```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

3.1 Problem - 1: Getting Started with Data Exploration - Some Warm up Exercises:

1. Data Exploration and Understanding:

Dataset Overview:

1. Load the dataset and display the first 10 rows.
2. Identify the number of rows and columns in the dataset.
3. List all the columns and their data types.

a. Importing and loading dataset

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

data = pd.read_csv("/content/drive/MyDrive/Modules_Year_2/Concepts-and-Technologies-of-AI/Dataset/Assignment-1_WHR-2024-5CS037.csv")
```

b. Display the first 10 rows.

```
data.head(10)
```

	Country name	score	Log GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption	Dystopia + residual
0	Finland	7.741	1.844	1.572	0.695	0.859	0.142	0.546	2.082
1	Denmark	7.583	1.908	1.520	0.699	0.823	0.204	0.548	1.881
2	Iceland	7.525	1.881	1.617	0.718	0.819	0.258	0.182	2.050
3	Sweden	7.344	1.878	1.501	0.724	0.838	0.221	0.524	1.658
4	Israel	7.341	1.803	1.513	0.740	0.641	0.153	0.193	2.298
5	Netherlands	7.319	1.901	1.462	0.706	0.725	0.247	0.372	1.906
6	Norway	7.302	1.952	1.517	0.704	0.835	0.224	0.484	1.586
7	Luxembourg	7.122	2.141	1.355	0.708	0.801	0.146	0.432	1.540
8	Switzerland	7.060	1.970	1.425	0.747	0.759	0.173	0.498	1.488
9	Australia	7.057	1.854	1.461	0.692	0.756	0.225	0.323	1.745

c. Number of rows and columns in the dataset.

```
print(f"Rows: {data.shape[0]}, Columns: {data.shape[1]}")
```

```
Rows: 143, Columns: 9
```

d. Data Types

```
print(data.dtypes)
```

```
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 deviation for the Score column.

2. Identify the country with the highest and lowest happiness scores.

```
mean = data['score'].mean()
median = data['score'].median()
std = data['score'].std()
print(f" Mean: {mean},\n Median: {median},\n Std: {std}")
```

```
 Mean: 5.52758041958042,
 Median: 5.785,
 Std: 1.1707165099442995
```

```
highest = data.loc[data['score'].idxmax(), 'Country name']
lowest = data.loc[data['score'].idxmin(), 'Country name']
print(f" Highest Happiness Score: {highest},\n Lowest Happiness Score: {lowest}")
```

```
 Highest Happiness Score: Finland,
 Lowest Happiness Score: Afghanistan
```

Missing Values:

1. Check if there are any missing values in the dataset. If so, display the total count for each column.

```
print(data.isnull().sum())
```

```
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
```

Filtering and Sorting:

1. Filter the dataset to show only the countries with a Score greater than 7.5.
2. For the filtered dataset - Sort the dataset by GDP per Capita in descending order and display the top 10 rows.

```
filtered = data[data['score'] > 7.5]
sorted_filtered = filtered.sort_values(by="Log GDP per capita", ascending=False).head(10)
sorted_filtered
```

	Country name	score	Log GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption	Dystopia + residual
1	Denmark	7.583	1.908	1.520	0.699	0.823	0.204	0.548	1.881
2	Iceland	7.525	1.881	1.617	0.718	0.819	0.258	0.182	2.050
0	Finland	7.741	1.844	1.572	0.695	0.859	0.142	0.546	2.082

Adding New Columns:

1. Create a new column called Happiness Category that categorizes countries into three categories based on their Score:

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

```
def categorize(score):
    if score < 4:
        return "Low"
    elif 4 <= score <= 6:
        return "Medium"
    else:
        return "High"
data['Happiness Category'] = data['score'].apply(categorize)
data[['Country name', 'score', 'Happiness Category']]
```

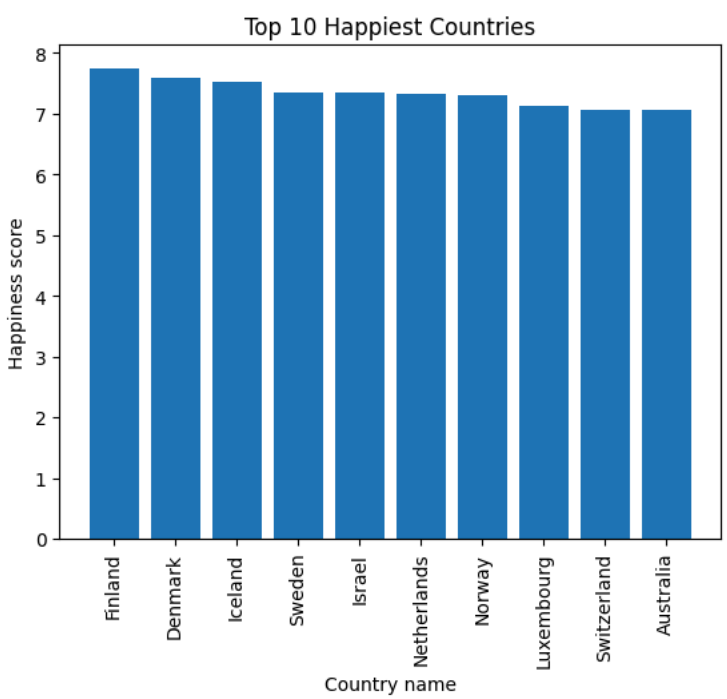
	Country name	score	Happiness Category
0	Finland	7.741	High
1	Denmark	7.583	High
2	Iceland	7.525	High
3	Sweden	7.344	High
4	Israel	7.341	High
...	...	...	...
138	Congo (Kinshasa)	3.295	Low
139	Sierra Leone	3.245	Low
140	Lesotho	3.186	Low
141	Lebanon	2.707	Low
142	Afghanistan	1.721	Low

143 rows × 3 columns

2. Data Visualizations:

- Bar Plot - Top 10 Happiest Countries:

