```
#Problem - 1: Getting Started with Data Exploration
#1. Data Exploration and Understanding:
#• Dataset Overview:
#1. Load the dataset and display the first 10 rows.
import pandas as pd
dataset = pd.read_csv("/content/drive/MyDrive/Copy
print("First 10 rows are:")
print(dataset.head(10))

#2. Identify the number of rows and columns in the
dataset = pd.read_csv("/content/drive/MyDrive/Copy
rows, columns = dataset.shape
print("Number of rows:",rows)
print("Number of Columns:",columns)

#3. List all the columns and their data types.
dataset = pd.read_csv("/content/drive/MyDrive/Copy
print("All the columns and their data types:")
print(dataset.dtypes)
```

```
First 10 rows are:
  Country name  score  Log GDP per capita  Soc
0      Finland  7.741               1.844
1      Denmark  7.583               1.908
2      Iceland  7.525               1.881
3       Sweden  7.344               1.878
4       Israel  7.341               1.803
5  Netherlands  7.319               1.901
6       Norway  7.302               1.952
7   Luxembourg  7.122               2.141
8  Switzerland  7.060               1.970
9    Australia  7.057               1.854

   Healthy life expectancy  Freedom to make li
0                    0.695
1                    0.699
2                    0.718
3                    0.724
4                    0.740
5                    0.706
6                    0.704
7                    0.708
8                    0.747
9                    0.692

   Perceptions of corruption  Dystopia + resic
0                      0.546               2.
1                      0.548               1.
2                      0.182               2.
3                      0.524               1.
4                      0.193               2.
5                      0.372               1.
6                      0.484               1.
7                      0.432               1.
```

KeyError KeyError •••

You

Please explain the error:

```
TypeError: Could not convert
```

Gemini

Hypothesis

The error message "TypeError: Could not convert ['NepalPakistanIndiaBangladeshAfg to numeric" indicates that the `mean()` function is being applied to a column containing string values, specifically the 'Country name' column. You are including 'Country name' in the list of columns to calculate the mean for, but it is not numeric.

Suggested Changes

```
pport', 'Healthy life expec'
support', 'Healthy life expe
```

Use code with caution

Rate this answer

```
8                    0.498                1.
9                    0.323                1.
Number of rows: 143
Number of Columns: 9
All the columns and their data types:
Country name                     object
score                           float64
Log GDP per capita              float64
Social support                  float64
Healthy life expectancy         float64
Freedom to make life choices    float64
Generosity                      float64
Perceptions of corruption       float64
Dystopia + residual             float64
dtype: object
```

```
#• Basic Statistics:
#1. Calculate the mean, median, and standard devia

# Loading dataset
dataset = pd.read_csv('/content/drive/MyDrive/Copy

# Calculating basic statistics from the 'score' co
mean = dataset['score'].mean()
median= dataset['score'].median()
standard_deviation= dataset['score'].std()

# Displaying results
print(f"Mean Score is: {mean}")
print(f"Median Score is: {median}")
print(f"Standard deviation score is: {standard_dev

#2. Identify the country with the highest and lowe

# Loading dataset
dataset = pd.read_csv('/content/drive/MyDrive/Copy

# To find the highest and lowest score
highest = dataset.nlargest(1, 'score')
lowest = dataset.nsmallest(1, 'score')

#displaying result
print("Country with the highest happiness score:",
print("Country with the lowest score :",lowest)
```

```
Mean Score is: 5.52758041958042
Median Score is: 5.785
Standard deviation score is: 1.170716509944299
Country with the highest happiness score:   Co
0      Finland  7.741               1.844

   Healthy life expectancy  Freedom to make li
0                    0.695
```

```
   Perceptions of corruption  Dystopia + resic
0                      0.546                2.
Country with the lowest score :     Country na
142  Afghanistan  1.721               0.628

     Healthy life expectancy  Freedom to make
142                    0.242

     Perceptions of corruption  Dystopia + res
142                      0.088
```

```
#Missing Values:
#1. Check if there are any missing values in the c

# Loading dataset
dataset = pd.read_csv('/content/drive/MyDrive/Copy

# Searching missing values
missing_values = dataset.isnull().sum()

# Displaying missing values
print("Missing values in each columns:\n", dataset
```

```
Missing values in each columns:
 Country name                     0
score                            0
Log GDP per capita               3
Social support                   3
Healthy life expectancy          3
Freedom to make life choices     3
Generosity                       3
Perceptions of corruption        3
Dystopia + residual              3
dtype: int64
```

```
 and Sorting:
ne dataset to show only the countries with a Score

aset
.read_csv('/content/drive/MyDrive/Copy of WHR-2024-

and displaying countries with Score greater than 7.
t[dataset['score'] > 7.5])


filtered dataset - Sort the dataset by GDP per Capi
.

taset
.read_csv('/content/drive/MyDrive/Copy of WHR-2024-

 sorting, and displaying top 10 rows
aset[dataset['score'] > 7.5].sort_values(by='Log GD

  result
```

```
)
```

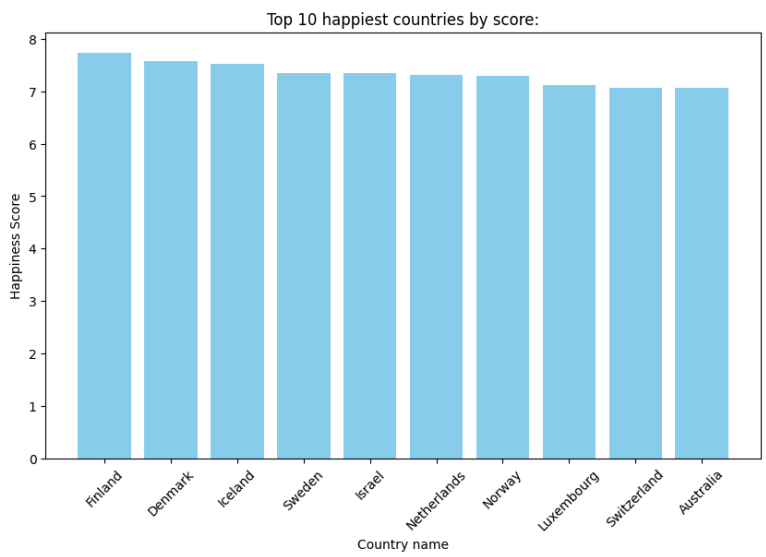
```
  Country name  score  Log GDP per capita  Soc
0      Finland  7.741               1.844
1      Denmark  7.583               1.908
2      Iceland  7.525               1.881

   Healthy life expectancy  Freedom to make li
0                    0.695
1                    0.699
2                    0.718

   Perceptions of corruption  Dystopia + resic
0                      0.546               2.
1                      0.548               1.
2                      0.182               2.
  Country name  score  Log GDP per capita  Soc
1      Denmark  7.583               1.908
2      Iceland  7.525               1.881
0      Finland  7.741               1.844

   Healthy life expectancy  Freedom to make li
1                    0.699
2                    0.718
0                    0.695

   Perceptions of corruption  Dystopia + resic
1                      0.548               1.
2                      0.182               2.
0                      0.546               2.
```

