**Program**

# Import necessary libraries

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from scipy import stats

from pandas.plotting import lag\_plot, autocorrelation\_plot

**# Define the dataset within the code**

data = {

    'Country': ['Finland', 'Denmark', 'Norway', 'Iceland', 'Netherlands', 'Switzerland', 'Sweden', 'New Zealand',

                'Canada', 'Austria', 'Australia', 'Israel', 'Costa Rica', 'Ireland', 'Germany', 'United States',

                'Czech Republic', 'Belgium', 'United Kingdom', 'Mexico'],

    'Happiness Rank': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20],

    'Happiness Score': [7.769, 7.600, 7.554, 7.494, 7.488, 7.480, 7.343, 7.307, 7.278, 7.246, 7.228, 7.190, 7.167,

                        7.129, 7.076, 6.892, 6.852, 6.834, 6.796, 6.595],

    'GDP per Capita': [1.340, 1.383, 1.488, 1.380, 1.396, 1.452, 1.387, 1.303, 1.365, 1.376, 1.372, 1.276, 1.034,

                       1.499, 1.373, 1.433, 1.269, 1.356, 1.333, 1.080],

    'Social Support': [1.587, 1.573, 1.582, 1.624, 1.522, 1.526, 1.487, 1.557, 1.505, 1.475, 1.548, 1.455, 1.441,

                       1.553, 1.454, 1.457, 1.487, 1.504, 1.538, 1.323],

    'Healthy Life Expectancy': [0.986, 0.996, 1.028, 1.026, 0.999, 1.052, 1.009, 1.026, 1.039, 1.016, 1.036, 1.029,

                                0.963, 1.010, 0.987, 0.874, 0.920, 0.986, 0.996, 0.861],

    'Freedom to Make Life Choices': [0.596, 0.592, 0.603, 0.591, 0.557, 0.572, 0.574, 0.585, 0.584, 0.532, 0.557,

                                     0.371, 0.558, 0.516, 0.495, 0.454, 0.457, 0.473, 0.450, 0.482],

    'Generosity': [0.153, 0.252, 0.271, 0.354, 0.322, 0.263, 0.267, 0.330, 0.285, 0.244, 0.322, 0.261, 0.144, 0.298,

                   0.261, 0.280, 0.046, 0.160, 0.348, 0.074],

    'Perceptions of Corruption': [0.272, 0.410, 0.341, 0.118, 0.298, 0.343, 0.373, 0.389, 0.308, 0.226, 0.290, 0.082,

                                  0.093, 0.310, 0.265, 0.128, 0.036, 0.210, 0.278, 0.153],

    'Dystopia Residual': [2.123, 2.064, 2.154, 2.264, 2.276, 2.163, 2.038, 2.081, 2.131, 2.153, 2.192, 2.309, 2.499,

                          1.980, 2.064, 1.921, 2.361, 2.113, 1.969, 2.686]

}

**# Convert to a pandas DataFrame**

df = pd.DataFrame(data)

**# Display the first few rows**

print(df.head())

**# Paired t-test: Check if there's a significant difference between Happiness Score and GDP per** Capita

t\_stat\_paired, p\_value\_paired = stats.ttest\_rel(df['Happiness Score'], df['GDP per Capita'])

print(f"Paired T-statistic: {t\_stat\_paired}, Paired P-value: {p\_value\_paired}")

if p\_value\_paired < 0.05:

    print("Significant difference between Happiness Score and GDP per Capita (Reject null hypothesis).")

else:

    print("No significant difference between Happiness Score and GDP per Capita (Fail to reject null hypothesis).")

**# Independent t-test: Compare Happiness Score for countries with above and below median GDP per Capita**

median\_gdp = df['GDP per Capita'].median()

above\_median\_gdp = df[df['GDP per Capita'] > median\_gdp]['Happiness Score']

below\_median\_gdp = df[df['GDP per Capita'] <= median\_gdp]['Happiness Score']

t\_stat\_ind, p\_value\_ind = stats.ttest\_ind(above\_median\_gdp, below\_median\_gdp)

print(f"Independent T-statistic: {t\_stat\_ind}, Independent P-value: {p\_value\_ind}")

if p\_value\_ind < 0.05:

    print("Significant difference in Happiness Score between countries with above and below median GDP per Capita (Reject null hypothesis).")

else:

    print("No significant difference in Happiness Score between countries with above and below median GDP per Capita (Fail to reject null hypothesis).")

