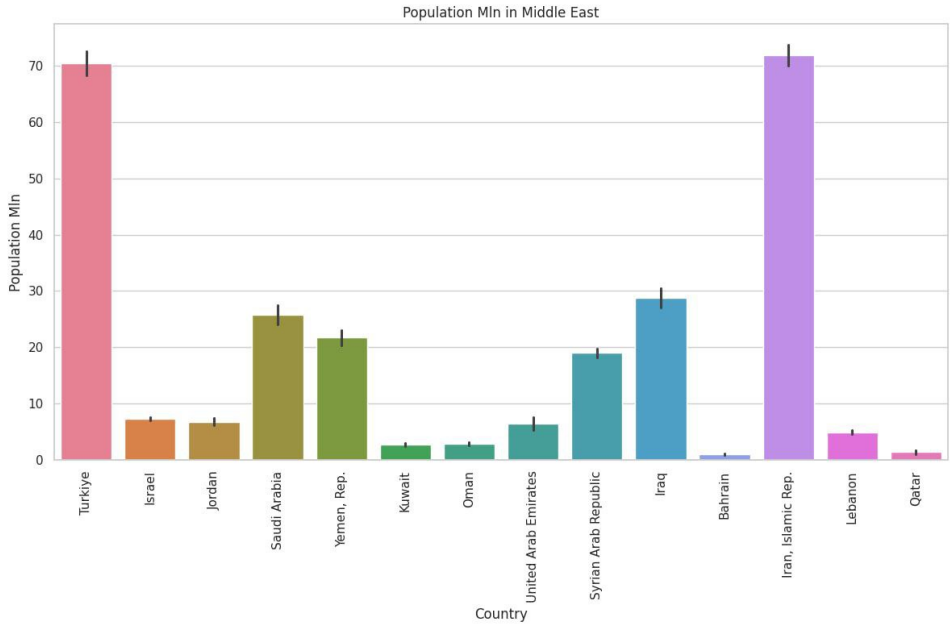
Based on the chart, it is evident that the adult mortality rate is highest in Yemen and lowest in Qatar and Israel.



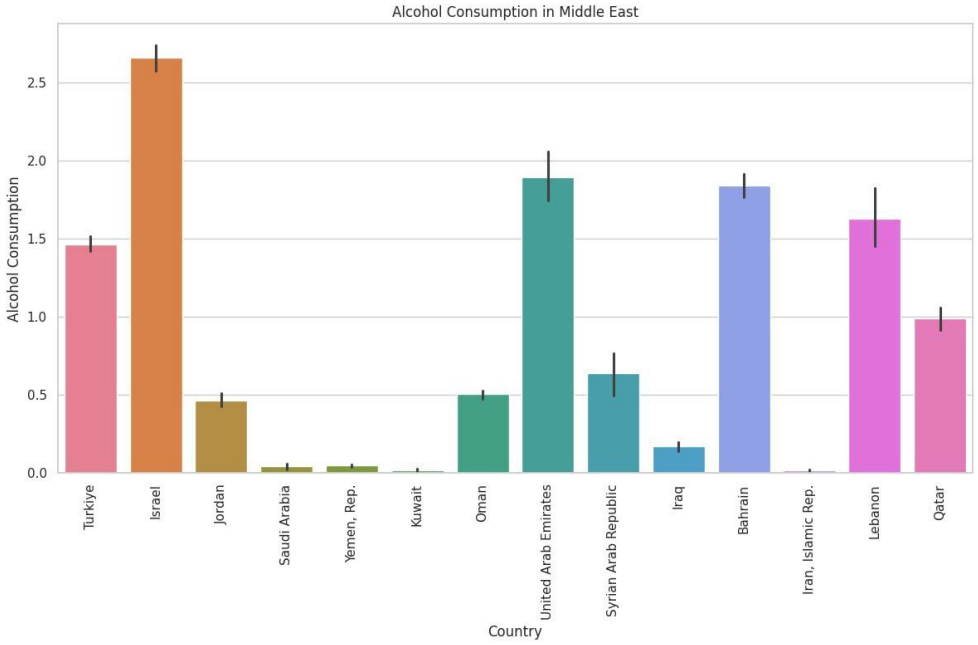
Based on the chart, it is evident that the infant mortality rate is also highest in Yemen and lowest in Israel.



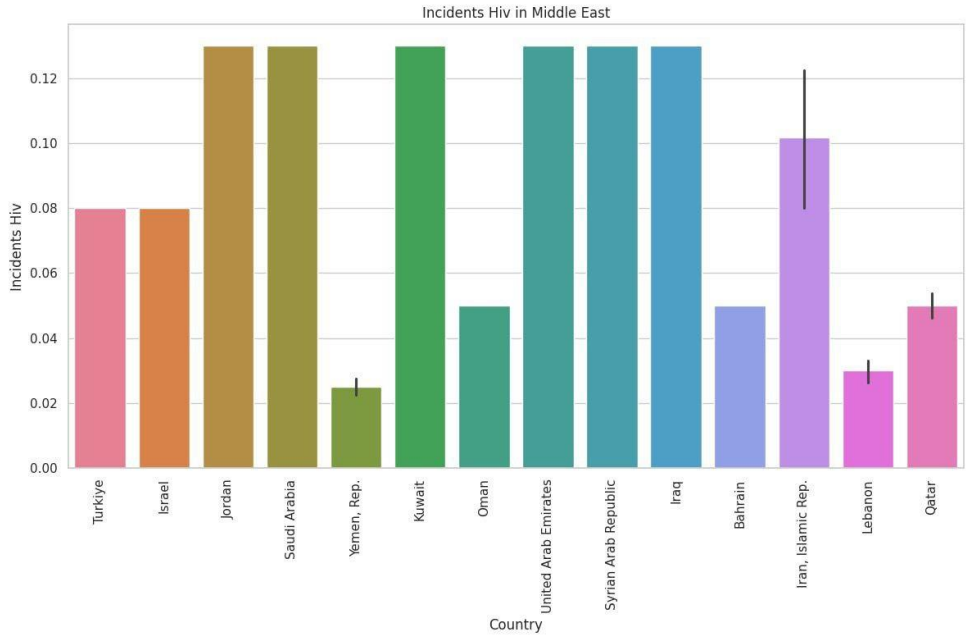
Based on the observed chart, the highest Gross Domestic Product (GDP) rate is attributed to Qatar, while the lowest rate is associated with Syria. An important point is that despite having abundant resources, Iran, unlike its oil and gas-rich neighbors, does not have a high GDP rate.



Based on the chart, Iran and Turkey have the highest populations, while Bahrain has the lowest population.



Based on the chart, Israel has the highest alcohol consumption rate, while Iran has the lowest rate.

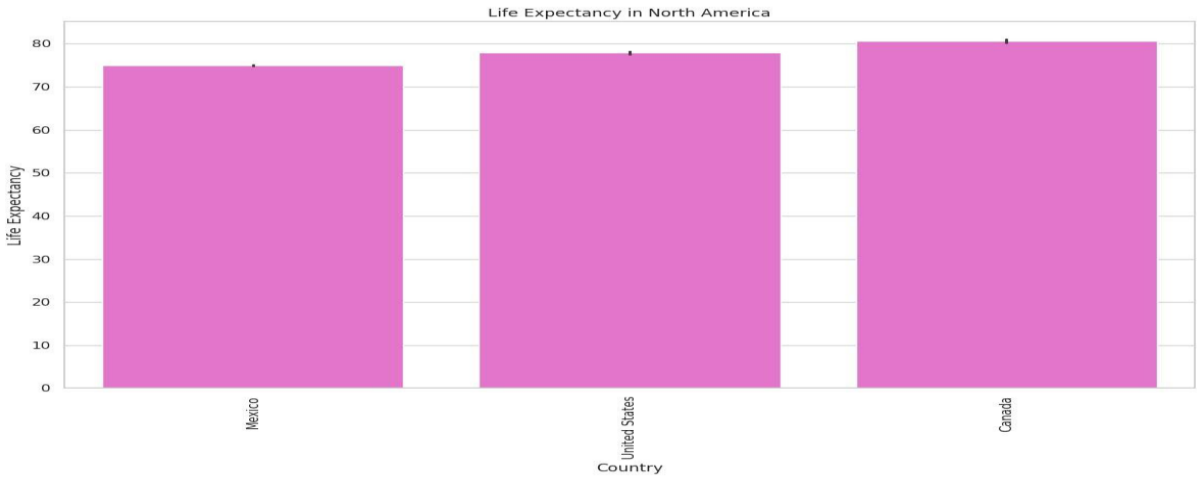


As observed, the HIV incidence per thousand people is high in countries like Jordan, Saudi Arabia, Kuwait, United Arab Emirates, Syria, and Iraq. The lowest incidence is seen in Yemen.