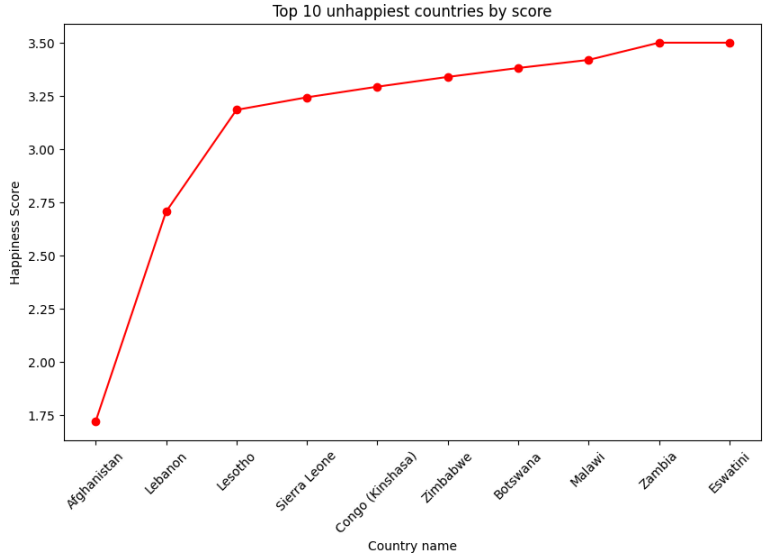
```
#• Adding New Columns:
#1. Create a new column called Happiness Category
# based on their Score:

#Low - (Score < 4)
#Medium - (4 ≤ Score ≤ 6)
#High - (Score > 6)

# Load the dataset
dataset = pd.read_csv('/content/drive/MyDrive/Copy

# Add 'Happiness Category' column
dataset['Happiness Category'] = ['Low' if x < 4 e

# Display the result
print(dataset[['Country name', 'score', 'Happiness
```

```
          Country name  score Happiness Categor
0              Finland  7.741               Hig
1              Denmark  7.583               Hig
2              Iceland  7.525               Hig
3               Sweden  7.344               Hig
4               Israel  7.341               Hig
..                 ...    ...                ..
```

```
      138  Congo (Kinshasa)  3.295                Lc
      139      Sierra Leone  3.245                Lc
      140           Lesotho  3.186                Lc
      141           Lebanon  2.707                Lc
      142       Afghanistan  1.721                Lc

      [143 rows x 3 columns]
```

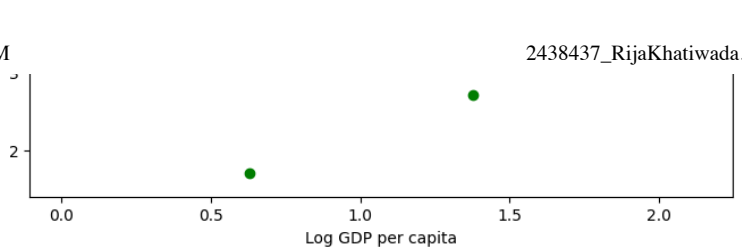
```
#2. Data Visualizations:
#• Bar Plot: Plot the top 10 happiest countries by
import matplotlib.pyplot as plt
dataset = pd.read_csv('/content/drive/MyDrive/Copy
# Bar Plot: Top 10 happiest countries
Top_10_Highest = dataset.nlargest(10, 'score')
plt.figure(figsize=(10, 6))
plt.bar(Top_10_Highest ['Country name'], Top_10_Hig
plt.xticks(rotation=45)
plt.title('Top 10 happiest countries by score:')
plt.xlabel('Country name')
plt.ylabel('Happiness Score')
plt.show()
#• Line Plot: Plot the top 10 unhappiest countries
Top_10_Unhappiest = dataset.nsmallest(10, 'score')
plt.figure(figsize=(10, 6))
plt.plot(Top_10_Unhappiest['Country name'], Top_10_
plt.xticks(rotation=45)
plt.title('Top 10 unhappiest countries by score')
plt.xlabel('Country name')
plt.ylabel('Happiness Score')
plt.show()
#• Plot a histogram for the Score column to show it
plt.figure(figsize=(8, 5))
plt.hist(dataset['score'], bins=15, color='purple',
plt.title('Distribution of Happiness Scores')
plt.xlabel('Happiness Score')
plt.ylabel('Frequency')
plt.show()
#• Scatter Plot: Plot a scatter plot between GDP pe
plt.figure(figsize=(8, 6))
plt.scatter(dataset['Log GDP per capita'], dataset[
plt.title('Relationship between  Log GDP per capita
plt.xlabel('Log GDP per capita')
plt.ylabel('Happiness Score')
plt.show()
```


Top 10 happiest countries by score:
Happiness Score
Finland
Denmark
Iceland
Sweden
Israel
Netherlands
Norway
Luxembourg
Switzerland
Australia
Country name


Top 10 unhappiest countries by score
Happiness Score
Afghanistan
Lebanon
Lesotho
Sierra Leone
Congo (Kinshasa)
Zimbabwe
Botswana
Malawi
Zambia
Eswatini
Country name