**# Line Plot: GDP per Capita vs Happiness Score over the Rank**

plt.figure(figsize=(10, 6))

plt.plot(df['Happiness Rank'], df['Happiness Score'], label='Happiness Score', marker='o')

plt.plot(df['Happiness Rank'], df['GDP per Capita'], label='GDP per Capita', marker='o')

plt.xlabel('Happiness Rank')

plt.ylabel('Score')

plt.title('Line Plot: Happiness Score and GDP per Capita over Rank')

plt.legend()

plt.show()

**# Box Plot: Showing distribution of Happiness Score and GDP per Capita**

plt.figure(figsize=(10, 6))

sns.boxplot(data=df[['Happiness Score', 'GDP per Capita']], palette="Set2")

plt.title('Box Plot of Happiness Score and GDP per Capita')

plt.show()

**# Heatmap: Correlation between the variables**

plt.figure(figsize=(10, 6))

corr\_matrix = df[['Happiness Score', 'GDP per Capita', 'Social Support', 'Healthy Life Expectancy',

                  'Freedom to Make Life Choices', 'Generosity', 'Perceptions of Corruption']].corr()

sns.heatmap(corr\_matrix, annot=True, cmap='coolwarm', linewidths=0.5)

plt.title('Heatmap of Happiness Factors Correlation')

plt.show()

**# Lag Plot: Lag of Happiness Score to check for autocorrelation**

plt.figure(figsize=(10, 6))

lag\_plot(df['Happiness Score'])

plt.title('Lag Plot of Happiness Score')

plt.show()

**# Autocorrelation Plot: For Happiness Score**

plt.figure(figsize=(10, 6))

autocorrelation\_plot(df['Happiness Score'])

plt.title('Autocorrelation Plot of Happiness Score')

**plt.show()**

**# 1. Bar Plot: Happiness Score by Country**

plt.figure(figsize=(10, 6))

sns.barplot(x='Country', y='Happiness Score', data=df)

plt.xticks(rotation=90)

plt.title('Bar Plot of Happiness Score by Country')

plt.show()

**# 2. Pair Plot: Relationships between all factors**

sns.pairplot(df[['Happiness Score', 'GDP per Capita', 'Social Support', 'Healthy Life Expectancy']])

plt.show()

**# 3. Violin Plot: Distribution of Happiness Score**

plt.figure(figsize=(10, 6))

sns.violinplot(x='Happiness Rank', y='Happiness Score', data=df)

plt.title('Violin Plot of Happiness Score Distribution')

plt.show()

**# 4. KDE Plot: Happiness Score Density**

plt.figure(figsize=(10, 6))

sns.kdeplot(df['Happiness Score'], shade=True)

plt.title('KDE Plot of Happiness Score Density')

plt.show()

**# 5. Scatter Plot: GDP per Capita vs Happiness Score**

plt.figure(figsize=(10, 6))

plt.scatter(df['GDP per Capita'], df['Happiness Score'], color='blue')

plt.xlabel('GDP per Capita')

plt.ylabel('Happiness Score')

plt.title('Scatter Plot of GDP per Capita vs Happiness Score')

plt.show()

**# 6. Boxen Plot: Distribution of Freedom to Make Life Choices**

plt.figure(figsize=(10, 6))

sns.boxenplot(x='Happiness Rank', y='Freedom to Make Life Choices', data=df)

plt.title('Boxen Plot of Freedom to Make Life Choices')

plt.show()

**# 7. Strip Plot: Healthy Life Expectancy**

plt.figure(figsize=(10, 6))

sns.stripplot(x='Happiness Rank', y='Healthy Life Expectancy', data=df)

plt.title('Strip Plot of Healthy Life Expectancy')

plt.show()

**# 8. Swarm Plot: Social Support**

plt.figure(figsize=(10, 6))

sns.swarmplot(x='Happiness Rank', y='Social Support', data=df)

plt.title('Swarm Plot of Social Support')

plt.show()

**# 9. Rug Plot: Generosity**

plt.figure(figsize=(10, 6))

sns.rugplot(df['Generosity'])

plt.title('Rug Plot of Generosity')

plt.show()

**# 10. Point Plot: Happiness Score by Rank**

plt.figure(figsize=(10, 6))

sns.pointplot(x='Happiness Rank', y='Happiness Score', data=df)

plt.title('Point Plot of Happiness Score by Rank')

plt.show()

**# Import the necessary library for hypothesis testing**

from scipy import stats

**# Perform a paired t-test**

t\_stat, p\_value = stats.ttest\_rel(df['Happiness Score'], df['GDP per Capita'])

**# Output the results**

print(f"T-statistic: {t\_stat}, P-value: {p\_value}")

**# Check the result and print conclusion**

if p\_value < 0.05:

    print("There is a significant difference between Happiness Score and GDP per Capita (Reject the null hypothesis).")

else:

    print("There is no significant difference between Happiness Score and GDP per Capita (Fail to reject the null hypothesis).")

**# Calculate the median GDP per Capita**

median\_gdp = df['GDP per Capita'].median()

**# Split the data into two groups: above and below the median GDP per Capita**

above\_median\_gdp = df[df['GDP per Capita'] > median\_gdp]['Happiness Score']

below\_median\_gdp = df[df['GDP per Capita'] <= median\_gdp]['Happiness Score']

**# Perform an independent t-test (two-sample t-test)**

t\_stat, p\_value = stats.ttest\_ind(above\_median\_gdp, below\_median\_gdp)

**# Output the results**

print(f"T-statistic: {t\_stat}, P-value: {p\_value}")

**# Check the result and print conclusion**

if p\_value < 0.05:

    print("There is a significant difference in Happiness Score between countries with above and below median GDP per Capita (Reject the null hypothesis).")

else:

    print("There is no significant difference in Happiness Score between countries with above and below median GDP per Capita (Fail to reject the null hypothesis).")

**Output**