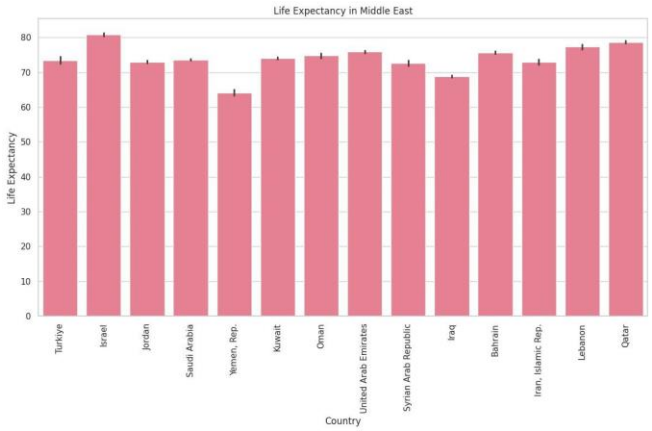
5.2. Bar chart of life expectancy in different regions

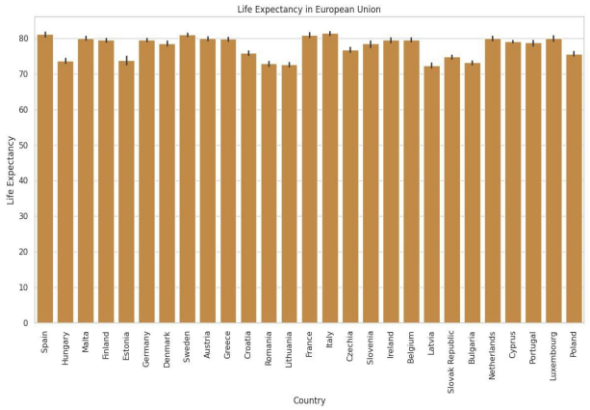
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

def plot_life_expectancy_by_region(file_path, life_expectancy_col='Life_expectancy',
region_col='Region', country_col='Country'):

    data = pd.read_csv(file_path)

    sns.set(style="whitegrid")

    regions = data[region_col].unique()

    colors = sns.color_palette("husl", len(regions))
    for i, region in enumerate(regions):
        plt.figure(figsize=(12, 8))
        region_data = data[data[region_col] == region]
        sns.barplot(x=country_col, y=life_expectancy_col, data=region_data,
palette=[colors[i]])

        plt.xticks(rotation=90)
        plt.xlabel('Country')
        plt.ylabel('Life Expectancy')
        plt.title(f'Life Expectancy in {region}')
        plt.tight_layout()
        plt.show()
```

The output of the above code is as follows:

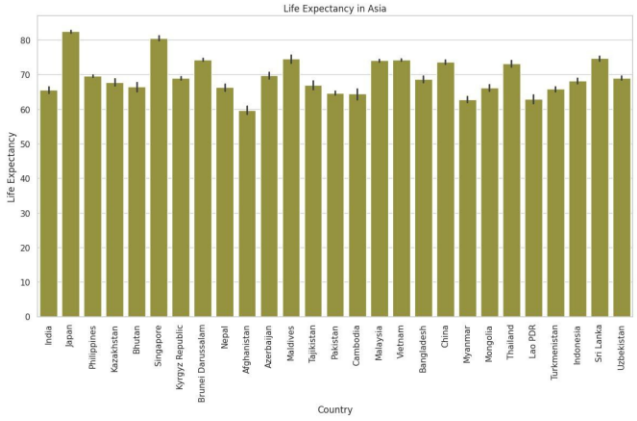


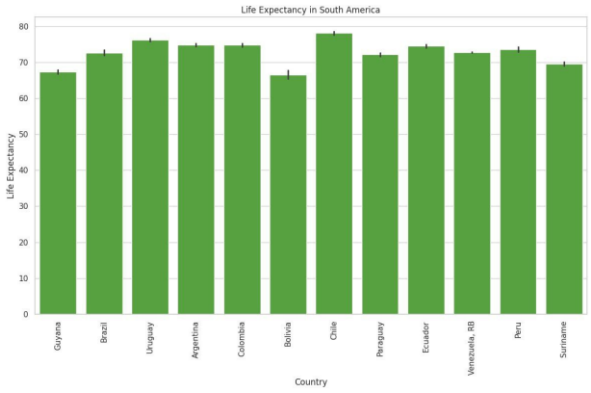
Based on the observed chart, the highest life expectancy rate in the North America region is attributed to Canada.





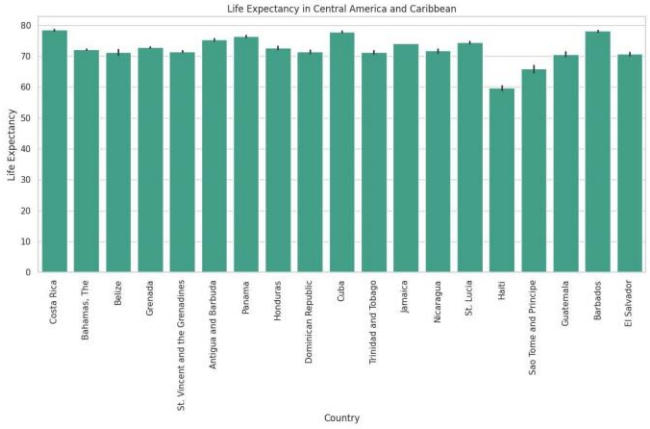
In the Middle East, the highest life expectancy rate is attributed to Israel.
In the European Union countries, the highest rates are observed in Italy and Spain.

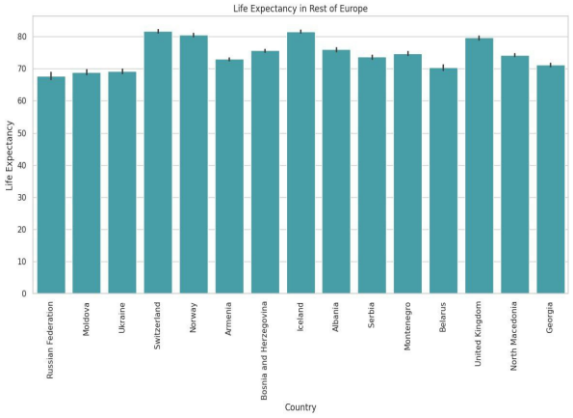




In Asia, the highest life expectancy rate is attributed to Japan, while the lowest rate is in Afghanistan.

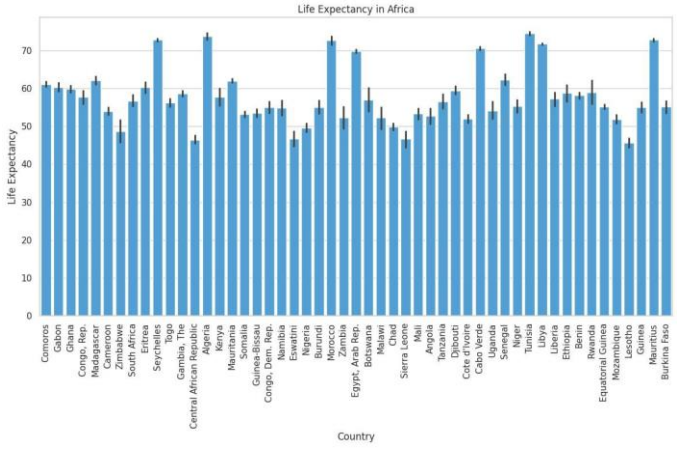
In South America, the highest life expectancy rate is in Chile, while the lowest rate is in Bolivia.

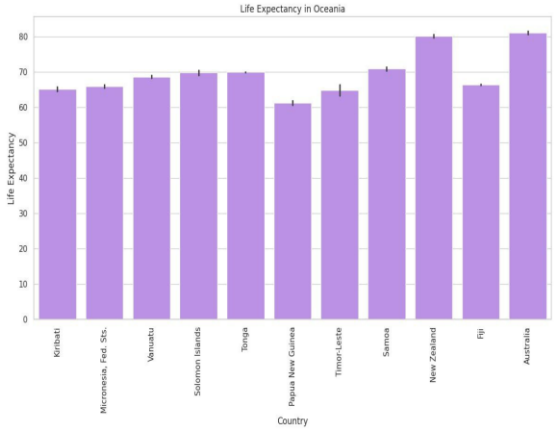




Based on the observed chart, in Central America, Costa Rica has the highest life expectancy rate.

Among non-EU European countries, Switzerland and Iceland have the highest life expectancy rates.





In Africa, Tunisia has the highest life expectancy rate.

In Oceania, Australia has the highest life expectancy rate.

5.3. Line graph

In this section, the average life expectancy based on each year in each region is examined and compared with other regions. The difference from the previous section is that the differences are observable on a yearly basis.

First, the average life expectancy of countries is calculated for each region based on each year.

```
def calculate_mean(data):
    mean_data = data.groupby('Region')['Life_expectancy'].mean().reset_index()
    return mean_data

mean_life_expectancy_by_region = calculate_mean(data)
data_grouped.head(20)
```

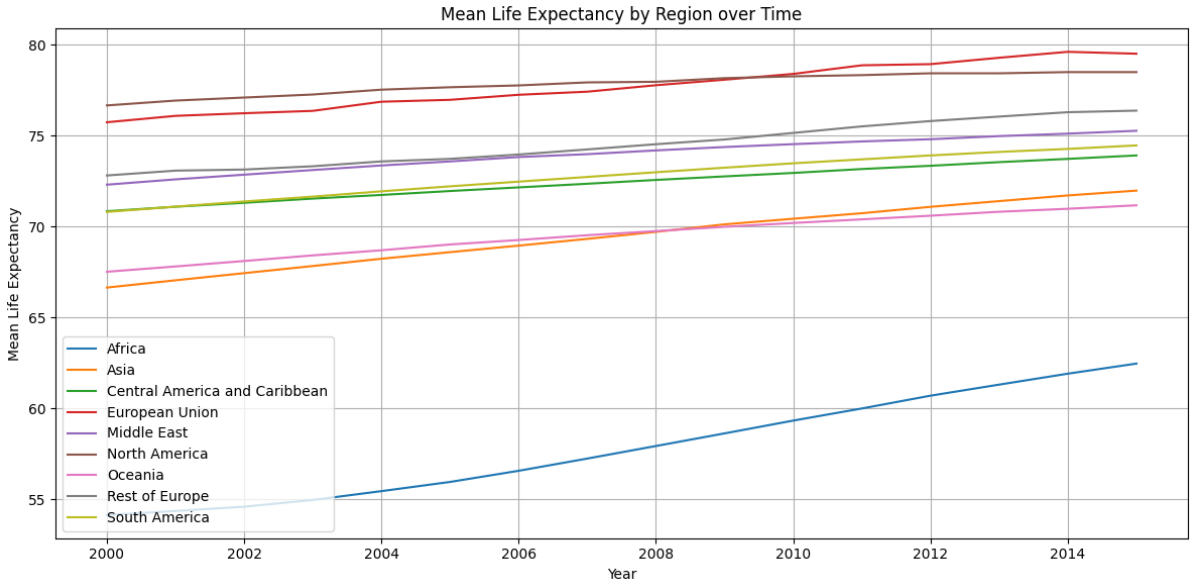
The output of the above code is as follows:

	Region	Year	Life_expectancy
0	Africa	2000	54.137255
1	Africa	2001	54.358824
2	Africa	2002	54.594118
3	Africa	2003	54.964706
4	Africa	2004	55.450980
5	Africa	2005	55.956863
6	Africa	2006	56.570588
7	Africa	2007	57.243137
8	Africa	2008	57.933333
9	Africa	2009	58.627451
10	Africa	2010	59.331373
11	Africa	2011	59.998039
12	Africa	2012	60.701961
13	Africa	2013	61.309804
14	Africa	2014	61.911765
15	Africa	2015	62.466667
16	Asia	2000	66.648148
17	Asia	2001	67.048148
18	Asia	2002	67.440741
19	Asia	2003	67.833333

After calculating the averages, a line chart has been drawn with the following command:

```
#line_chart
def plot_line_chart(data):
    plt.figure(figsize=(12, 6))
    for region in data['Region'].unique():
        region_data = data[data['Region'] == region]
        plt.plot(region_data['Year'], region_data['Life_expectancy'], label=region)

    plt.xlabel('Year')
    plt.ylabel('Mean Life Expectancy')
    plt.title('Mean Life Expectancy by Region over Time')
    plt.legend()
    plt.grid(True)
    plt.tight_layout()
    plt.show()


plot_line_chart(data_grouped)
```

The output of the above code is as follows:



As evident, this chart illustrates the average life expectancy of countries in each region by year.

It is noticeable that the average life expectancy for all regions is increasing each year.

The growth rate of life expectancy in Africa is relatively higher compared to other regions.

Until the year 2010, the average life expectancy in countries in North America was the highest, but after 2010, the average life expectancy in European Union countries has surpassed it.

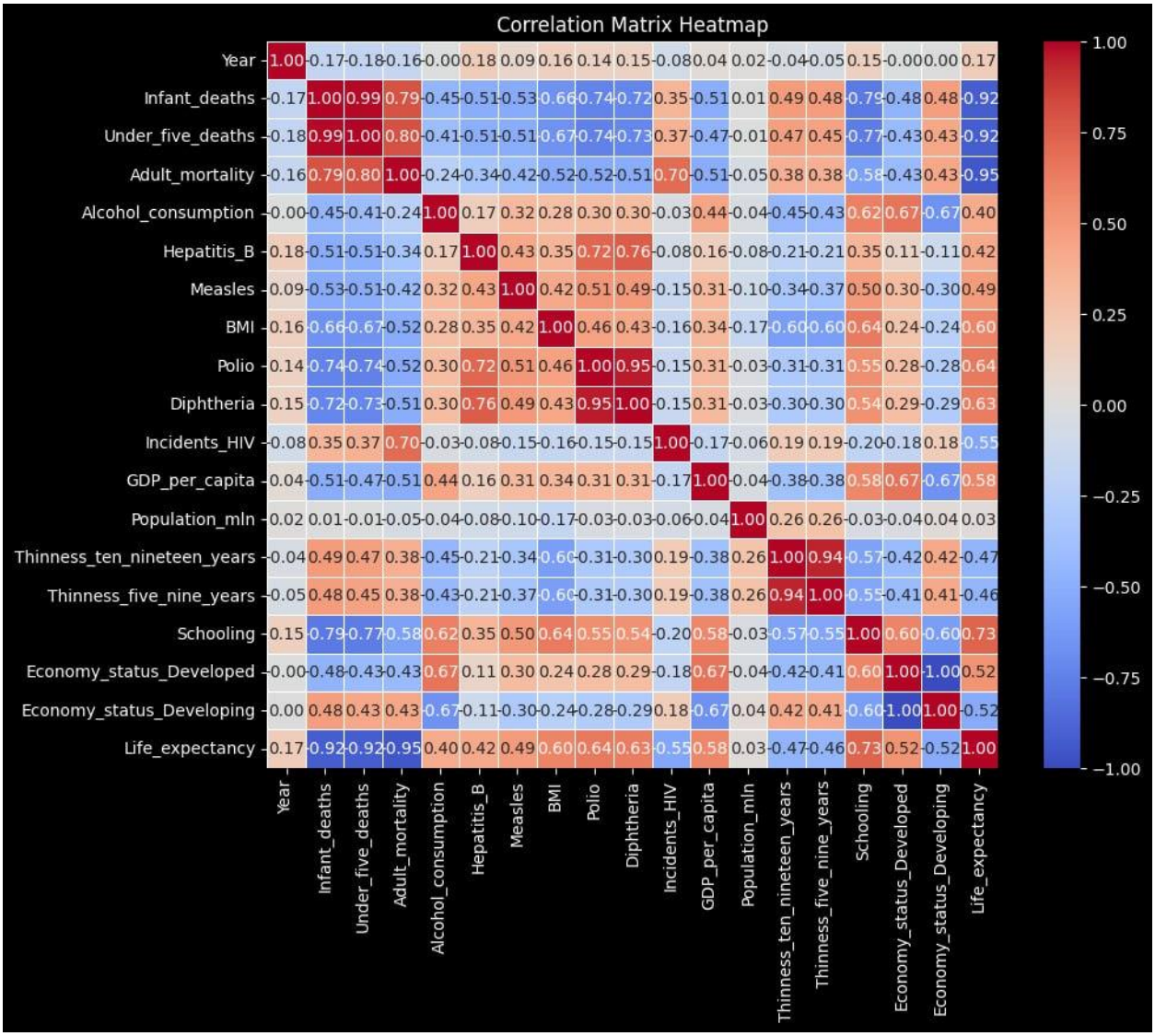
5.4. Heat Maps

Heat maps are used for a better understanding and visualization of correlations between variables.

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

def plot_correlation_heatmap(df_numeric):


    numeric_columns = df_numeric.select_dtypes(include=['number'])
    correlation_matrix = numeric_columns.corr()


    plt.figure(figsize=(10, 8))
    sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f",
linewidths=.5)
    plt.title('Correlation Matrix Heatmap')
    plt.show()


df = pd.read_csv('/content/primary_dataset.csv')
plot_correlation_heatmap(df)
```

If a square is close to blue, it means there is a negative correlation between the two variables. If it is close to red, it means there is a positive correlation between the two variables. If it is close to white, it means there is no correlation between the two variables.

The output of the above code is as follows:



As observed, there is a strong negative correlation between life expectancy and variables like Infant_deaths, Under_five_deaths, and Adult_mortality. Furthermore, there is a strong positive correlation between life expectancy and the variable Schooling.

6. Application of Findings to Decision-Making

In this part of the project, to present analysis and make decisions, we need to know which features are the most effective. In the statistical analysis and statistical tests section, we can provide speculation about this subject. However, to refine our analysis, we used the SHAP module; SHAP is a Python module that determines the relationship between various features and the target feature using mathematical algorithms. For this purpose, we need to use learning algorithms, and here we used random forest and XGBoost.

```
# drop feature that we dont need it

features_drop = ['Country', 'Year', 'Region', 'Life_expectancy',
'Economy_status_Developing']
X = df.drop(columns=features_drop, axis=1)
X.dtypes
```

```
Infant_deaths                 float64
Under_five_deaths             float64
Adult_mortality               float64
Alcohol_consumption           float64
Hepatitis_B                     int64
Measles                         int64
BMI                           float64
Polio                           int64
Diphtheria                      int64
Incidents_HIV                 float64
GDP_per_capita                  int64
Population_mln                float64
Thinness_ten_nineteen_years   float64
Thinness_five_nine_years      float64
Schooling                     float64
Economy_status_Developed        int64
dtype: object
```

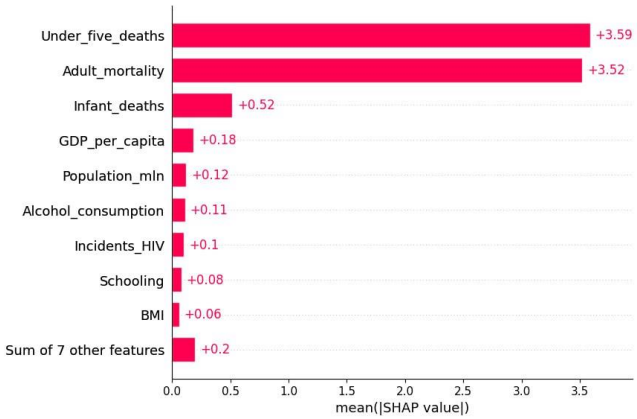
```
    y = df.Life_expectancy
    y.head()
```

```
0    63.4
1    63.0
2    62.5
3    62.1
4    61.6
Name: Life_expectancy, dtype: float64
```