Distribution of Happiness Scores
Frequency
Happiness Score


Relationship between Log GDP per capita and Happiness Score
Happiness Score


2
0.0
0.5
1.0
1.5
2.0
Log GDP per capita

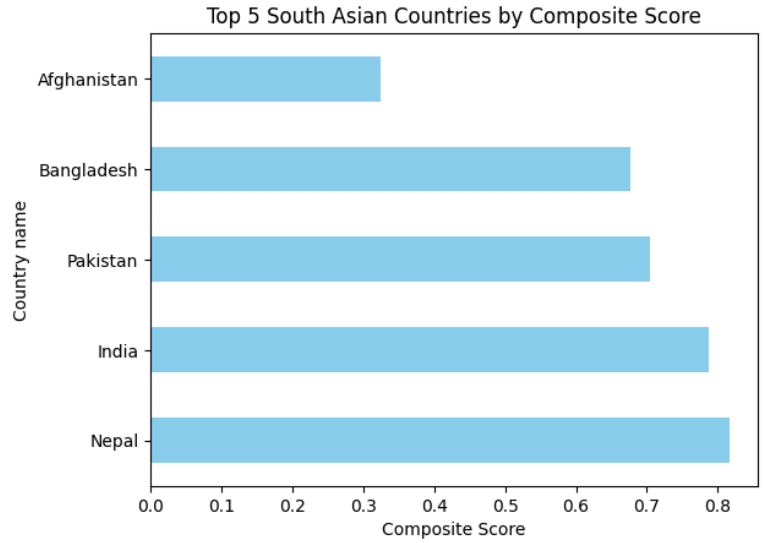
```
#Problem - 2 - Some Advance Data Exploration Task:
#Task - 1 - Setup Task - Preparing the South-Asia
#Steps:
#1. Define the countries in South Asia with a list
#south asian countries = ["Afghanistan", "Banglade
#"Maldives", "Nepal", "Pakistan", "Srilanka"]
dataset = pd.read_csv('/content/drive/MyDrive/Copy

# Filtering South Asian countries
South_Asian1 = ["Afghanistan", "Bangladesh", "Bhut
South_Asia_dataset1 = dataset[dataset['Country nar

print(South_Asia_dataset1)


#2. Use the list from step - 1 to filtered the dat

dataset = pd.read_csv('/content/drive/MyDrive/Copy
South_Asian2 = ["Afghanistan", "Bangladesh", "Bhut
print(dataset[dataset['Country name'].isin(South_A

#3. Save the filtered dataframe as separate CSV fi

dataset3 = pd.read_csv('/content/drive/MyDrive/Cop
South_Asian3 = ["Afghanistan", "Bangladesh", "Bhut
South_Asia_dataset3 = dataset[dataset['Country nar
South_Asia_dataset3.to_csv('/content/drive/MyDrive
```

```
    Country name  score  Log GDP per capita  S
92         Nepal  5.158               0.965
107     Pakistan  4.657               1.069
125        India  4.054               1.166
128   Bangladesh  3.886               1.122
142  Afghanistan  1.721               0.628

     Healthy life expectancy  Freedom to make
92                     0.443
107                    0.321
125                    0.417
128                    0.513
142                    0.242

     Perceptions of corruption  Dystopia + res
92                       0.115
107                      0.074
125                      0.122
128                      0.167
142                      0.088
    Country name  score  Log GDP per capita  S
92         Nepal  5.158               0.965
107     Pakistan  4.657               1.069
125        India  4.054               1.166
128   Bangladesh  3.886               1.122
```

```
      142  Afghanistan  1.721                0.628

           Healthy life expectancy  Freedom to make
      92                     0.443
      107                    0.321
      125                    0.417
      128                    0.513
      142                    0.242

           Perceptions of corruption  Dystopia + res
      92                       0.115
      107                      0.074
      125                      0.122
      128                      0.167
      142                      0.088


  #Task - 2 - Composite Score Ranking:
  #Tasks:
  #1. Using the SouthAsia DataFrame, create a new col
  #following metrics:Composite Score = 0.40 × GDP per

  dataset = pd.read_csv('/content/drive/MyDrive/Copy
  south_asian = ["Afghanistan", "Bangladesh", "Bhutan
  south_asia_dataset = dataset[dataset['Country name'

  south_asia_dataset['Composite Score'] = 0.40 * sout

  print(south_asia_dataset[['Country name', 'Composit

  #2. Rank the South Asian countries based on the Com

  dataset = pd.read_csv('/content/drive/MyDrive/Copy
  south_asian = ["Afghanistan", "Bangladesh", "Bhutan
  south_asia_dataset = dataset[dataset['Country name'

  south_asia_dataset['Composite Score'] = 0.40 * sout

  south_asia_dataset = south_asia_dataset.sort_values

  print(south_asia_dataset[['Country name', 'Composit

  #3. Visualize the top 5 countries using a horizonta

  dataset = pd.read_csv('/content/drive/MyDrive/Copy
  south_asian = ["Afghanistan", "Bangladesh", "Bhutan

  south_asia_dataset = dataset[dataset['Country name'

  south_asia_dataset['Composite Score'] = 0.40 * sout

  Top_5 = south_asia_dataset.nlargest(5, 'Composite S

  Top_5.plot.barh(x='Country name', y='Composite Scor
  plt.title('Top 5 South Asian Countries by Composite
  plt.xlabel('Composite Score')
  plt.show()
```