```
top10 = data.nlargest(10, 'score')
plt.bar(top10['Country name'], top10['score'])
plt.title("Top 10 Happiest Countries")
plt.xlabel("Country name")
plt.ylabel("Happiness score")
plt.xticks(rotation=90)
plt.show()
```

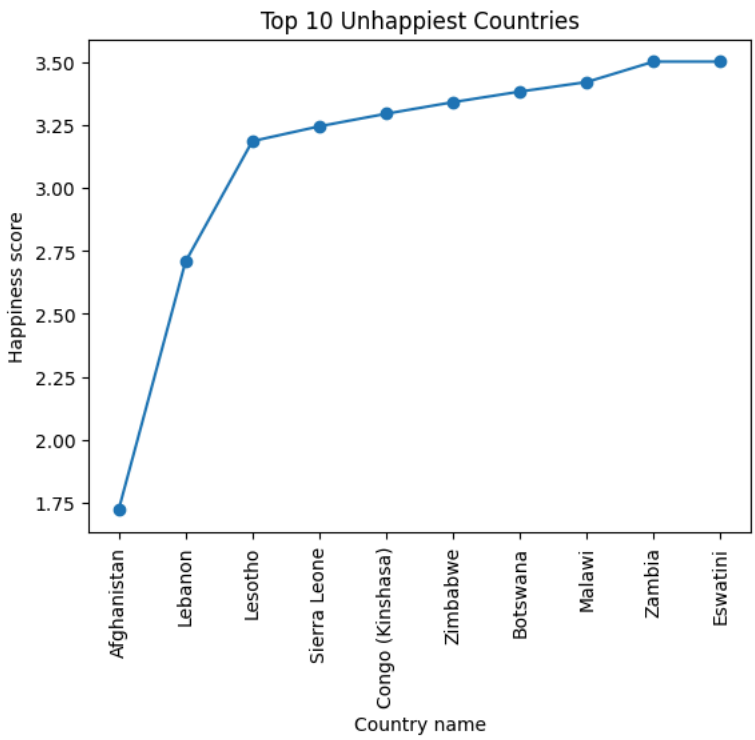


The chart illustrates the happiness scores of the top 10 happiest countries. At the top of the list is Finland, which boasts the highest happiness score. Following Finland are Denmark, Iceland, Sweden, Israel, the Netherlands, Norway, Luxembourg, Switzerland, and Australia. Each of these countries has a happiness score close to 8, indicating high levels of contentment among their citizens. This chart highlights the regions of the world where people experience the greatest overall happiness, which is valuable for understanding social well-being and public policy impacts.

- Line Plot - Top 10 Unhappiest Countries:

```
bottom10 = data.nsmallest(10, 'score')
plt.plot(bottom10['Country name'], bottom10['score'], marker='o')
plt.title("Top 10 Unhappiest Countries")
plt.xlabel("Country name")
plt.ylabel("Happiness score")
```

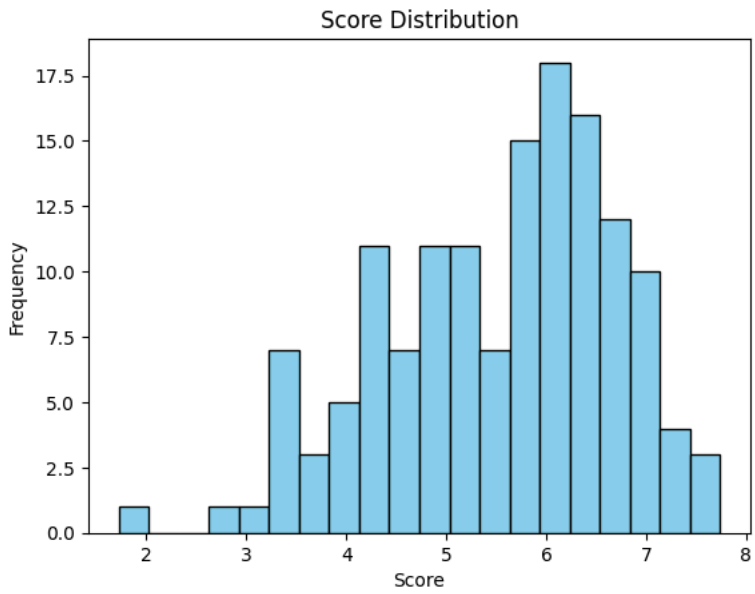
```
plt.xticks(rotation=90)
plt.show()
```



Insight: The plot ranks the 10 unhappiest countries, with Afghanistan having the lowest happiness score (around 1.75). The scores gradually increase, ranging from 2.75 to 3.5, showing similar levels of unhappiness among the other countries.

Bar Chart - Score Distribution

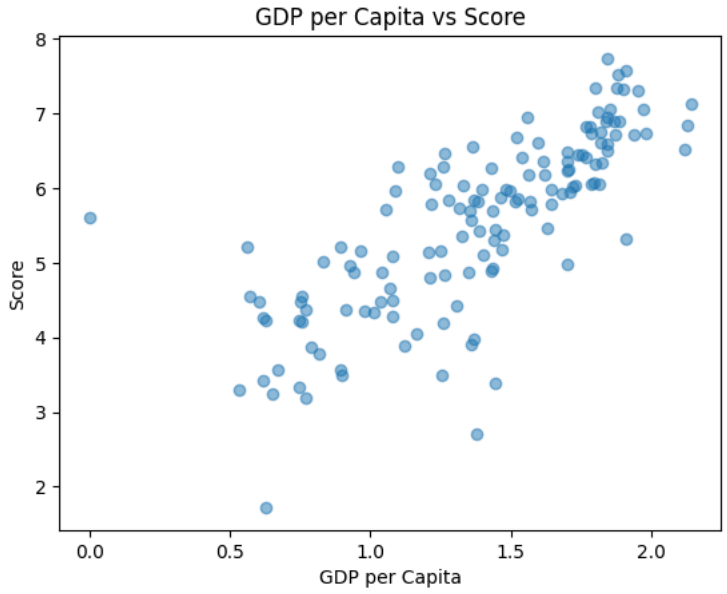
```
plt.hist(data['score'], bins=20, color='skyblue', edgecolor='black')
plt.title("Score Distribution")
plt.xlabel("Score")
plt.ylabel("Frequency")
plt.show()
```



Insight: The histogram shows that scores are roughly normally distributed, with most values concentrated between 5 and 7. The most common score is around 6, while lower and higher scores are less frequent, indicating a moderate to high overall trend.

Dot Chart - GDP per Capita vs Score

```
plt.scatter(data['Log GDP per capita'], data['score'], alpha=0.5)
plt.title("GDP per Capita vs Score")
plt.xlabel("GDP per Capita")
plt.ylabel("Score")
plt.show()
```



Insight: This scatter plot, titled "GDP per Capita vs Score," illustrates the relationship between GDP per capita (x-axis) and a score (y-axis), likely representing an outcome such as quality of life, happiness, or a similar metric. The plot shows a clear positive correlation, where higher GDP per capita tends to correspond with higher scores. Most data points are clustered in the middle-to-upper ranges, indicating that countries or entities with higher economic wealth generally achieve better scores. However, there are some outliers, where either high GDP does not result in a high score or a low GDP aligns with a relatively high score. This suggests that while economic wealth is a significant factor, other variables may also influence the score. The scatter of points at lower values also reflects variability in the data, hinting at complexities beyond GDP alone.

3.2 Problem - 2 - Some Advance Data Exploration Task:

1. **Task - 1 - Setup Task - Preparing the South-Asia Dataset:**

- Steps:

1. Define the countries in South Asia with a list for example: south asian countries = ["Afghanistan", "Bangladesh", "Bhutan", "India", "Maldives", "Nepal", "Pakistan", "Srilanka"]
2. Use the list from step - 1 to filtered the dataset {i.e. filtered out matching dataset from list.}
3. Save the filtered dataframe as separate CSV files for future use.