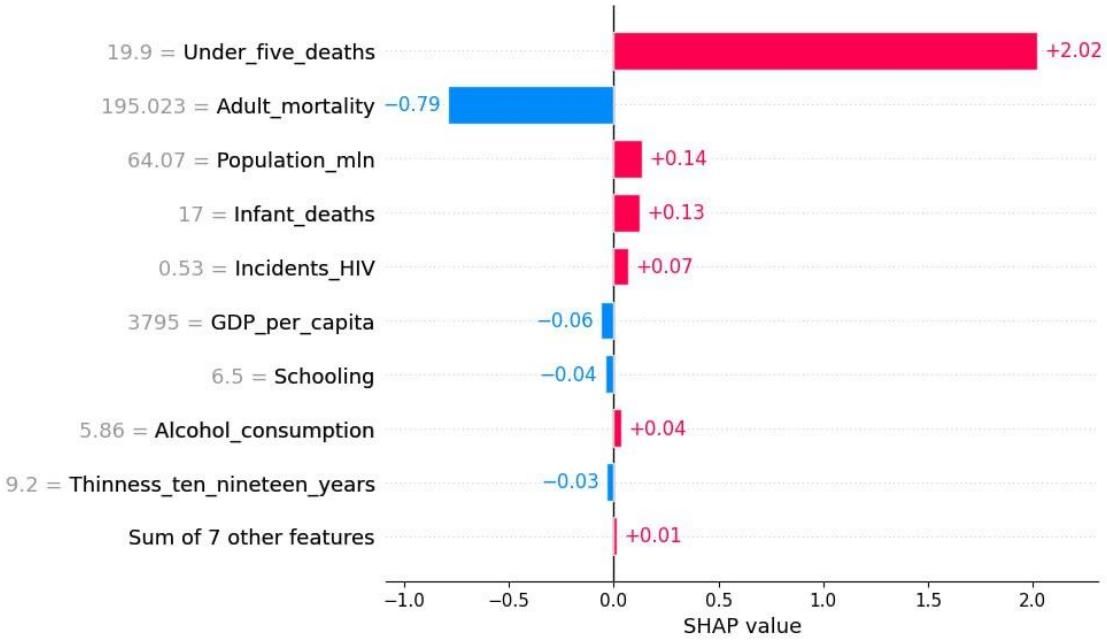
```
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

```
    # train a machine learning model

    model = RandomForestRegressor()
    model.fit(X_train, y_train)
```

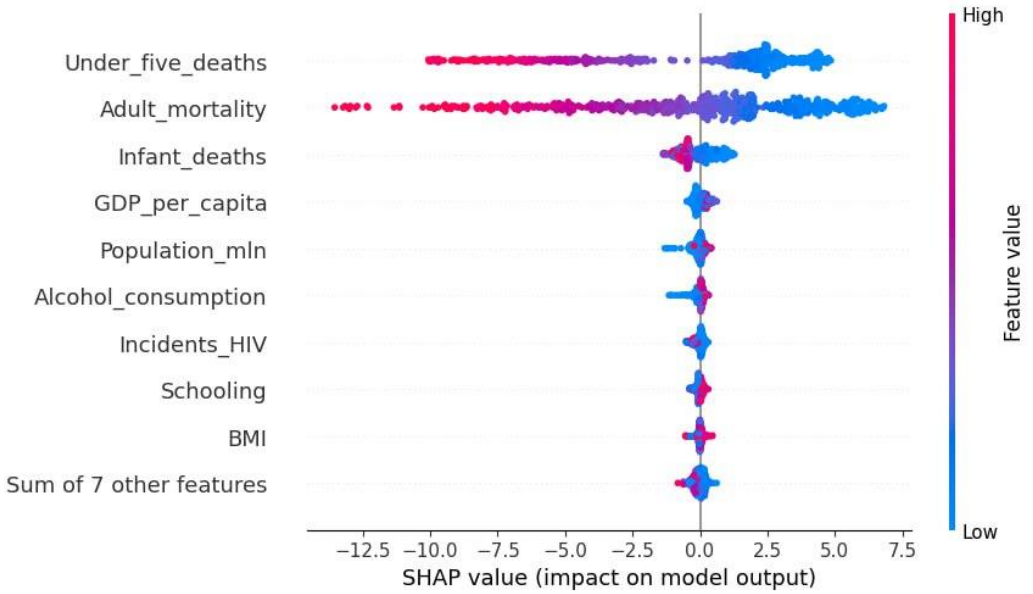
```
# fits the explainer

explainer = shap.ExactExplainer(model.predict, X_test)
shap_values = explainer(X_test)

shap.plots.bar(shap_values)
```



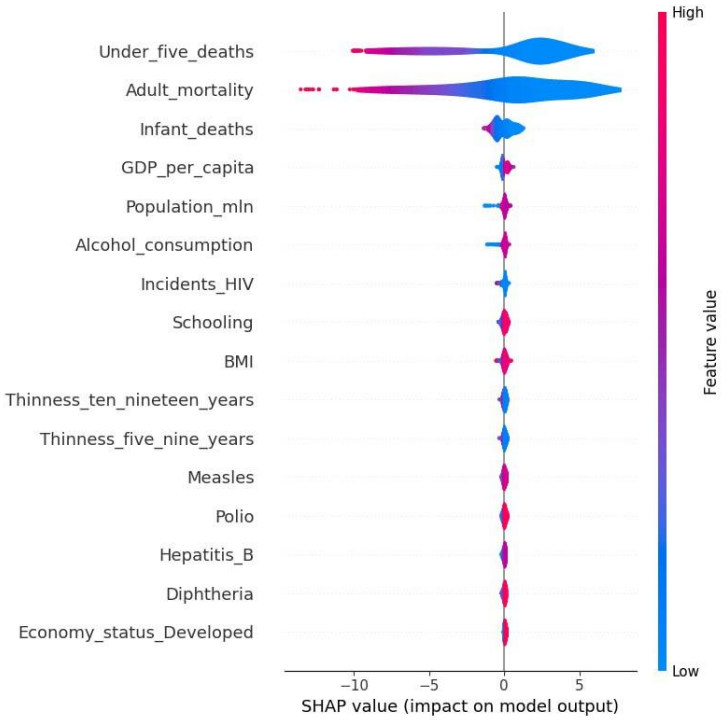
```
shap.plots.bar(shap_values[0])
```



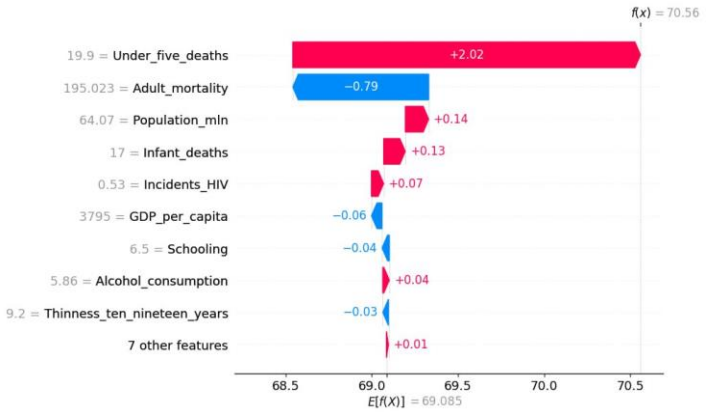

```
shap.plots.beeswarm(shap_values)
```



```
shap.summary_plot(shap_values, plot_type='violin')
```

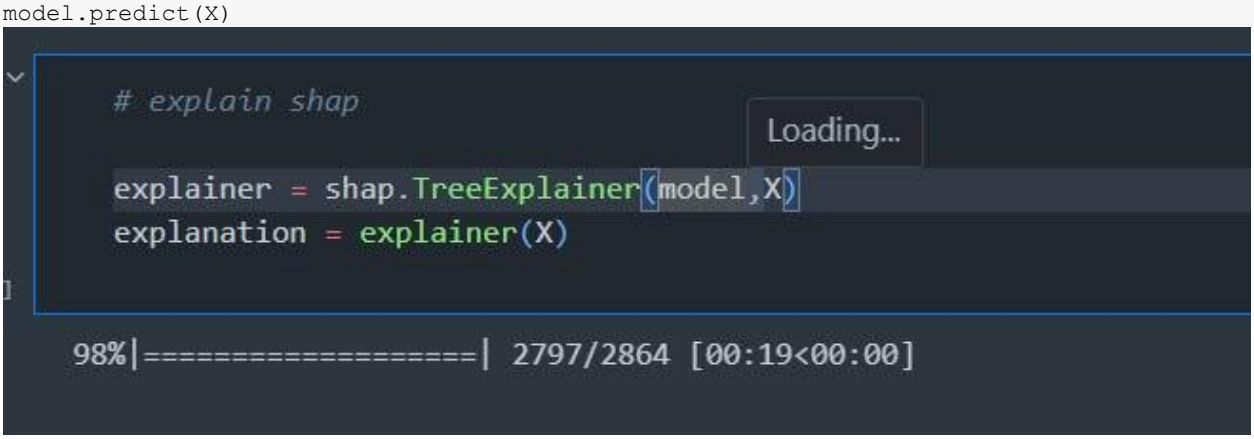