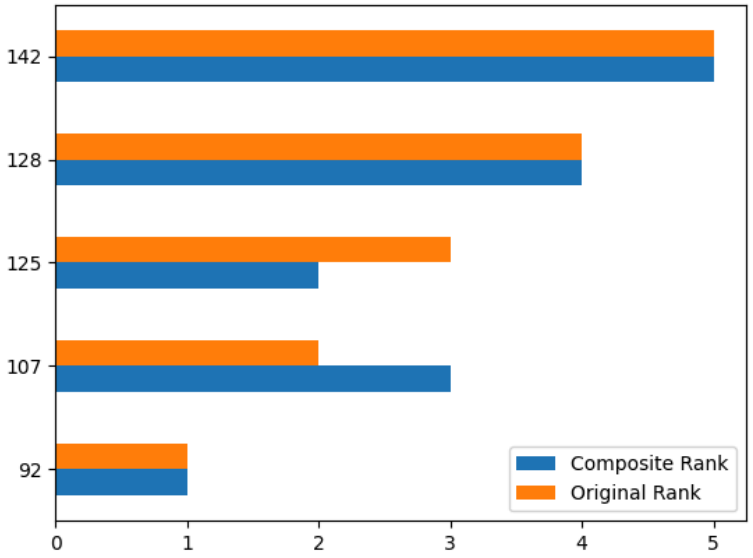
```
#4. Discuss whether the rankings based on the Compo

dataset = pd.read_csv('/content/drive/MyDrive/Copy
south_asian = ["Afghanistan", "Bangladesh", "Bhutan

south_asia_dataset = dataset[dataset['Country name'

south_asia_dataset['Composite Score'] = 0.40 * sout

south_asia_dataset['Composite Rank'] = south_asia_d
south_asia_dataset['Original Rank'] = south_asia_da

south_asia_dataset[['Composite Rank', 'Original Ran
plt.show()
```

```
<ipython-input-48-b03a2d992208>:10: SettingWit
A value is trying to be set on a copy of a sli
Try using .loc[row_indexer,col_indexer] = valu

See the caveats in the documentation: https://
  south_asia_dataset['Composite Score'] = 0.40
<ipython-input-48-b03a2d992208>:20: SettingWit
A value is trying to be set on a copy of a sli
Try using .loc[row_indexer,col_indexer] = valu

See the caveats in the documentation: https://
  south_asia_dataset['Composite Score'] = 0.40
<ipython-input-48-b03a2d992208>:33: SettingWit
A value is trying to be set on a copy of a sli
Try using .loc[row_indexer,col_indexer] = valu

See the caveats in the documentation: https://
  south_asia_dataset['Composite Score'] = 0.40
    Country name  Composite Score
92         Nepal           0.8159
107     Pakistan           0.7039
125        India           0.7874
128   Bangladesh           0.6774
142  Afghanistan           0.3238
    Country name  Composite Score
92         Nepal           0.8159
125        India           0.7874
107     Pakistan           0.7039
128   Bangladesh           0.6774
142  Afghanistan           0.3238
```




```
<ipython-input-48-b03a2d992208>:49: SettingWit
A value is trying to be set on a copy of a sli
Try using .loc[row_indexer,col_indexer] = valu

See the caveats in the documentation: https://
  south_asia_dataset['Composite Score'] = 0.40
<ipython-input-48-b03a2d992208>:51: SettingWit
A value is trying to be set on a copy of a sli
Try using .loc[row_indexer,col_indexer] = valu

See the caveats in the documentation: https://
  south_asia_dataset['Composite Rank'] = south
<ipython-input-48-b03a2d992208>:52: SettingWit
A value is trying to be set on a copy of a sli
```

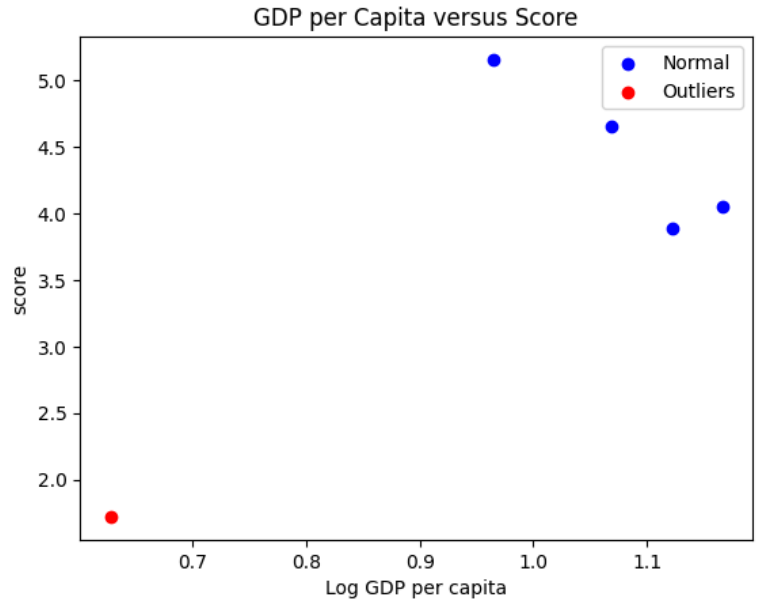
```
Try using .loc[row_indexer,col_indexer] = valu

See the caveats in the documentation: https://
  south_asia_dataset['Original Rank'] = south_
```



```
#Task - 3 - Outlier Detection:
#Tasks:
#1. Identify outlier countries in South Asia basec

dataset = pd.read_csv('/content/drive/MyDrive/Copy
south_asian = ["Afghanistan", "Bangladesh", "Bhuta

south_asia_dataset = dataset[dataset['Country name

def detect_outliers(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    return df[(df[column] < lower_bound) | (df[co

#detecting outliers based on gdp and score
Outliers_Score = detect_outliers(south_asia_datase
Outliers_Gdp = detect_outliers(south_asia_dataset,

Outliers = pd.concat([Outliers_Score, Outliers_Gdp


#Displaying the result
print(Outliers[['Country name', 'score', 'Log GDP

#2. Define outliers using the 1.5 × IQR rule.

dataset = pd.read_csv('/content/drive/MyDrive/Copy
South_Asian = ["Afghanistan", "Bangladesh", "Bhuta

South_Asia_dataset = dataset[dataset['Country name

def detect_outliers(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    return df[(df[column] < Q1 - 1.5 * IQR) | (df

Outliers_Score = detect_outliers(south_asia_datase
Outliers_Gdp = detect_outliers(south_asia_dataset,

Outliers = pd.concat([Outliers_Score, Outliers_Gdp

print(Outliers[['Country name', 'score', 'Log GDP


#3. Create a scatter plot with GDP per Capita on t

dataset = pd.read_csv('/content/drive/MyDrive/Copy
South_Asian = ["Afghanistan", "Bangladesh", "Bhuta

South_Asia_Dataset = dataset[dataset['Country name
```

```
def detect_outliers(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    return df[(df[column] < Q1 - 1.5 * IQR) | (df

Outliers = pd.concat([detect_outliers(south_asia_c
Normal = South_Asia_dataset[~South_Asia_dataset['C

plt.scatter(Normal['Log GDP per capita'], Normal[
plt.scatter(Outliers['Log GDP per capita'], Outlie
plt.xlabel('Log GDP per capita')
plt.ylabel('score')
plt.title('GDP per Capita versus Score')
plt.legend()
plt.show()

#4. Discuss the characteristics of these outliers
```