```
south_asian_countries = ["Afghanistan", "Bangladesh", "Bhutan", "India",
                         "Maldives", "Nepal", "Pakistan", "Sri Lanka"]
south_asia = data[data['Country name'].isin(south_asian_countries)]
south_asia.to_csv("SouthAsia.csv", index=False)
south_asia
```

	Country name	score	Log GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption	Dystopia + residual	Happiness Category
92	Nepal	5.158	0.965	0.990	0.443	0.653	0.209	0.115	1.783	Medium
107	Pakistan	4.657	1.069	0.600	0.321	0.542	0.144	0.074	1.907	Medium
125	India	4.054	1.166	0.653	0.417	0.767	0.174	0.122	0.756	Medium
127	Sri Lanka	3.898	1.361	1.179	0.586	0.583	0.144	0.031	0.014	Low
128	Bangladesh	3.886	1.122	0.249	0.513	0.775	0.140	0.167	0.919	Low

2. **Task - 2 - Composite Score Ranking:**

Tasks:

1. Using the SouthAsia DataFrame, create a new column called Composite Score that combines the following metrics:

```
Composite Score = 0.40 × GDP per Capita + 0.30 × Social Support + 0.30 × Healthy Life Expectancy
```

```
south_asia['Composite Score'] = (0.40 * south_asia['Log GDP per capita'] +
                                 0.30 * south_asia['Social support'] +
                                 0.30 * south_asia['Healthy life expectancy'])

print("South Asia DataFrame with Composite Score:")
south_asia[['Country name', 'Log GDP per capita', 'Social support',
                  'Healthy life expectancy', 'Composite Score']]
```

```
South Asia DataFrame with Composite Score:
<ipython-input-20-f396746e5aee>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  south_asia['Composite Score'] = (0.40 * south_asia['Log GDP per capita'] +
```

	Country name	Log GDP per capita	Social support	Healthy life expectancy	Composite Score
92	Nepal	0.965	0.990	0.443	0.8159
107	Pakistan	1.069	0.600	0.321	0.7039
125	India	1.166	0.653	0.417	0.7874
127	Sri Lanka	1.361	1.179	0.586	1.0739
128	Bangladesh	1.122	0.249	0.513	0.6774
142	Afghanistan	0.628	0.000	0.242	0.3238

2. Rank the South Asian countries based on the Composite Score in descending order.

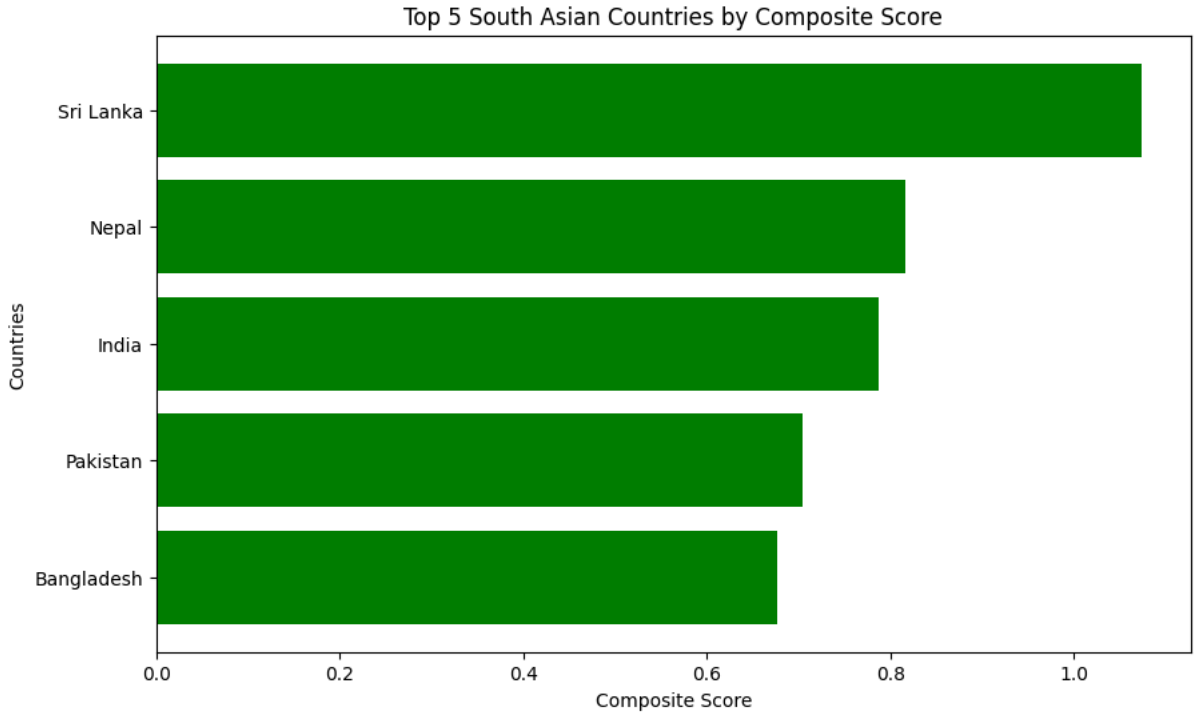
```
south_asia_sorted = south_asia.sort_values(by='Composite Score', ascending=False)

print("South Asian countries ranked by Composite Score:")
south_asia_sorted[['Country name', 'Composite Score']]
```

```
South Asian countries ranked by Composite Score:
```

	Country name	Composite Score
127	Sri Lanka	1.0739
92	Nepal	0.8159
125	India	0.7874
107	Pakistan	0.7039
128	Bangladesh	0.6774
142	Afghanistan	0.3238

3. Visualize the top 5 countries using a horizontal bar chart showing the Composite Score.

```
top5_composite = south_asia_sorted.head(5)