```
shap.plots.waterfall(shap_values[0])
```

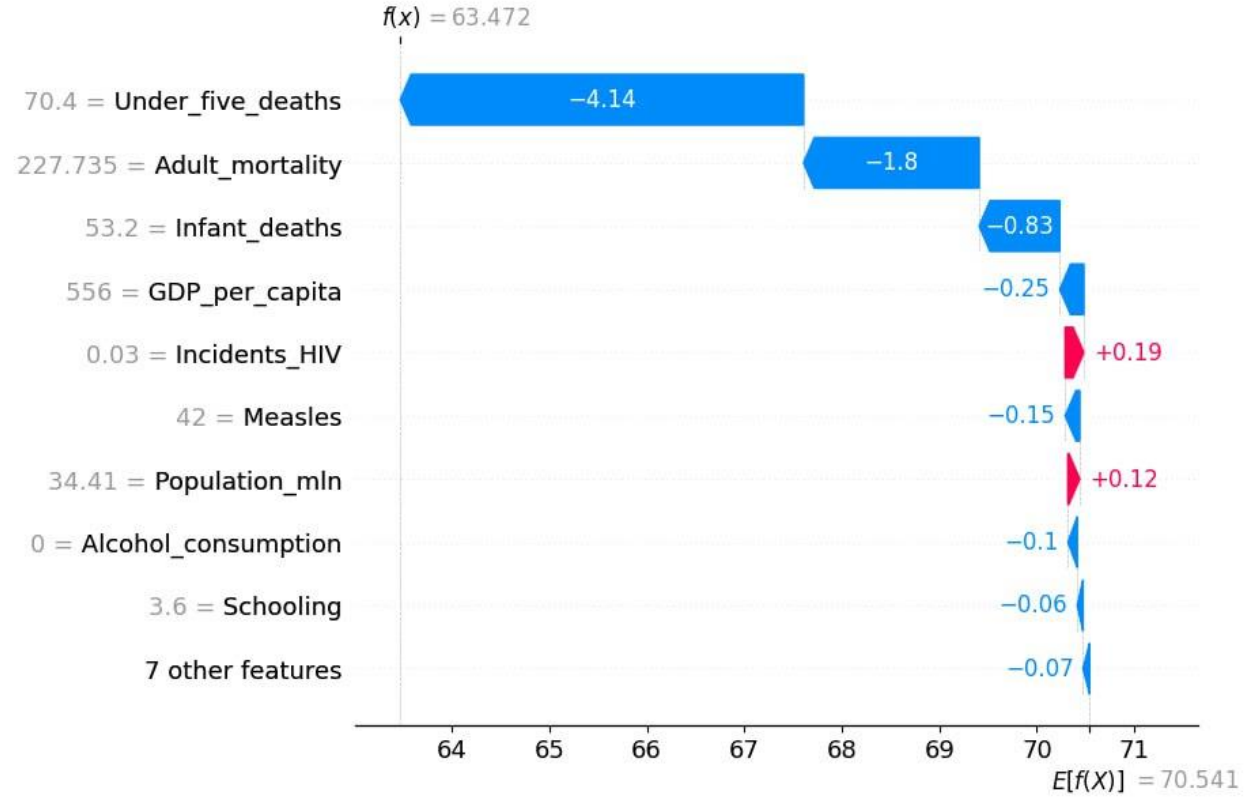


To use Random Forest, you first need to define the target feature as ‘y’ and specify the data that the machine should learn from as ‘X_train’. Then, you fit the model on the data. Now, it’s time for the SHAP module to provide us with the feature importance data.

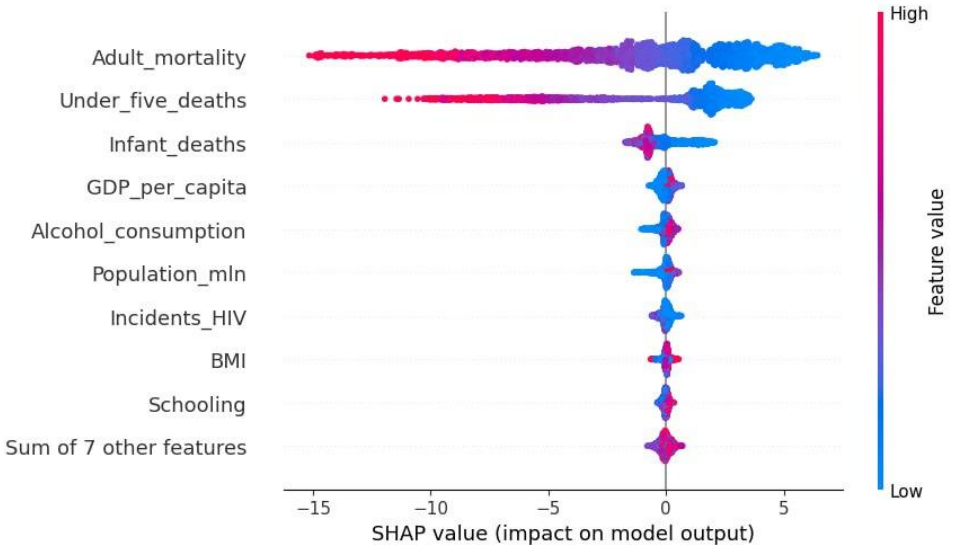
```
# create and fit model
model = xgboost.XGBRegressor(n_estimators=100, learning_rate=0.1, random_state=3)
model.fit(X,y)
model.predict(X)
```



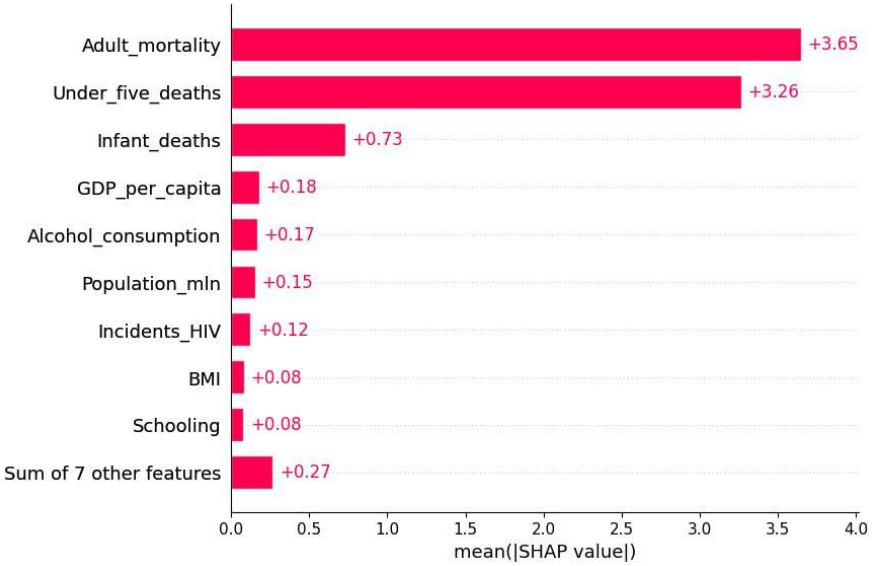

```
shap.plots.waterfall(explanation[0])
```



```
shap.plots.beeswarm(explanation)
```



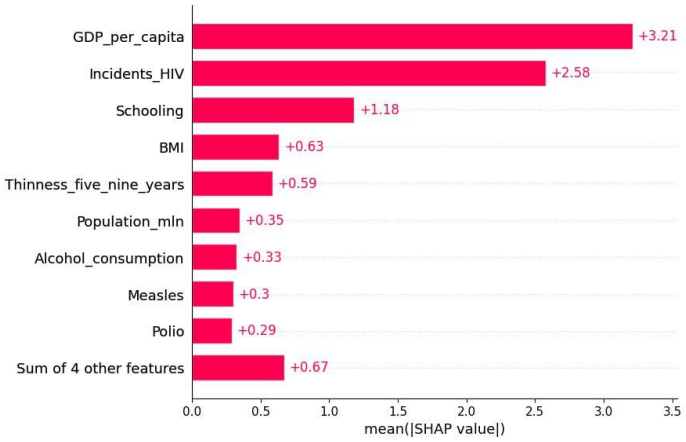

```
shap.plots.bar(explanation)
```

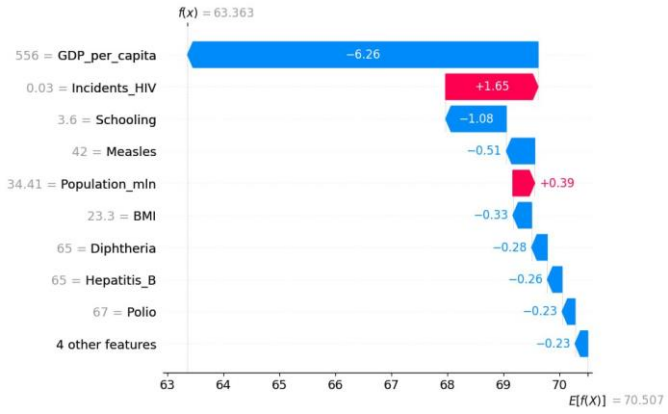


For using XGBoost, you would follow similar steps as with Random Forest. Define the target feature as ‘y’ and the data for the machine to learn from as ‘X_train’. Fit the model on the data. Finally, using the SHAP module, you can plot the desired graphs to visualize the feature importance.

It seems that the results obtained from both algorithms are similar for the top 3 most influential features. Therefore, you are planning to remove the first 3 features, namely Adult mortality, under-five deaths, and infant deaths. You will then repeat the steps with XGBoost to determine the effectiveness of the remaining features more accurately.

```
df = pd.read_csv('primary_dataset.csv')
features_drop = ['Country', 'Year', 'Region', 'Life_expectancy',
'Economy_status_Developing', 'Adult_mortality', 'Under_five_deaths', 'Infant_deaths']
X = df.drop(columns=features_drop, axis=1)

y = df.Life_expectancy
```





In the end, by comparing the outputs of both algorithms, it can be said that the most effective features are as follows:

1. Adult mortality
2. Infant death
3. GDP per capita
4. Population
5. Alcohol consumption
6. Incidence of HIV

After completing the feature importance process, you can provide analyses. For our analysis, we have chosen Iran as the country of focus. To present the analysis and make decisions on the topic of hope for life, we have taken this approach. Initially, we proceeded by removing the Year feature as we need to identify the countries with the highest life expectancy over a 15-year period.