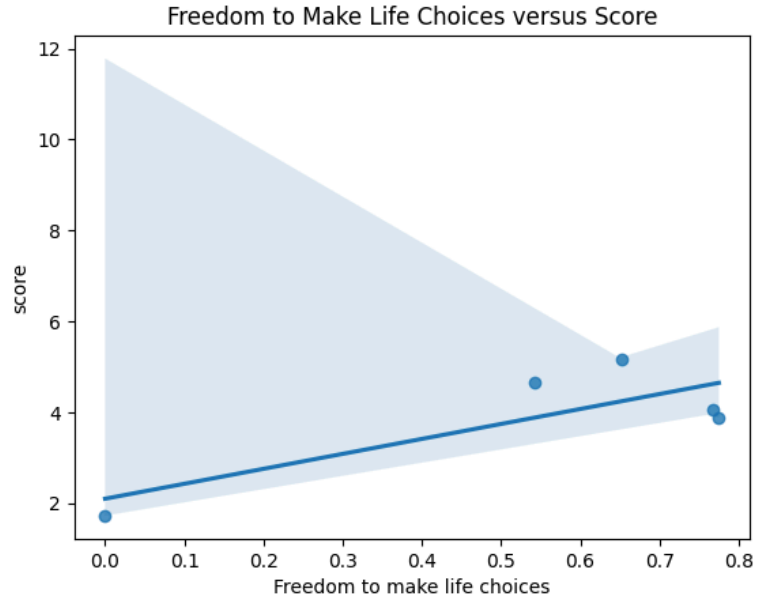
```
    Country name  score  Log GDP per capita
142  Afghanistan  1.721               0.628
    Country name  score  Log GDP per capita
142  Afghanistan  1.721               0.628
```

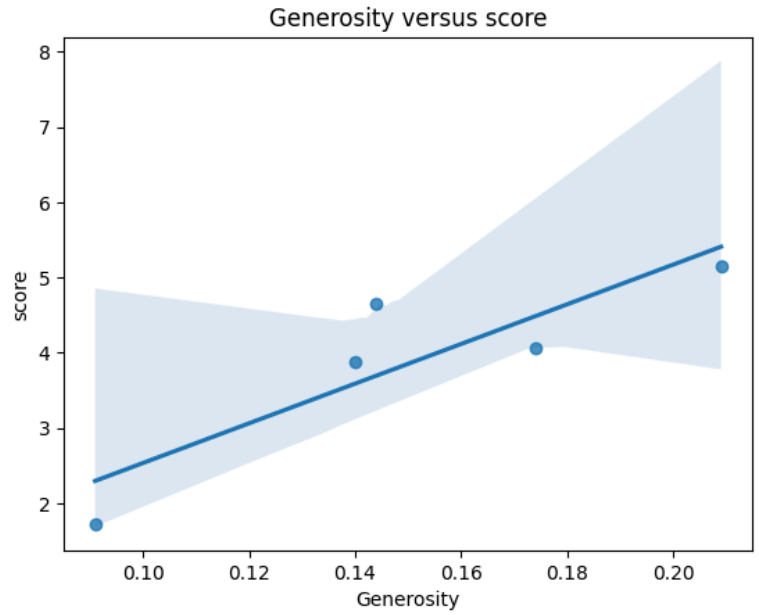



```
#Task - 4 - Exploring Trends Across Metrics:
#Tasks:
##1. Choose two metrics (e.g., Freedom to Make Li
#{pearson correlation} with the Score for South As
```

```
dataset = pd.read_csv('/content/drive/MyDrive/Copy
South_Asian = ["Afghanistan", "Bangladesh", "Bhuta

South_Asia_Dataset = dataset[dataset['Country name

Correlation_Freedom = South_Asia_dataset['score'].
Correlation_Generosity = South_Asia_dataset['score

print(Correlation_Freedom)
print(Correlation_Generosity)

#2. Create scatter plots with trendlines for these
import seaborn as sns

dataset = pd.read_csv('/content/drive/MyDrive/Copy
South_Asian = ["Afghanistan", "Bangladesh", "Bhuta

South_Asia_dataset = dataset[dataset['Country name

sns.regplot(x='Freedom to make life choices', y='s
plt.title('Freedom to Make Life Choices versus Sco
plt.show()

sns.regplot(x='Generosity', y='score', data=South_
plt.title('Generosity versus score')
plt.show()


#3. Identify and discuss the strongest and weakest
```

```
0.801497903141921
0.8773326267276358
```