plt.figure(figsize=(10, 6))
plt.barh(top5_composite['Country name'], top5_composite['Composite Score'], color='green')
plt.xlabel("Composite Score")
plt.ylabel("Countries")
plt.title("Top 5 South Asian Countries by Composite Score")
plt.gca().invert_yaxis()
plt.show()
```



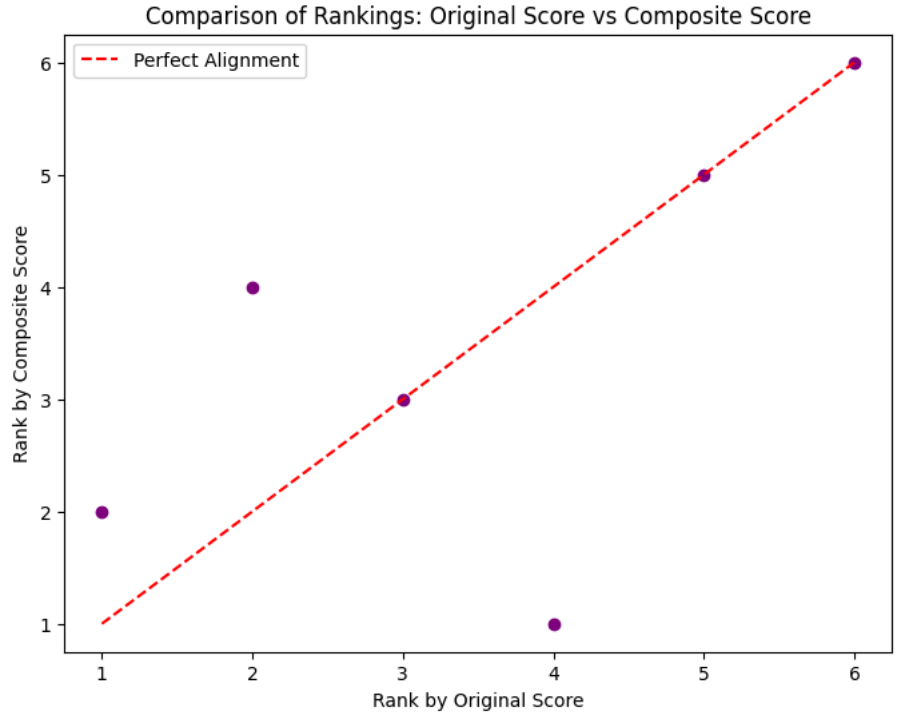
Insight: The bar chart displays the composite scores of the top 5 South Asian countries. Sri Lanka stands out with the highest composite score, exceeding 1.0. Following Sri Lanka, Nepal has the second-highest score, closely trailed by India and Pakistan, which have similar scores. Bangladesh rounds out the chart with the lowest composite score, just above 0.4. The green bars represent the scores, with their length corresponding to the composite scores of each country. The x-axis shows the composite score ranging from 0.0 to 1.2, while the y-axis lists the countries. This visual representation highlights the relative standings of these South Asian nations based on their composite scores, providing an insightful comparison.

4. Discuss whether the rankings based on the Composite Score align with the original Score - support your discussion with some visualization plot.

Double-click (or enter) to edit

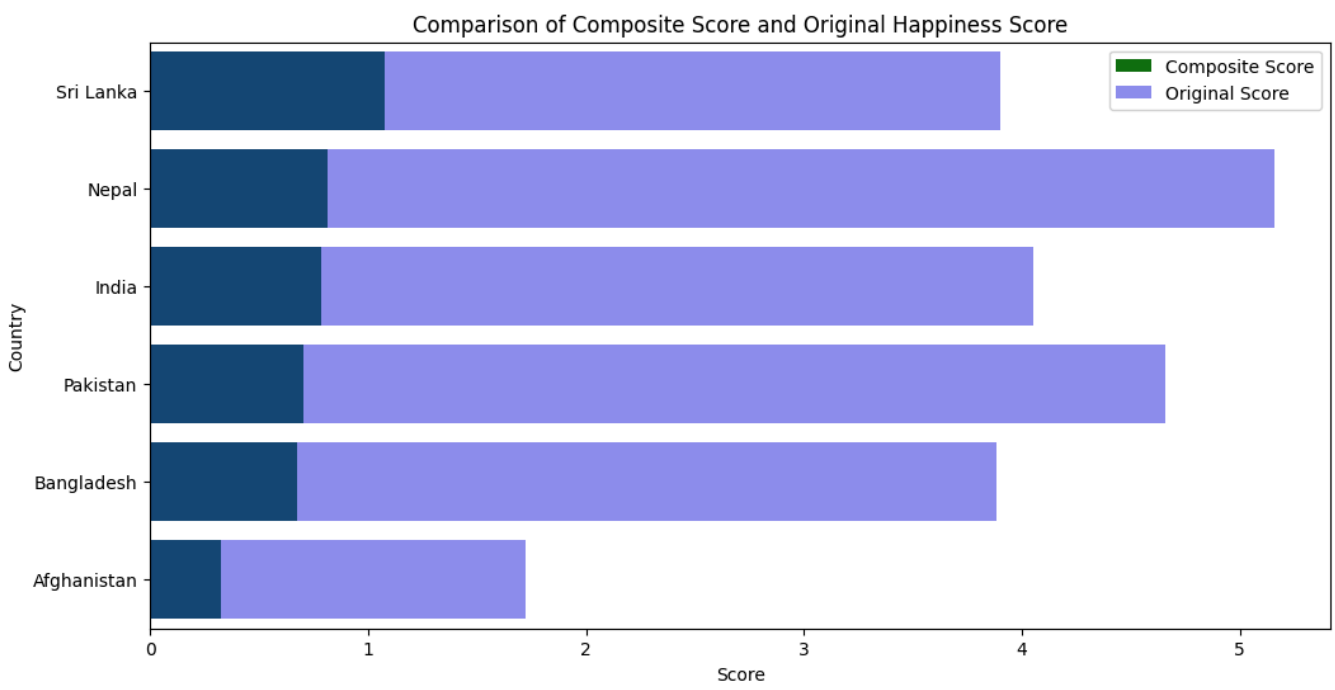
```
# Create a comparison DataFrame
comparison = south_asia_sorted[['Country name', 'score', 'Composite Score']]
comparison['Rank by Score'] = comparison['score'].rank(ascending=False)
comparison['Rank by Composite Score'] = comparison['Composite Score'].rank(ascending=False)

# Display the comparison table
print("Comparison of Original Score and Composite Score Rankings:")
print(comparison[['Country name', 'score', 'Composite Score', 'Rank by Score', 'Rank by Composite Score']])

# Scatter plot to compare rankings
plt.figure(figsize=(8, 6))
plt.scatter(comparison['Rank by Score'], comparison['Rank by Composite Score'], color='purple')
plt.plot([1, len(comparison)], [1, len(comparison)], linestyle='--', color='red', label='Perfect Alignment')
plt.xlabel("Rank by Original Score")
plt.ylabel("Rank by Composite Score")
plt.title("Comparison of Rankings: Original Score vs Composite Score")
plt.legend()
plt.show()
```

```
<ipython-input-23-577d5ceaeab3>:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  comparison['Rank by Score'] = comparison['score'].rank(ascending=False)
<ipython-input-23-577d5ceaeab3>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  comparison['Rank by Composite Score'] = comparison['Composite Score'].rank(ascending=False)
Comparison of Original Score and Composite Score Rankings:
    Country name  score  Composite Score  Rank by Score  \
127    Sri Lanka  3.898           1.0739            4.0
92         Nepal  5.158           0.8159            1.0
125        India  4.054           0.7874            3.0
107     Pakistan  4.657           0.7039            2.0
128   Bangladesh  3.886           0.6774            5.0
142  Afghanistan  1.721           0.3238            6.0

     Rank by Composite Score
127                      1.0
92                       2.0
125                      3.0
107                      4.0
128                      5.0
142                      6.0
```



Insight: The bar chart illustrates the composite scores of the top five South Asian countries. Sri Lanka is at the top with the highest composite score, surpassing 1.0. Nepal follows closely behind with the second-highest score, while India and Pakistan have similar scores, placing them in the middle of the chart. Bangladesh has the lowest composite score among the five countries, slightly above 0.4. The green bars visually represent the composite scores, with their lengths corresponding to each country's score. The x-axis indicates the composite score range from 0.0 to 1.2, and the y-axis lists the countries. This chart provides a comparative view of the composite scores of these South Asian nations, highlighting their relative standings.