```
df = df.groupby(['Country', 'Region']).mean().drop(columns='Year').reset_index()
df.head()
```

Python

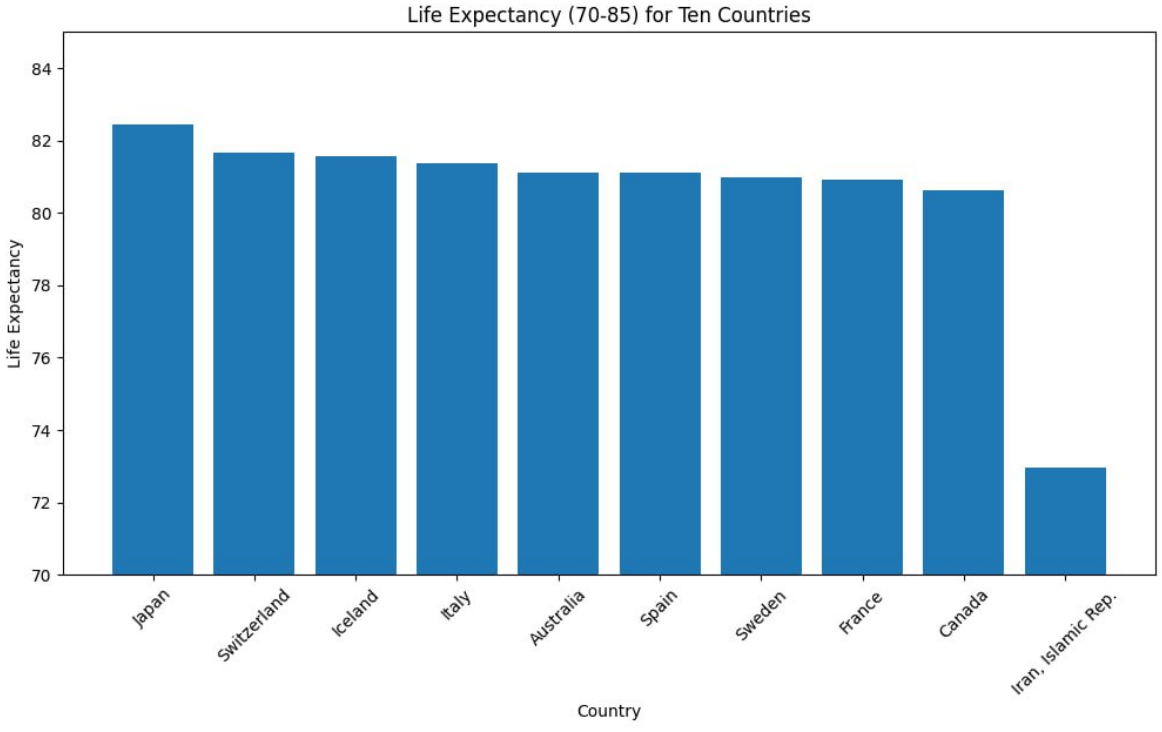
	Country	Region	Infant_deaths	Under_five_deaths	Adult_mortality	Alcohol_consumption	Hepatitis_B	Measles	BMI	Polio	Diphtheria	Incidents_HIV	GDP_per_capita
0	Afghanistan	Asia	71.08125	98.61250	265.804969	0.016125	64.5625	24.3750	22.46250	55.3750	55.1250	0.022500	408.5625
1	Albania	Rest of Europe	15.25625	17.14375	83.132969	4.696875	98.0000	95.9375	25.85625	98.1250	98.0625	0.025625	3071.1250
2	Algeria	Africa	26.75625	31.19375	113.439281	0.400625	88.3125	93.2500	24.86875	91.7500	91.8750	0.021875	3745.1250
3	Angola	Africa	88.76875	144.16250	297.844063	4.935625	68.8125	64.0000	22.51875	35.7500	55.5625	1.303750	2647.8125
4	Antigua and Barbuda	Central America and Caribbean	9.47500	11.51875	142.478813	7.755000	98.2500	75.4375	25.85000	96.9375	98.3125	0.125000	14678.7500

At this stage, you will extract a list of the top 9 countries with the highest life expectancy. Alongside Iran, you will plot bar graphs for each of the influential features to compare them among the countries. This comparison will provide valuable insights for your analysis and decision-making process.

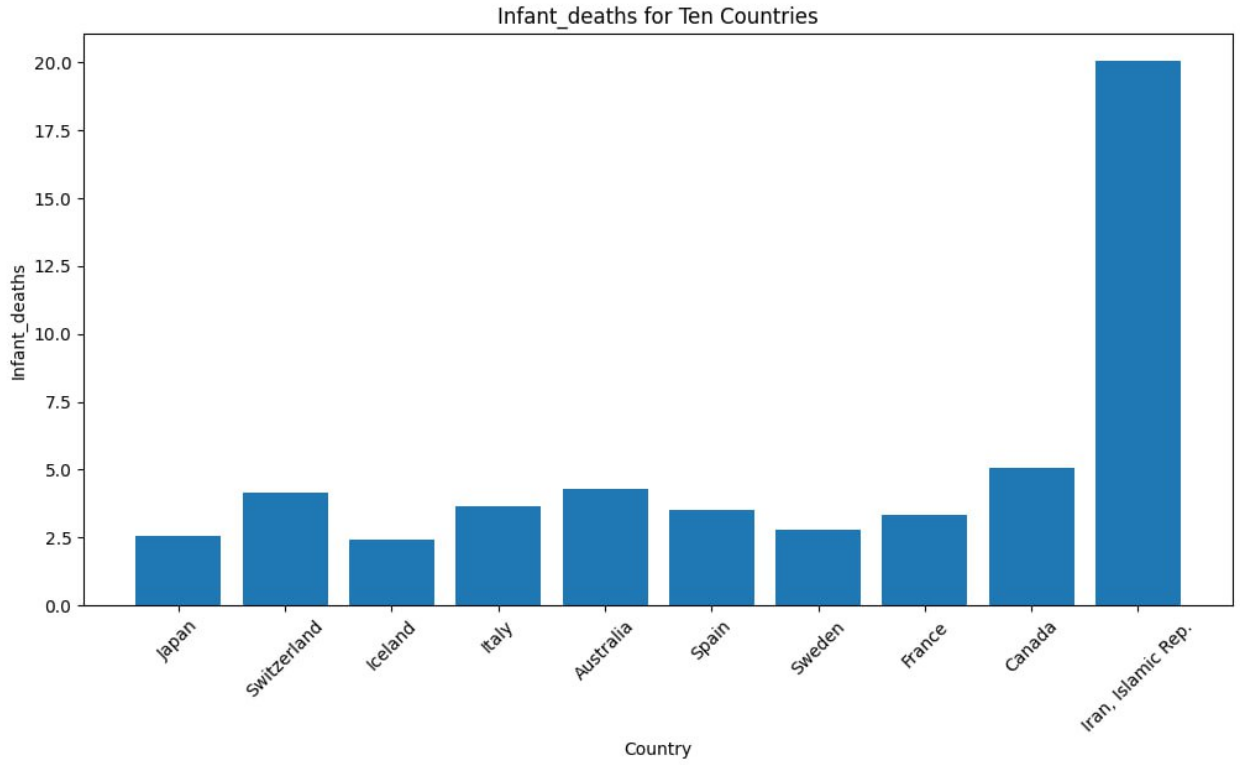
```
sorted_life_df = df.sort_values(by='Life_expectancy', ascending=False)
max_life_country = sorted_life_df.head(10)['Country'].tolist()
max_life_country
```

	Country	Region	Infant_deaths	Under_five_deaths	Adult_mortality	Alcohol_consumption	Hepatitis_B	Measles	BMI	Polio	Diphtheria	Incidents_HIV	GDP_per_capita