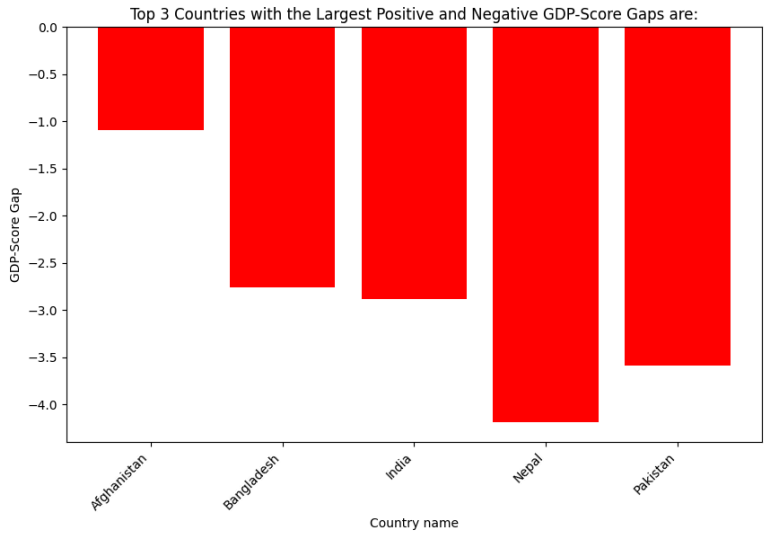



```
#Task - 5 - Gap Analysis:
#Tasks:
#1. Add a new column, GDP-Score Gap, which is the
```

```
#Loading dataset
dataset = pd.read_csv('/content/drive/MyDrive/Copy

# Countries list
South_Asian = ["Afghanistan", "Bangladesh", "Bhuta

# Countries list
South_Asia_dataset = dataset[dataset['Country name

# Adding gdp score column
South_Asia_dataset['GDP-Score Gap'] = south_asia_(

# Displaying the uploaded dataset
print(South_Asia_dataset[['Country name', 'Log GDI

#2. Rank the South Asian countries by this gap in

#Loading dataset
dataset = pd.read_csv('/content/drive/MyDrive/Copy

# Countries list
South_Asian = ["Afghanistan", "Bangladesh", "Bhuta

# Countries list
South_Asia_dataset = dataset[dataset['Country name

# Adding gdp score column
South_Asia_dataset['GDP-Score Gap'] = South_Asia_(

# Ranking  by GDP-Score Gap in ascending order
Ascending_Rank = South_Asia_dataset.sort_values('(

# Ranking  by GDP-Score Gap in descending order
Descending_Rank = South_Asia_dataset.sort_values(

# Displaying the results
print("Ascending Rank:")
print(Ascending_Rank[['Country name', 'GDP-Score (

print("\nDescending Rank:")
print(Descending_Rank[['Country name', 'GDP-Score

#3. Highlight the top 3 countries with the largest

dataset = pd.read_csv('/content/drive/MyDrive/Copy
South_Asian = ["Afghanistan", "Bangladesh", "Bhuta

South_Asia_dataset = dataset[dataset['Country name
South_Asia_dataset['GDP-Score Gap'] = South_Asia_(

Top_Positive_Gap = South_Asia_dataset.sort_values(
Top_Negative_Gap = South_Asia_dataset.sort_values(

Top_Gaps = pd.concat([Top_Positive_Gap, Top_Negati
```

```
plt.figure(figsize=(10, 6))
plt.bar(Top_Gaps['Country name'], Top_Gaps['GDP-Sc
plt.xlabel('Country name')
plt.ylabel('GDP-Score Gap')
plt.title('Top 3 Countries with the Largest Positi
plt.xticks(rotation=45, ha='right')
plt.show()

#4. Analyze the reasons behind these gaps and the
```

```
<ipython-input-66-30292651331c>:15: SettingWit
A value is trying to be set on a copy of a sli
Try using .loc[row_indexer,col_indexer] = valu

See the caveats in the documentation: https://
  South_Asia_dataset['GDP-Score Gap'] = south_
<ipython-input-66-30292651331c>:32: SettingWit
A value is trying to be set on a copy of a sli
Try using .loc[row_indexer,col_indexer] = valu

See the caveats in the documentation: https://
  South_Asia_dataset['GDP-Score Gap'] = South_
<ipython-input-66-30292651331c>:53: SettingWit
A value is trying to be set on a copy of a sli
Try using .loc[row_indexer,col_indexer] = valu

See the caveats in the documentation: https://
  South_Asia_dataset['GDP-Score Gap'] = South_
    Country name  Log GDP per capita  score  G
92         Nepal                0.965  5.158
107     Pakistan                1.069  4.657
125        India                1.166  4.054
128   Bangladesh                1.122  3.886
142  Afghanistan                0.628  1.721
Ascending Rank:
    Country name  GDP-Score Gap
92         Nepal         -4.193
107     Pakistan         -3.588
125        India         -2.888
128   Bangladesh         -2.764
142  Afghanistan         -1.093

Descending Rank:
    Country name  GDP-Score Gap
142  Afghanistan         -1.093
128   Bangladesh         -2.764
125        India         -2.888
107     Pakistan         -3.588
92         Nepal         -4.193
```



```
#Problem - 3 - Comparative Analysis:
#Task - 1 - Setup Task - Preparing the Middle Easte
#Tasks:
#1. Similar in Task - 1 of Problem 2 create a dataf


Dataset = pd.read_csv('/content/drive/MyDrive/Copy

Middle_Eastern_Countries = ["Bahrain", "Iran", "Ira
                            "Kuwait", "Lebanon", "O

Middle_East_Dataset = dataset[dataset['Country name

print(Middle_East_Dataset)


#1. Descriptive Statistics:• Calculate the mean, St

Dataset = pd.read_csv('/content/drive/MyDrive/Copy

South_Asian = ["Afghanistan", "Bangladesh", "Bhutan
Middle_Eastern_Countries = ["Bahrain", "Iran", "Ira

South_Asia_Dataset = dataset[dataset['Country name'
Middle_East_Dataset = dataset[dataset['Country name

South_Asia_Mean = South_Asia_Dataset['score'].mean(
South_Asia_Std = South_Asia_Dataset['score'].std()

Middle_East_Mean = Middle_East_Dataset['score'].mea
Middle_East_Std = Middle_East_Dataset['score'].std(

print(f"South Asia - Mean Score: {South_Asia_Mean }
print(f"Middle East - Mean Score: {Middle_East_Mean

#• Which region has higher happiness Scores on aver
if South_Asia_Mean > Middle_East_Mean:
    print("South Asia has a higher average happines
else:
    print("Middle East has a higher average happine
```

