

```
In [15]: import pandas as pd

# Load the data
df = pd.read_csv("/Users/ranurjajput/Desktop/world_correlation_dataset.csv")

# View first 5 rows
df.head(5)

# Check for nulls
df.info()
df.isnull().sum()

# Basic stats
df.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 7 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Country                               1000 non-null   object
1   Education_Index                       960 non-null    float64
2   GDP_Per_Capita                        944 non-null    float64
3   Life_Expectancy                       948 non-null    float64
4   Happiness_Score                       951 non-null    float64
5   Internet_Penetration                 949 non-null    float64
6   Corruption_Index                     940 non-null    float64
dtypes: float64(6), object(1)
memory usage: 54.8+ KB
```

```
Out[15]:
```

	Education_Index	GDP_Per_Capita	Life_Expectancy	Happiness_Score	Interi
count	960.000000	944.000000	948.000000	951.000000	
mean	0.670031	36211.166631	70.065612	4.949127	
std	0.160862	20183.446361	8.762856	1.719156	
min	0.400000	1222.060000	55.000000	2.000000	
25%	0.530000	18135.695000	62.800000	3.455000	
50%	0.670000	37091.185000	70.100000	4.920000	
75%	0.810000	53573.470000	77.800000	6.425000	
max	0.950000	69959.550000	84.900000	8.000000	

```
In [19]: # Drop values with null values
df_cleaned = df.dropna()
df_cleaned.columns = df_cleaned.columns.str.replace(" ", "_")
```

```
In [30]: df_cleaned = df_cleaned.select_dtypes(include=['number'])
correlation_matrix = df_cleaned.corr()
print(correlation_matrix)
```

	Education_Index	GDP_Per_Capita	Life_Expectancy	\
Education_Index	1.000000	0.010837	-0.011487	
GDP_Per_Capita	0.010837	1.000000	0.034116	
Life_Expectancy	-0.011487	0.034116	1.000000	
Happiness_Score	-0.044791	0.002707	-0.004329	
Internet_Penetration	-0.003456	0.022447	0.004037	
Corruption_Index	-0.031062	0.009076	-0.036431	

	Happiness_Score	Internet_Penetration	Corruption_In
dex			
Education_Index	-0.044791	-0.003456	-0.031
062			
GDP_Per_Capita	0.002707	0.022447	0.009
076			
Life_Expectancy	-0.004329	0.004037	-0.036
431			
Happiness_Score	1.000000	-0.052697	0.029
205			
Internet_Penetration	-0.052697	1.000000	-0.046
631			
Corruption_Index	0.029205	-0.046631	1.000
000			

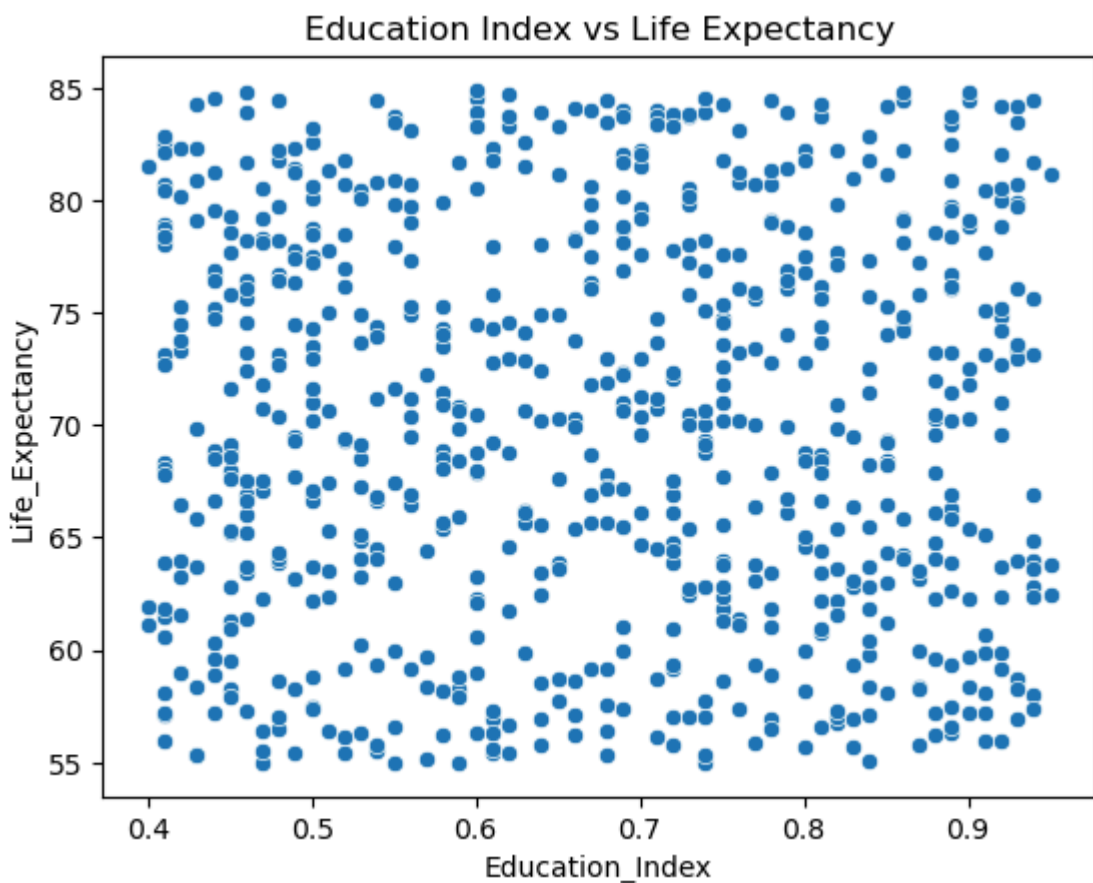
```
In [36]: import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm")
plt.title("Correlation Matrix of Socio-Economic Indicators")
plt.show()
```



```
In [38]: sns.scatterplot(data=df_cleaned, x="GDP_Per_Capita", y="Happiness_Score")
plt.title("GDP vs Happiness")
plt.show()
```

```
sns.scatterplot(data=df_cleaned, x="Education_Index", y="Life_Expectancy")  
plt.title("Education Index vs Life Expectancy")  
plt.show()
```



```
In [40]: from scipy.stats import pearsonr, spearmanr

pearson_corr, _ = pearsonr(df_cleaned['GDP_Per_Capita'], df_cleaned['Happ
print(f"Pearson Correlation between GDP and Happiness: {pearson_corr:.2f}")
```

Pearson Correlation between GDP and Happiness: 0.00

```
In [ ]: # Document Insights
# GDP_Per_Capita has Very weak positive or no correlation with Education_
# Internet_Penetration has Very weak negative Corruption_Index
```