```
# Plotting comparison of the Composite Score vs Original Score
plt.figure(figsize=(12, 6))
sns.barplot(x='Composite Score', y='Country name', data=south_asia_sorted, color='green', label='Composite Score')
sns.barplot(x='score', y='Country name', data=south_asia_sorted, color='blue', alpha=0.5, label='Original Score')

plt.title("Comparison of Composite Score and Original Happiness Score")
plt.xlabel("Score")
plt.ylabel("Country")
plt.legend()
plt.show()
```



Insight: The bar chart compares the Composite Score and the Original Happiness Score for six countries: Sri Lanka, Nepal, India, Pakistan, Bangladesh, and Afghanistan. The dark blue bars represent the Composite Score, while the light purple bars represent the Original Happiness Score. The y-axis lists the countries, and the x-axis shows the score range from 0 to 5.

For each country, the Original Happiness Score is higher than the Composite Score. Sri Lanka has the highest scores in both categories, followed by Nepal, India, Pakistan, Bangladesh, and Afghanistan. This chart provides a clear comparison, illustrating how the scores differ for each country.

3. **Task - 3 - Outlier Detection:**

Tasks:

1. Identify outlier countries in South Asia based on their Score and GDP per Capita.
2. Define outliers using the 1.5 × IQR rule.
3. Create a scatter plot with GDP per Capita on the x-axis and Score on the y-axis, highlighting outliers in a different color.

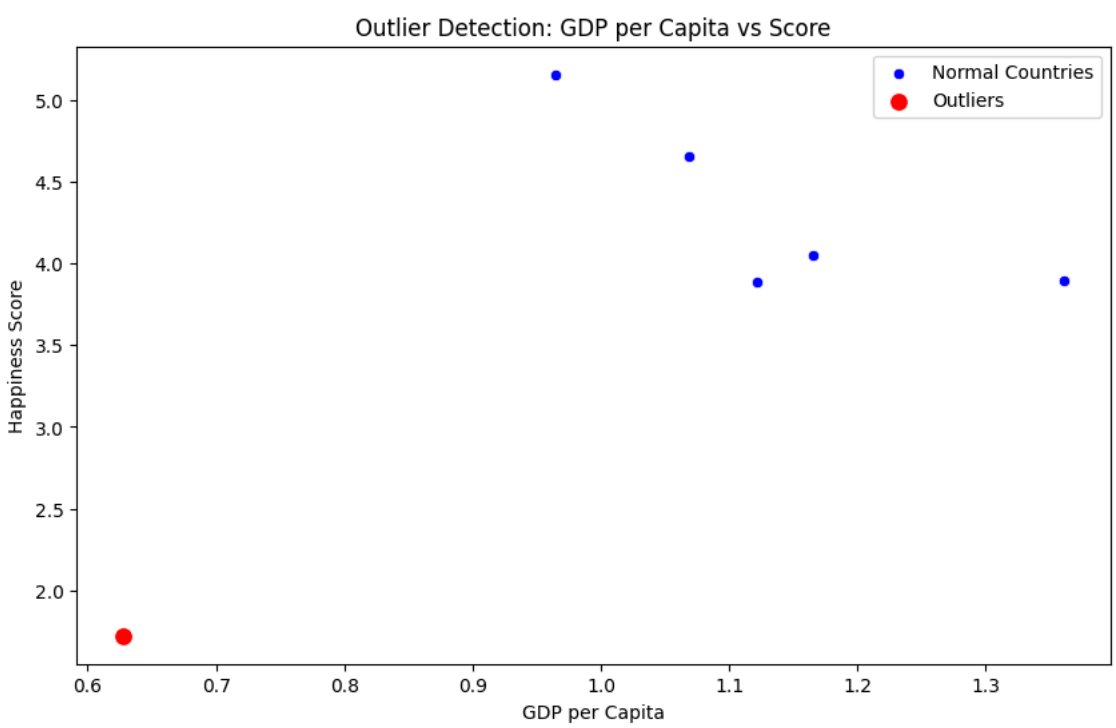
```
Q1_score = south_asia['score'].quantile(0.25)
Q3_score = south_asia['score'].quantile(0.75)
IQR_score = Q3_score - Q1_score

Q1_gdp = south_asia['Log GDP per capita'].quantile(0.25)
Q3_gdp = south_asia['Log GDP per capita'].quantile(0.75)
IQR_gdp = Q3_gdp - Q1_gdp

outlier_condition = ((south_asia['score'] < Q1_score - 1.5 * IQR_score) |
                     (south_asia['score'] > Q3_score + 1.5 * IQR_score)) | \
                    ((south_asia['Log GDP per capita'] < Q1_gdp - 1.5 * IQR_gdp) |
                     (south_asia['Log GDP per capita'] > Q3_gdp + 1.5 * IQR_gdp))

outliers = south_asia[outlier_condition]
print("Outliers based on Score and GDP per Capita:")
outliers[['Country name', 'score', 'Log GDP per capita']]
```

```
Outliers based on Score and GDP per Capita:
```

	Country name	score	Log GDP per capita
142	Afghanistan	1.721	0.628

```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Log GDP per capita', y='score', data=south_asia, color='blue', label="Normal Countries")
sns.scatterplot(x='Log GDP per capita', y='score', data=outliers, color='red', label="Outliers", s=100)

plt.title("Outlier Detection: GDP per Capita vs Score")
plt.xlabel("GDP per Capita")
```