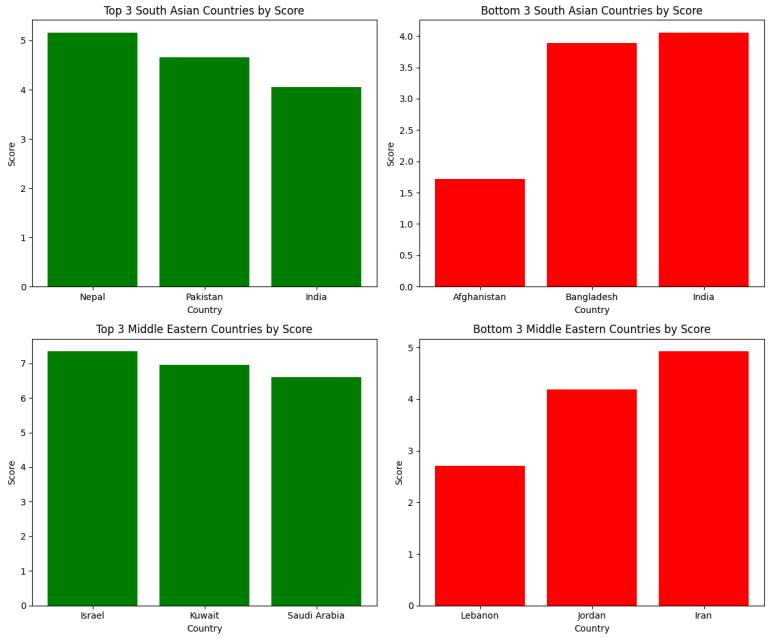
```
     Country name  score  Log GDP per capita
4          Israel  7.341               1.803
12         Kuwait  6.951               1.845
27   Saudi Arabia  6.594               1.842
```

```
61        Bahrain  5.959                NaN
91           Iraq  5.166              1.249
99           Iran  4.923              1.435
124        Jordan  4.186              1.262
141       Lebanon  2.707              1.377

     Healthy life expectancy  Freedom to make
4                      0.740
12                     0.661
27                     0.511
61                       NaN
91                     0.498
99                     0.571
124                    0.594
141                    0.556

     Perceptions of corruption  Dystopia + res
4                        0.193
12                       0.172
27                       0.188
61                         NaN
91                       0.048
99                       0.123
124                      0.189
141                      0.029               -
South Asia - Mean Score: 3.8952000000000004, S
Middle East - Mean Score: 5.478375, Standard D
Middle East has a higher average happiness scc


#2. Top and Bottom Performers:
#• Identify the top 3 and bottom 3 countries in ea


Dataset = pd.read_csv('/content/drive/MyDrive/Copy

South_Asian = ["Afghanistan", "Bangladesh", "Bhuta
Middle_Eastern_Countries = ["Bahrain", "Iran", "Ir

South_Asia_Dataset = Dataset[Dataset['Country name
Middle_East_Dataset = Dataset[Dataset['Country nar

# Top 3 and bottom 3 countries based on Score in S
Top_3_South_Asia = South_Asia_Dataset.sort_values(
Bottom_3_South_Asia = South_Asia_Dataset.sort_valu

# Top 3 and bottom 3 countries based on Score in M
Top_3_Middle_East = Middle_East_Dataset.sort_value
Bottom_3_Middle_East = Middle_East_Dataset.sort_va

# Displaying results
print("Top 3 South Asian countries based on Score:
print(Top_3_South_Asia[['Country name', 'score']])

print("\nBottom 3 South Asian countries based on S
print(Bottom_3_South_Asia[['Country name', 'score
```

```
print("\nTop 3 Middle Eastern countries based on S
print(Top_3_Middle_East[['Country name', 'score']]

print("\nBottom 3 Middle Eastern countries based o
print(Bottom_3_Middle_East[['Country name', 'score

#• Plot bar charts comparing these charts.
import pandas as pd
import matplotlib.pyplot as plt

Dataset = pd.read_csv('/content/drive/MyDrive/Copy

South_Asian = ["Afghanistan", "Bangladesh", "Bhuta
Middle_Eastern_Countries = ["Bahrain", "Iran", "Ir

South_Asia_Dataset = Dataset[Dataset['Country name
Middle_East_Dataset = Dataset[Dataset['Country nam

# Top 3 and bottom 3 countries based on Score in S
Top_3_South_Asia = South_Asia_Dataset.sort_values(
Bottom_3_South_Asia = South_Asia_Dataset.sort_valu

# Top 3 and bottom 3 countries based on Score in M
Top_3_Middle_East = Middle_East_Dataset.sort_value
Bottom_3_Middle_East = Middle_East_Dataset.sort_va

# Plotting bar charts comparing the scores
fig, axes = plt.subplots(2, 2, figsize=(12, 10))

# South Asia - Top 3
axes[0, 0].bar(Top_3_South_Asia['Country name'], T
axes[0, 0].set_title('Top 3 South Asian Countries
axes[0, 0].set_xlabel('Country')
axes[0, 0].set_ylabel('Score')

# South Asia - Bottom 3
axes[0, 1].bar(Bottom_3_South_Asia['Country name']
axes[0, 1].set_title('Bottom 3 South Asian Countri
axes[0, 1].set_xlabel('Country')
axes[0, 1].set_ylabel('Score')

# Middle East - Top 3
axes[1, 0].bar(Top_3_Middle_East['Country name'],
axes[1, 0].set_title('Top 3 Middle Eastern Countri
axes[1, 0].set_xlabel('Country')
axes[1, 0].set_ylabel('Score')

# Middle East - Bottom 3
axes[1, 1].bar(Bottom_3_Middle_East['Country name
axes[1, 1].set_title('Bottom 3 Middle Eastern Cour
axes[1, 1].set_xlabel('Country')
axes[1, 1].set_ylabel('Score')

plt.tight_layout()
```

```
plt.show()
```

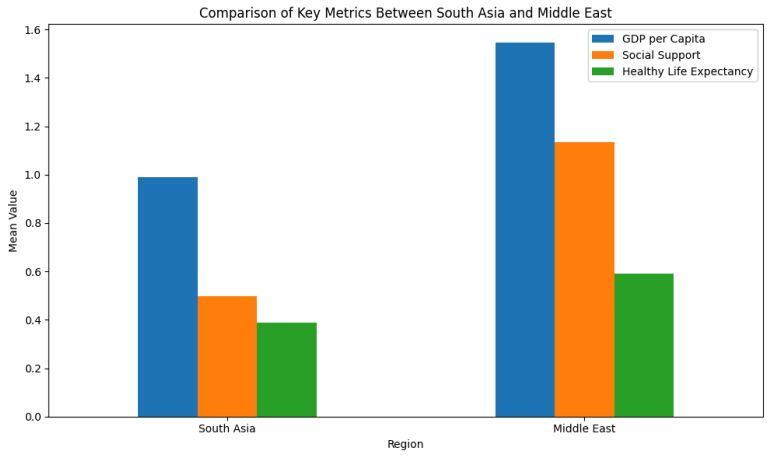
```
Top 3 South Asian countries based on Score:
    Country name  score
92         Nepal  5.158
107     Pakistan  4.657
125        India  4.054

Bottom 3 South Asian countries based on Score:
    Country name  score
142  Afghanistan  1.721
128   Bangladesh  3.886
125        India  4.054

Top 3 Middle Eastern countries based on Score:
    Country name  score
4         Israel  7.341
12        Kuwait  6.951
27  Saudi Arabia  6.594

Bottom 3 Middle Eastern countries based on Scc
    Country name  score
141      Lebanon  2.707
124       Jordan  4.186
99          Iran  4.923
```