82	Japan	Asia	2.57500	3.50625	64.519094	7.931875	83.0000	88.8125	22.61875	96.0000	96.2500	0.170000	32972.312
154	Switzerland	Rest of Europe	4.13750	4.85000	61.564500	10.294375	88.0000	78.0625	25.00625	95.3125	94.5625	0.073125	79544.812
73	Iceland	Rest of Europe	2.41250	3.03125	59.145781	6.896875	88.0000	92.3125	25.90000	94.8125	94.8125	0.038750	48572.812
80	Italy	European Union	3.65625	4.31250	62.063969	8.195000	95.5000	83.4375	25.60000	96.1250	94.7500	0.082500	32284.250
7	Australia	Oceania	4.31250	5.15625	65.958594	10.145625	93.5000	87.2500	26.75000	91.8125	91.9375	0.046875	51750.937
148	Spain	European Union	3.50000	4.30000	70.445781	10.280000	92.5000	92.1875	26.13750	96.6875	96.7500	0.091875	25511.687
153	Sweden	European Union	2.78750	3.38750	62.071031	6.949375	61.0625	94.6875	25.61875	98.2500	98.3125	0.080000	46750.812
58	France	European Union	3.34375	4.53750	87.903781	12.824375	51.2500	67.6250	24.98125	98.2500	98.0625	0.105000	35447.250
30	Canada	North America	5.06875	5.87500	72.634969	8.056250	33.9375	84.6875	26.68750	91.2500	91.0625	0.100000	39683.625
76	Iran, Islamic Rep.	Middle East	20.04375	23.80000	118.194781	0.017188	97.7500	65.0000	25.58750	98.2500	98.3750	0.101875	4936.500

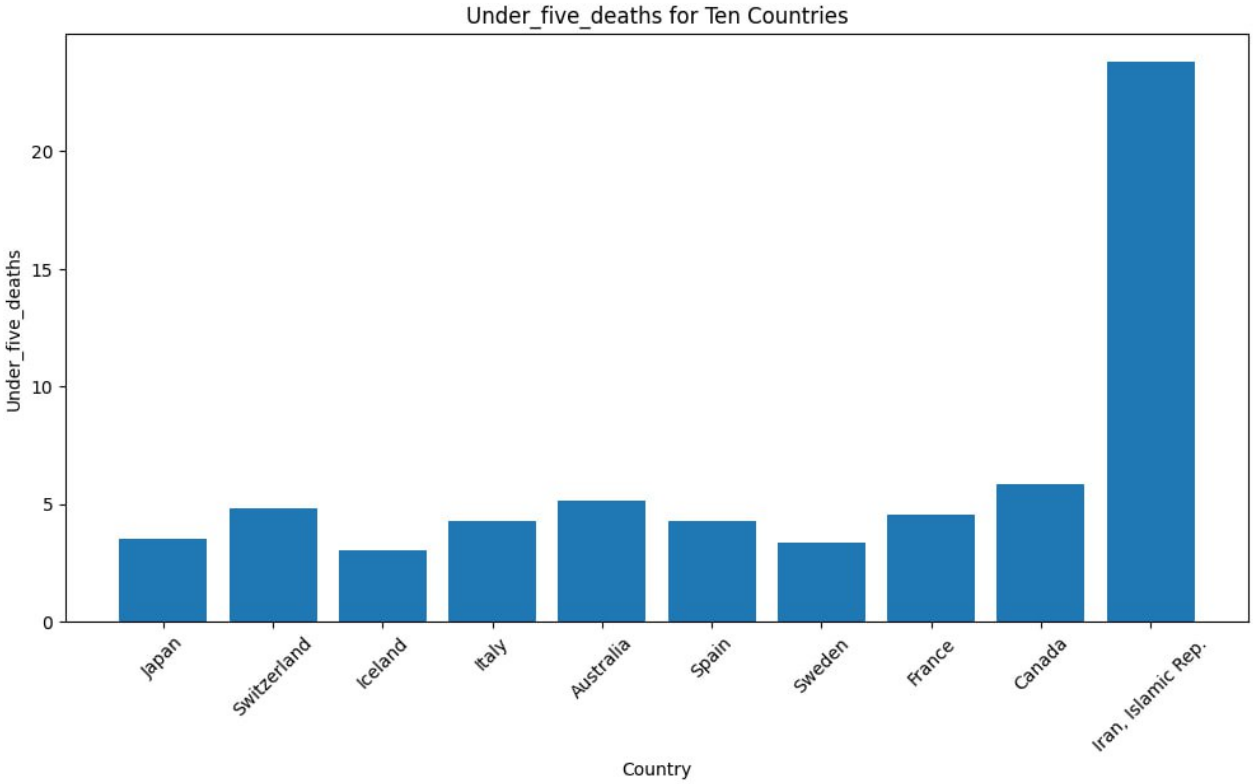
```
plt.figure(figsize=(12, 6))
plt.bar(max_life_country_df['Country'], max_life_country_df['Life_expectancy'])
plt.xlabel('Country')
plt.ylabel('Life Expectancy')
plt.title('Life Expectancy (70-85) for Ten Countries')
plt.ylim(70, 85)  # Set the y-axis limits to show only the range of 60-80
plt.xticks(rotation=45)
# plt.show()
plt.savefig("Life Expectancy (70-85) for Ten Countries.png")
```

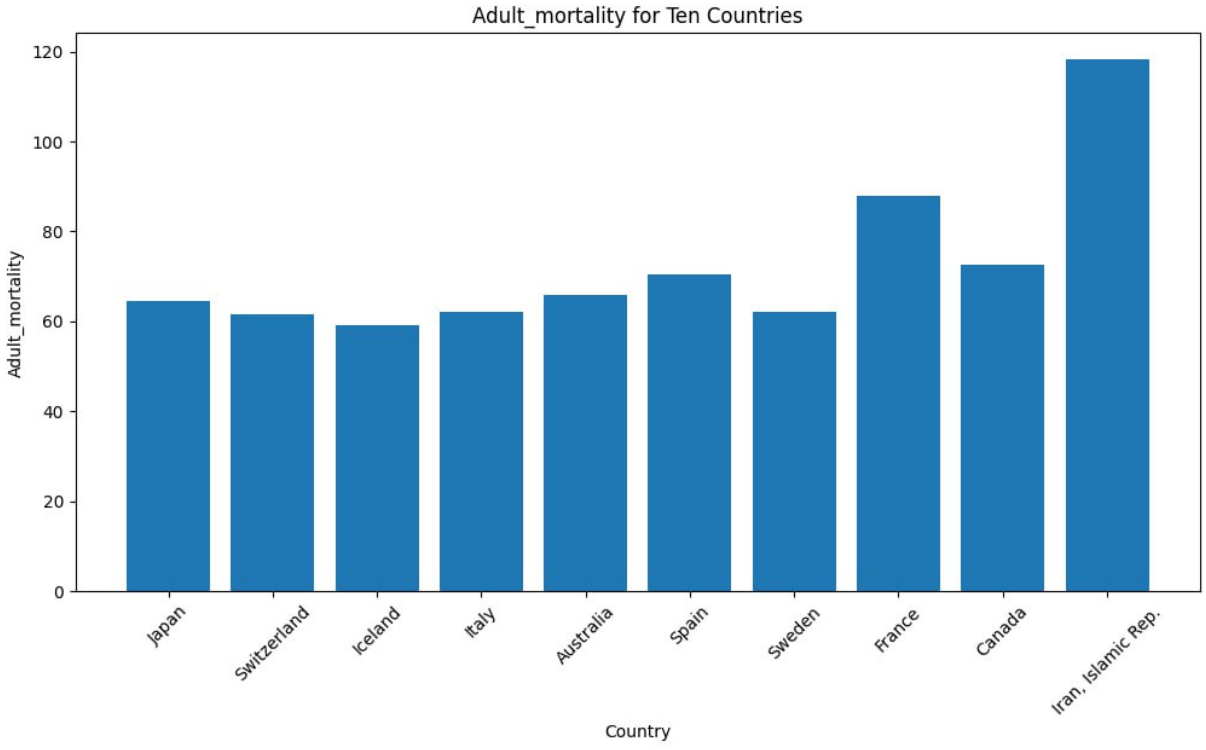