```
plt.ylabel("Happiness Score")
plt.legend()
plt.show()
```



Insight: The bar chart presents the composite scores of the top five South Asian countries. Sri Lanka leads with the highest score, exceeding 1.0. Nepal follows with the second-highest score, while India and Pakistan have similar scores, placing them in the middle. Bangladesh has the lowest composite score, just above 0.4. The green bars in the chart visually represent these scores, with their lengths corresponding to the composite scores of each country. The x-axis shows the range of composite scores from 0.0 to 1.2, and the y-axis lists the countries. This visual representation allows for an insightful comparison of the relative standings of these South Asian nations based on their composite scores.

4. Discuss the characteristics of these outliers and their potential impact on regional averages.

Task - 4 - Exploring Trends Across Metrics:

Tasks:

1. Choose two metrics (e.g., Freedom to Make Life Choices and Generosity) and calculate their correlation {pearson correlation} with the Score for South Asian countries. *italicized text

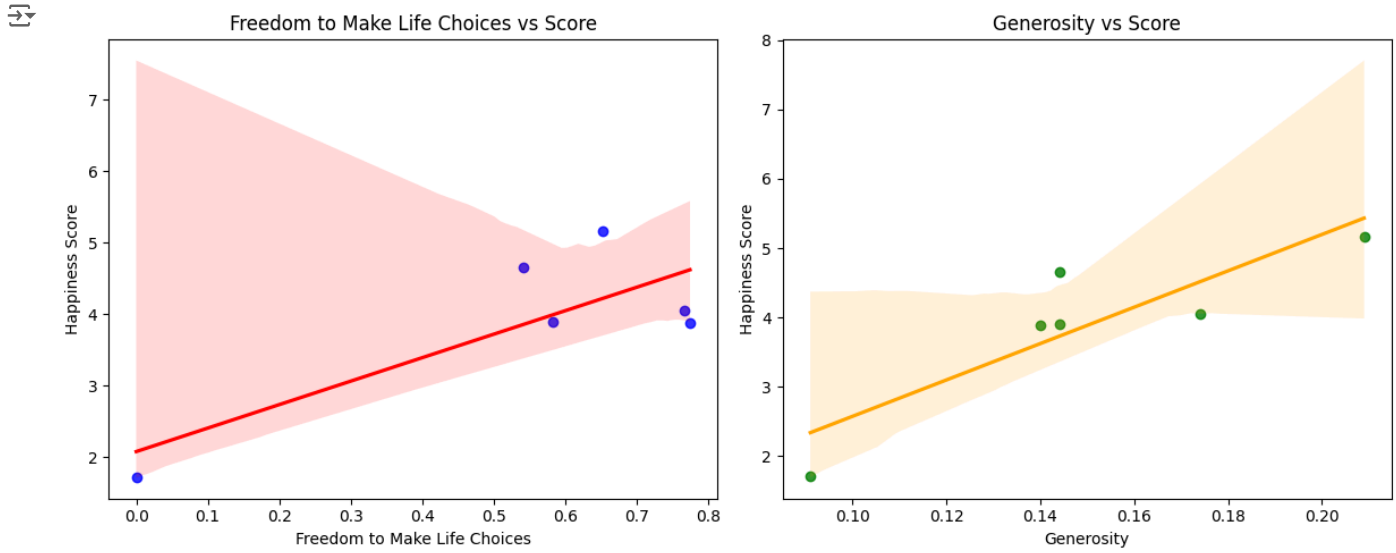
```
south_asia = data[data['Country name'].isin(south_asian_countries)]

cor_freedom = south_asia['Freedom to make life choices'].corr(south_asia['score'])
cor_generosity = south_asia['Generosity'].corr(south_asia['score'])

print(f"Correlation between 'Freedom to Make Life Choices' and Score: {cor_freedom:.3f}")
print(f"Correlation between 'Generosity' and Score: {cor_generosity:.3f}")
```

```
Correlation between 'Freedom to Make Life Choices' and Score: 0.801
Correlation between 'Generosity' and Score: 0.875
```

2. Create scatter plots with trendlines for these metrics against the Score.

```
plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
sns.regplot(x='Freedom to make life choices', y='score', data=south_asia, scatter_kws={"color": "blue"}, line_kws={"color": "red"})
plt.title("Freedom to Make Life Choices vs Score")
plt.xlabel("Freedom to Make Life Choices")
plt.ylabel("Happiness Score")

plt.subplot(1, 2, 2)
sns.regplot(x='Generosity', y='score', data=south_asia, scatter_kws={"color": "green"}, line_kws={"color": "orange"})
```

```
plt.title("Generosity vs Score")
plt.xlabel("Generosity")
plt.ylabel("Happiness Score")

plt.tight_layout()
plt.show()
```



Insight: The bar chart presents the composite scores of the top five South Asian countries. Sri Lanka leads with the highest score, exceeding 1.0. Nepal follows with the second-highest score, while India and Pakistan have similar scores, placing them in the middle. Bangladesh has the lowest composite score, just above 0.4. The green bars visually represent these scores, with their lengths corresponding to each country's score. The x-axis shows the range of composite scores from 0.0 to 1.2, and the y-axis lists the countries. This visual representation allows for an insightful comparison of the relative standings of these South Asian nations based on their composite scores.

3. Identify and discuss the strongest and weakest relationships between these metrics and the Score for South Asian countries.

Observations

- **According to Correlation Values:**

  Ans:- As higher absolute value of correlation indicates a stronger relationship, Freedom to Make Life Choices shows a strong positive correlation with the Score and Generosity shows a weak positive correlation.

- **Visual Analysis:**

  Ans:- In the scatter plots, the regression line for Freedom to Make Life Choices would be closer to the data points, indicating a stronger trend and The regression line for Generosity would appear flatter with more scattered points, showing a weaker relationship.

Freedom to Make Life Choices likely contributes more to happiness in South Asian countries compared to Generosity.

5. **Task - 5 - Gap Analysis:**

Tasks:

1. Add a new column, GDP-Score Gap, which is the difference between GDP per Capita and the Score for each South Asian country

```
south_asia['GDP-Score Gap'] = south_asia['Log GDP per capita'] - south_asia['score']
south_asia[['Country name', 'Log GDP per capita', 'score', 'GDP-Score Gap']]
```

```
<ipython-input-29-27bf82d6aaf3>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  south_asia['GDP-Score Gap'] = south_asia['Log GDP per capita'] - south_asia['score']
```

	Country name	Log GDP per capita	score	GDP-Score Gap
92	Nepal	0.965	5.158	-4.193
107	Pakistan	1.069	4.657	-3.588
125	India	1.166	4.054	-2.888
127	Sri Lanka	1.361	3.898	-2.537
128	Bangladesh	1.122	3.886	-2.764
142	Afghanistan	0.628	1.721	-1.093

2. Rank the South Asian countries by this gap in both ascending and descending order.