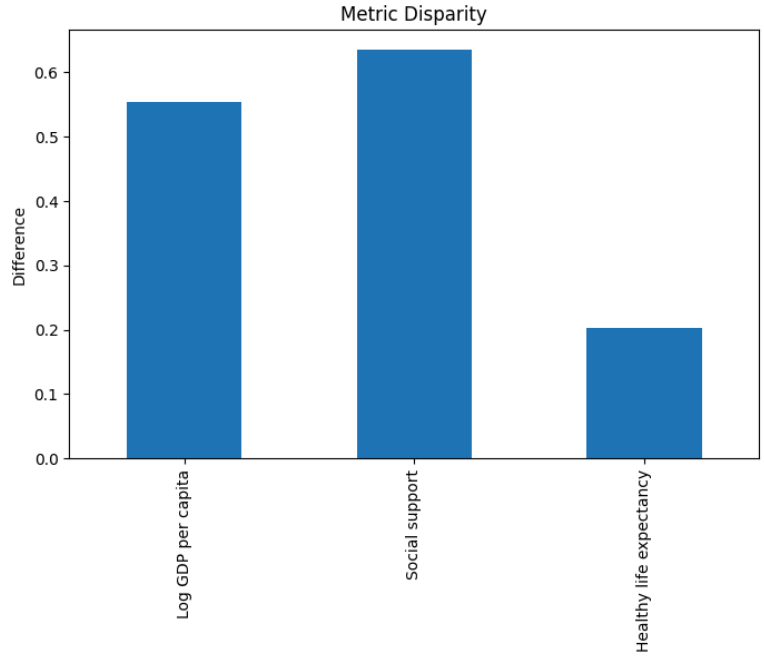

```
#3. Metric Comparisons:
```

```
#• Compare key metrics like GDP per capita, Social

import pandas as pd
import matplotlib.pyplot as plt

Dataset = pd.read_csv('/content/drive/MyDrive/Copy

South_Asian = ["Afghanistan", "Bangladesh", "Bhutan
Middle_Eastern_Countries = ["Bahrain", "Iran", "Ira

South_Asia_Dataset = Dataset[Dataset['Country name'
Middle_East_Dataset = Dataset[Dataset['Country name

# Remove any non-numeric or missing values for the
South_Asia_Dataset = South_Asia_Dataset[['Country n
Middle_East_Dataset = Middle_East_Dataset[['Country

# Calculate the mean values for each metric
south_asia_means = South_Asia_Dataset[['Log GDP per
middle_east_means = Middle_East_Dataset[['Log GDP p

# Create the comparison dataframe
comparison_df = pd.DataFrame({
    'Region': ['South Asia', 'Middle East'],
    'GDP per Capita': [south_asia_means['Log GDP pe
    'Social Support': [south_asia_means['Social sup
    'Healthy Life Expectancy': [south_asia_means['H
})

# Plotting the grouped bar chart
comparison_df.set_index('Region').plot(kind='bar',

plt.title('Comparison of Key Metrics Between South
plt.ylabel('Mean Value')
plt.xlabel('Region')
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()

#• Which metrics show the largest disparity between

Dataset = pd.read_csv('/content/drive/MyDrive/Copy

South_Asian = ["Afghanistan", "Bangladesh", "Bhutan
Middle_Eastern_Countries = ["Bahrain", "Iran", "Ira

South_Asia_Dataset = Dataset[Dataset['Country name'
Middle_East_Dataset = Dataset[Dataset['Country name

south_asia_means = South_Asia_Dataset[['Log GDP per
middle_east_means = Middle_East_Dataset[['Log GDP p

difference = abs(south_asia_means - middle_east_mea
largest_disparity_metric = difference.idxmax()

print(f"Largest disparity: {largest_disparity_metri
```

```
print("Largest disparity:", largest_disparity_metric)

difference.plot(kind='bar', figsize=(8, 5))
plt.title('Metric Disparity')
plt.ylabel('Difference')
plt.show()
```




```
Largest disparity: Social support with a diffe
```




```
#4. Happiness Disparity:• Compute the range (max -
Dataset = pd.read_csv('/content/drive/MyDrive/Copy
```

```
South_Asian = ["Afghanistan", "Bangladesh", "Bhuta
Middle_Eastern_Countries = ["Bahrain", "Iran", "Ir

South_Asia = Dataset[Dataset['Country name'].isin(
Middle_East = Dataset[Dataset['Country name'].isir

south_asia_range = South_Asia['score'].max() - Sou
middle_east_range = Middle_East['score'].max() - M

south_asia_cv = South_Asia['score'].std() / South_
middle_east_cv = Middle_East['score'].std() / Midd

print(f"South Asia - Range: {south_asia_range:.2f}
print(f"Middle East - Range: {middle_east_range:.2

#• Which region has greater variability in happine
if south_asia_cv > middle_east_cv:
    print("South Asia has greater variability in H
else:
    print("Middle East has greater variability in
```

```
South Asia - Range: 3.44, CV: 33.79%
Middle East - Range: 4.63, CV: 28.34%
South Asia has greater variability in happines
```

```
#5. Correlation Analysis:• Analyze the correlation

Dataset = pd.read_csv('/content/drive/MyDrive/Copy

South_Asian = ["Afghanistan", "Bangladesh", "Bhutan
Middle_Eastern_Countries = ["Bahrain", "Iran", "Ira

South_Asia = Dataset[Dataset['Country name'].isin(S
Middle_East = Dataset[Dataset['Country name'].isin(

south_asia_corr = South_Asia[['score', 'Freedom to
middle_east_corr = Middle_East[['score', 'Freedom t

print(south_asia_corr[['score']])
print(middle_east_corr[['score']])

#• Create scatter plots to visualize and interpret


Dataset = pd.read_csv('/content/drive/MyDrive/Copy

South_Asian = ["Afghanistan", "Bangladesh", "Bhutan
Middle_Eastern_Countries = ["Bahrain", "Iran", "Ira

South_Asia = Dataset[Dataset['Country name'].isin(S
Middle_East = Dataset[Dataset['Country name'].isin(

plt.figure(figsize=(12, 6))
```