```python
# now we want to plot diagram of importance features
importance_features = ['Infant_deaths', 'Under_five_deaths','Adult_mortality',
'GDP_per_capita', 'Alcohol_consumption', 'Population_mln']

def plot_features(feature):
    plt.figure(figsize=(12, 6))
    plt.bar(max_life_country_df['Country'], max_life_country_df[feature])
    plt.xlabel('Country')
    plt.ylabel(feature)
    plt.title(f'{feature} for Ten Countries')
    plt.xticks(rotation=45)
    # plt.show()
    plt.savefig(f'{feature} for Ten countries.png')

for feature in importance_features:
    plot_features(feature)
```

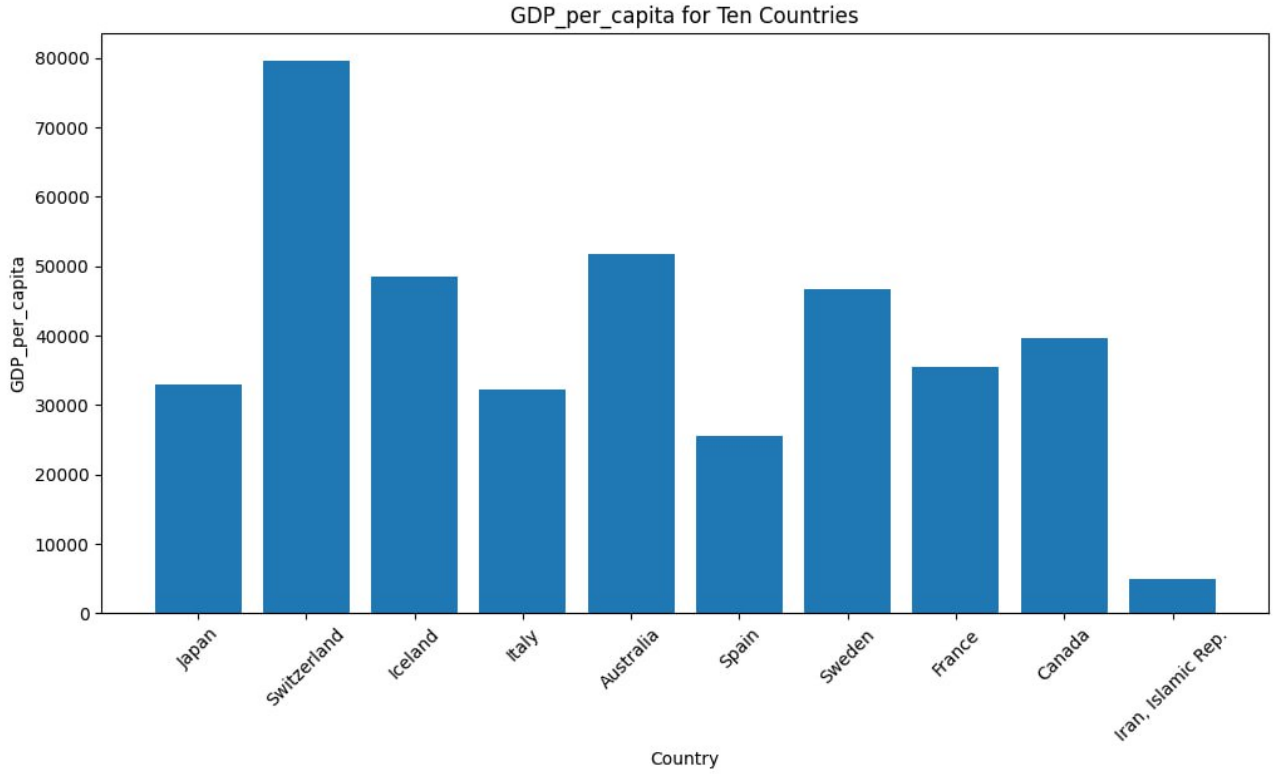


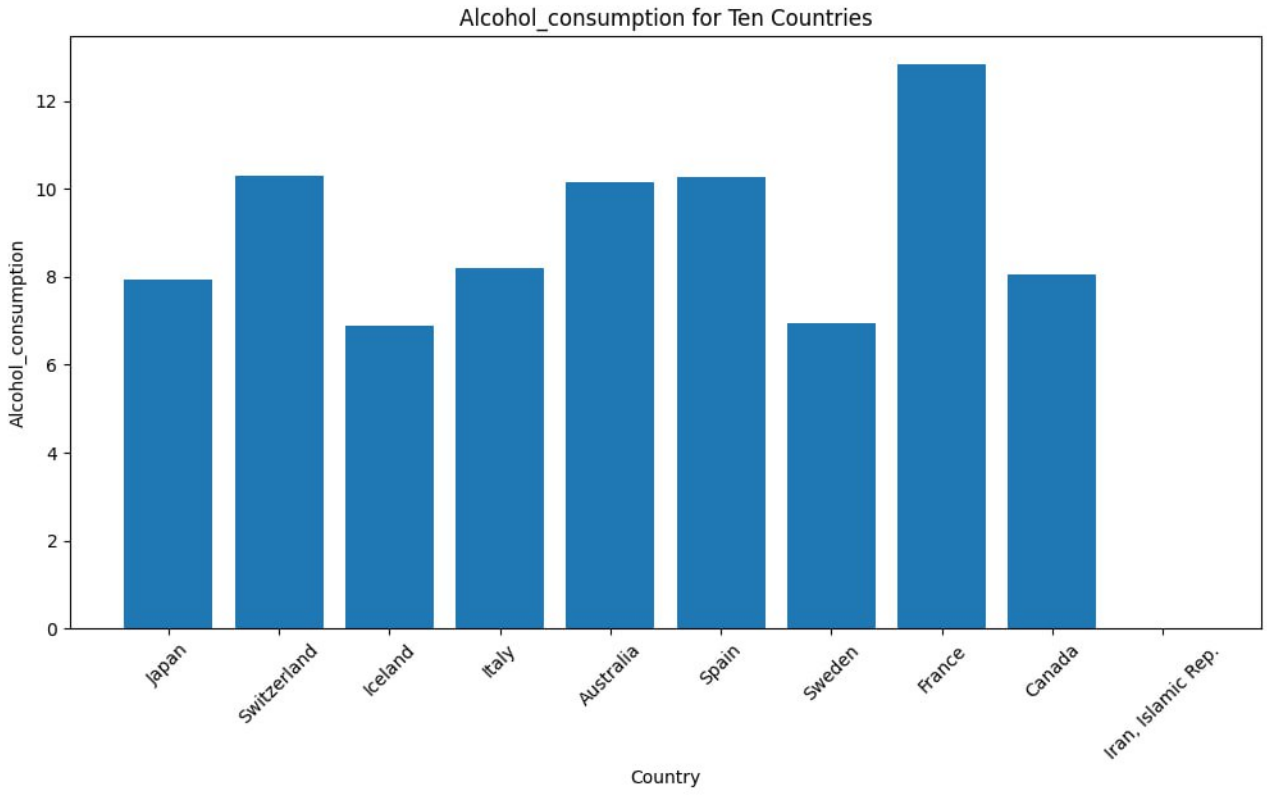
Advanced Programming the Final Project

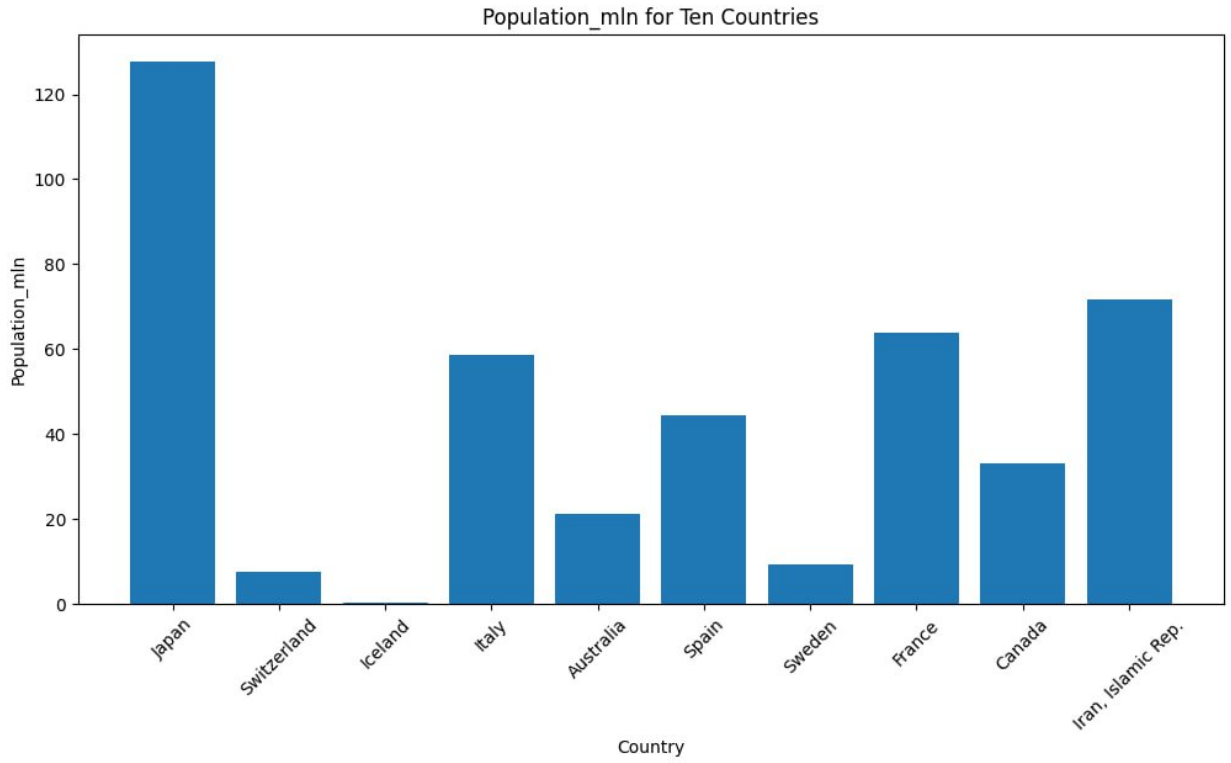




Advanced Programming the Final Project







Your statistical analysis provides valuable insights into various aspects of life expectancy and mortality rates in Iran compared to other countries. Here are some key takeaways from your analysis:

The life expectancy difference between Iran and the top 9 countries in the dataset is significant.

Infant mortality in Iran shows a substantial difference compared to the global average, suggesting a need to strengthen maternal and postnatal care facilities like NICU.

Similarly, under-five mortality rates in Iran also exhibit a notable difference, indicating the potential benefits of child screening programs, health index checks at health centers, parental education on care, and reducing child healthcare costs.

Adult mortality rates in Iran do not differ significantly from the global average, suggesting that improving healthcare facilities and preventing sudden deaths could further reduce this indicator.

There is a significant difference in GDP among countries, highlighting the importance of economic indicators like GDP for policy-making and economic development.

The analysis reveals that alcohol consumption in Iran is notably lower compared to European countries and Japan, reflecting cultural and religious differences. This underscores the need for nuanced analysis considering economic and cultural disparities among countries.

These insights provide a solid foundation for further detailed analysis and policy considerations in healthcare, economic development, and cultural contexts.

7. Limitations of the Dataset

As you continue to examine the constraints and suggestions for your dataset and analysis, consider the following key points:

Healthcare expenditure:
The level of expenditure in a country can significantly impact life expectancy and quality of life.

Cigarette consumption:
This factor can have a substantial impact on health and overall well-being.

Marital status:
Romantic relationships and marriage can influence life expectancy and quality of life.

Physical activity:
Engaging in sports and exercise can improve longevity and overall quality of life.

Access to healthcare facilities and personal hygiene practices:
These aspects are crucial for maintaining health and increasing life expectancy.

Data antiquity:
The historical context of data can affect the accuracy of your analysis.

Ambiguity in country definitions:
Clear and precise definitions of countries are essential for a more accurate analysis.

Examining the healthcare trends of countries, categorizing based on culture and religion, and comparing with similar countries culturally and religiously can help you provide better solutions for improving healthcare conditions and life expectancy. These critical points can enhance your analysis and enable you to present the best decisions for health and better living.

Data Sources:

Average life expectancy of both genders in different years from 2010 to 2015:
https://www.who.int/data/gho/data/indicators/indicator-details/GHO/life-expectancy-at-birth-(years)

Mortality-related attributes (infant deaths, under-five-deaths, adult mortality):
https://www.who.int/data/gho/data/themes/mortality-and-global-health-estimates

Alcohol consumption that is recorded in liters of pure alcohol per capita with 15+ years old:
https://www.who.int/data/gho/data/indicators/indicator-details/GHO/alcohol-recorded-per-capita-(15-)-consumption-(in-litres-of-pure-alcohol)

% of coverage of Hepatitis B (HepB3) immunization among 1-year-olds:
https://www.who.int/data/gho/data/indicators/indicator-details/GHO/hepatitis-b-(hepb3)-immunization-coverage-among-1-year-olds-(-)

% of coverage of Measles containing vaccine first dose (MCV1) immunization among 1-year-olds:
https://www.who.int/data/gho/data/indicators/indicator-details/GHO/measles-containing-vaccine-first-dose-(mcv1)-immunization-coverage-among-1-year-olds-(-)

% of coverage of Polio (Pol3) immunization among 1-year-olds:
https://www.who.int/data/gho/data/indicators/indicator-details/GHO/polio-(pol3)-immunization-coverage-among-1-year-olds-(-)

% of coverage of Diphtheria tetanus toxoid and pertussis (DTP3) immunization among 1-year-olds:
https://www.who.int/data/gho/data/indicators/indicator-details/GHO/diphtheria-tetanus-toxoid-and-pertussis-(dtp3)-immunization-coverage-among-1-year-olds-(-)

BMI:
https://www.who.int/europe/news-room/fact-sheets/item/a-healthy-lifestyle---who-recommendations

Incidents of HIV per 1000 population aged 15-49:
https://data.worldbank.org/indicator/SH.HIV.INCD.ZS

Prevalence of thinness among adolescents aged 10-19 years. BMI < -2 standard deviations below the median: https://www.who.int/data/gho/indicator-metadata-registry/imr-details/4805

GDP per capita in current USD:
https://data.worldbank.org/indicator/NY.GDP.PCAP.CD?most_recent_year_desc=true

Total population in millions:
https://data.worldbank.org/indicator/SP.POP.TOTL?most_recent_year_desc=true

Average years that people aged 25+ spent in formal education:
https://ourworldindata.org/grapher/mean-years-of-schooling-long-run