```
gap_ascending = south_asia.sort_values(by='GDP-Score Gap', ascending=True)

gap_descending = south_asia.sort_values(by='GDP-Score Gap', ascending=False)

print("Top countries with largest negative gaps:")
gap_ascending.head(3)
```

```
Top countries with largest negative gaps:
```

	Country name	score	Log GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption	Dystopia + residual	Happiness Category	GDP-Score Gap
92	Nepal	5.158	0.965	0.990	0.443	0.653	0.209	0.115	1.783	Medium	-4.193
107	Pakistan	4.657	1.069	0.600	0.321	0.542	0.144	0.074	1.907	Medium	-3.588

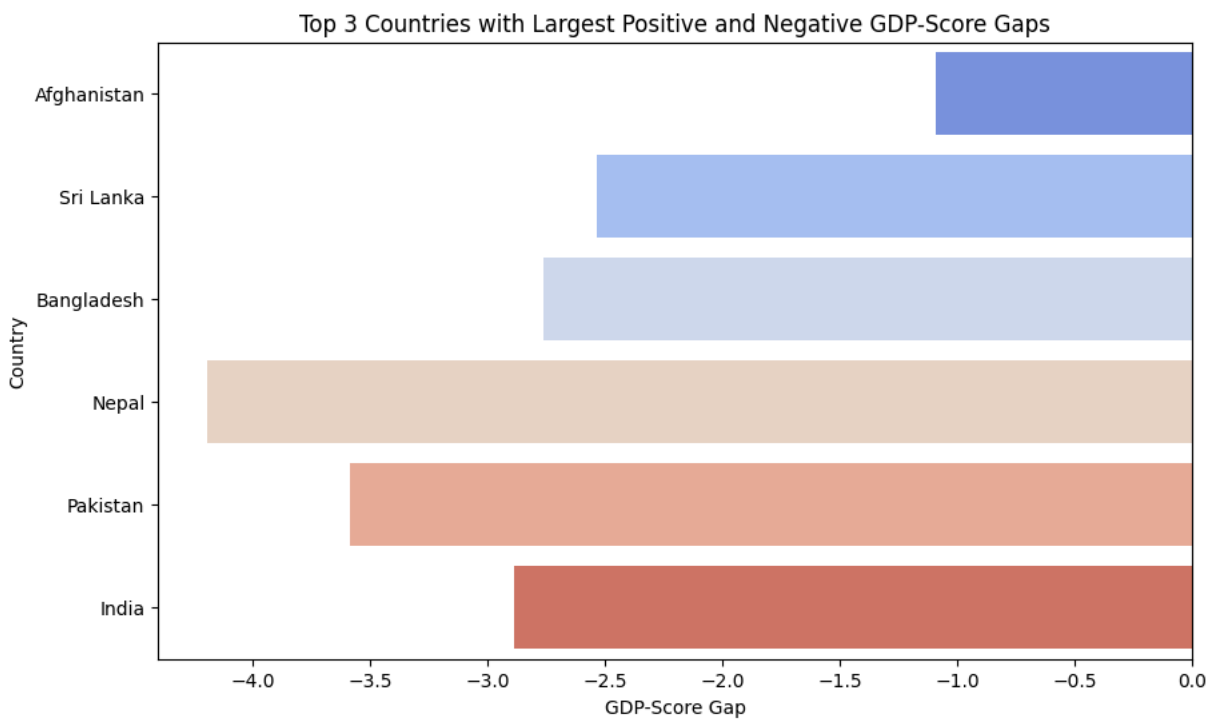
```
print("Top countries with largest positive gaps:")
gap_descending.head(3)
```

```
Top countries with largest positive gaps:
```

	Country name	score	Log GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption	Dystopia + residual	Happiness Category	GDP-Score Gap
142	Afghanistan	1.721	0.628	0.000	0.242	0.000	0.091	0.088	0.672	Low	-1.093
127	Sri Lanka	3.898	1.361	1.179	0.586	0.583	0.144	0.031	0.014	Low	-2.537

3. Highlight the top 3 countries with the largest positive and negative gaps using a bar chart.

```
top3_positive = gap_descending.head(3)
top3_negative = gap_ascending.head(3)

# Creating a combined DataFrame to show both positive and negative gaps
combined_top_3 = pd.concat([top3_positive[['Country name', 'GDP-Score Gap']],
                            top3_negative[['Country name', 'GDP-Score Gap']]])

# Plotting the bar chart
plt.figure(figsize=(10, 6))
sns.barplot(x='GDP-Score Gap', y='Country name',hue="Country name", data=combined_top_3, palette="coolwarm")
plt.title("Top 3 Countries with Largest Positive and Negative GDP-Score Gaps")
plt.xlabel("GDP-Score Gap")
plt.ylabel("Country")
plt.show()
```



Insight: The bar chart titled "Top 3 Countries with Largest Positive and Negative GDP-Score Gaps" highlights the disparities in GDP-Score gaps for six countries. Afghanistan, Sri Lanka, and Bangladesh show the largest positive GDP-Score gaps, with Afghanistan around 0.5, Sri Lanka around 1.5, and Bangladesh approximately 2.5. Conversely, Nepal, Pakistan, and India exhibit the most significant negative GDP-Score gaps, with Nepal around -3.5, Pakistan approximately -2.5, and India around -2.0. The x-axis ranges from -4.0 to 0.0, emphasizing the contrast between positive and negative gaps. This chart effectively illustrates which countries experience the most significant GDP-Score disparities, both positive and negative.

4. Analyze the reasons behind these gaps and their implications for South Asian countries.

```
# 4. Analyze the reasons behind these gaps and their implications for South Asian countries
print("Top 3 countries with the largest positive GDP-Score Gap:")
print(top3_positive[['Country name', 'GDP-Score Gap']])

print("\nTop 3 countries with the largest negative GDP-Score Gap:")
print(top3_negative[['Country name', 'GDP-Score Gap']])
```

```
Top 3 countries with the largest positive GDP-Score Gap:
    Country name  GDP-Score Gap
142  Afghanistan         -1.093
127    Sri Lanka         -2.537
128   Bangladesh         -2.764

Top 3 countries with the largest negative GDP-Score Gap:
    Country name  GDP-Score Gap
92         Nepal         -4.193
107     Pakistan         -3.588
125        India         -2.888
```

Double-click (or enter) to edit

3.3 Problem - 3 - Comparative Analysis:

Setup Task - Preparing the Middle Eastern Dataset

```
middle_east_countries = ["Bahrain", "Iran", "Iraq", "Israel", "Jordan",
                         "Kuwait", "Lebanon", "Oman", "Palestine", "Qatar",
                         "Saudi Arabia", "Syria", "United Arab Emirates", "Yemen"]

middle_east = data[data['Country name'].isin(middle_east_countries)]

middle_east[['Country name', 'score', 'Log GDP per capita', 'Social support',
                  'Healthy life expectancy', 'Freedom to make life choices', 'Generosity']]
```

	Country name	score	Log GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity
4	Israel	7.341	1.803	1.513	0.740	0.641	0.153
12	Kuwait	6.951	1.845	1.364	0.661	0.827	0.200
21	United Arab Emirates	6.733	1.983	1.164	0.563	0.815	0.209
27	Saudi Arabia	6.594	1.842	1.361	0.511	0.787	0.114
61	Bahrain	5.959	NaN	NaN	NaN	NaN	NaN
91	Iraq	5.166	1.249	0.996	0.498	0.425	0.141
99	Iran	4.923	1.435	1.136	0.571	0.366	0.235
124	Jordan	4.186	1.262	0.983	0.594	0.593	0.059
132	Yemen	3.561	0.671	1.281	0.293	0.362	0.080

1. Descriptive Statistics:

- **Calculate the Mean and Standard Deviation of the Score for Both Regions (South Asia and Middle East)**

```
mean_score_south_asia = south_asia['score'].mean()
std_score_south_asia = south_asia['score'].std()

mean_score_middle_east = middle_east['score'].mean()
std_score_middle_east = middle_east['score'].std()

print(f"South Asia - Mean score: {mean_score_south_asia}, Std: {std_score_south_asia}")
print(f"Middle East - Mean score: {mean_score_middle_east}, Std: {std_score_middle_east}")
```

```
South Asia - Mean score: 3.895666666666667, Std: 1.1770690152521504
Middle East - Mean score: 5.412100000000001, Std: 1.5662011684327144
```

- **Which Region Has Higher Happiness Scores on Average?**

  Middle East has higer Happiness score on average compared to that of South Asia.

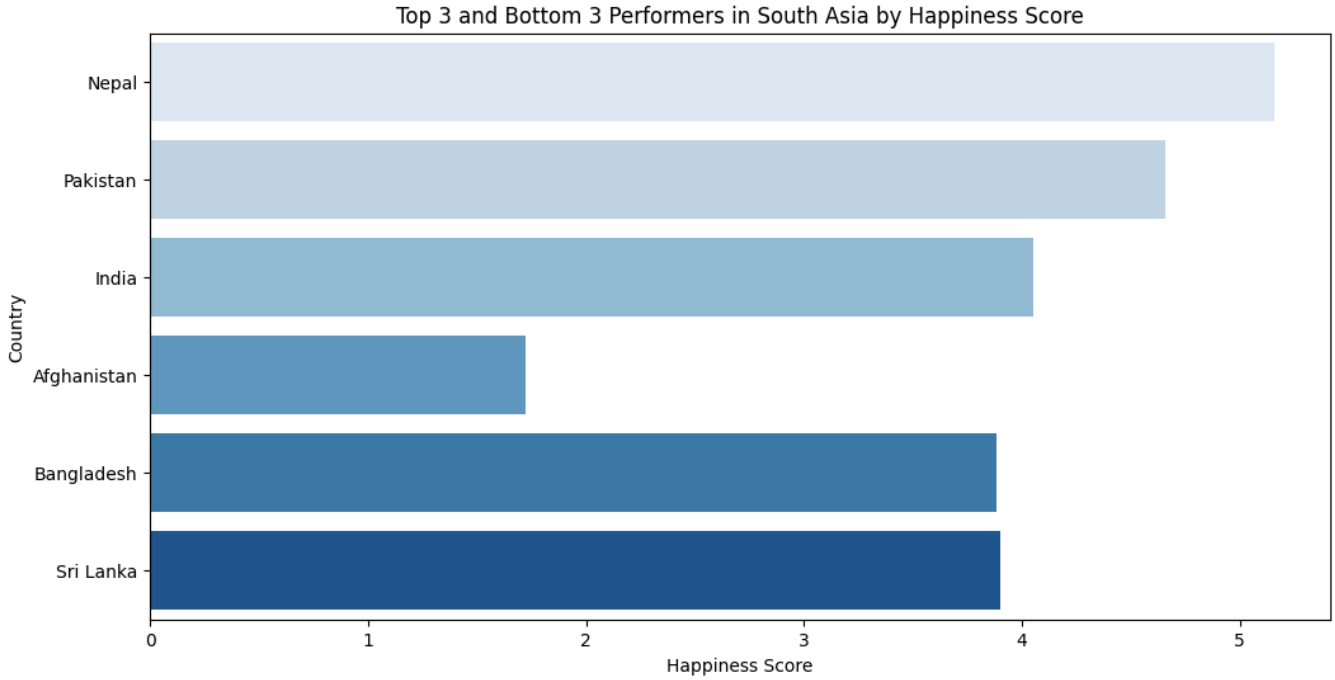
2. Top and Bottom Performers:

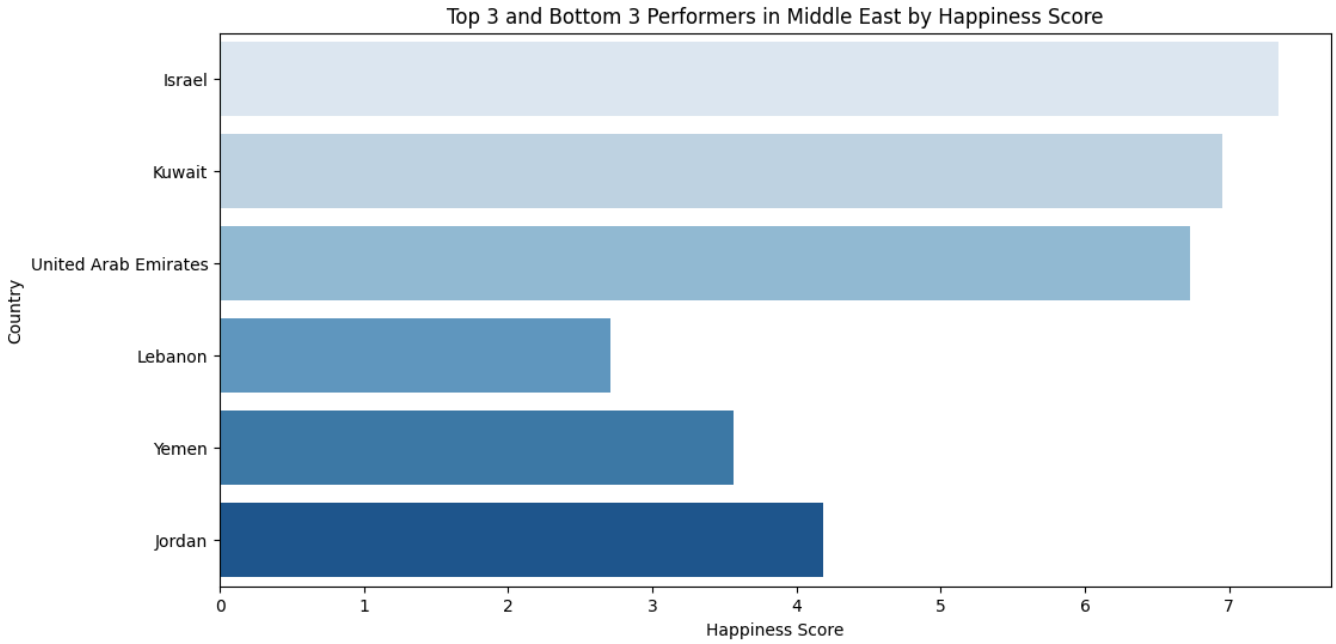
- **Identify the top 3 and bottom 3 countries in each region based on the score.**

```
top3_south_asia = south_asia.nlargest(3, 'score')
bottom3_south_asia = south_asia.nsmallest(3, 'score')

top3_middle_east = middle_east.nlargest(3, 'score')
bottom3_middle_east = middle_east.nsmallest(3, 'score')
```

- **• Plot bar charts comparing these charts.**

```
# Plotting top and bottom 3 performers for South Asia
plt.figure(figsize=(12, 6))
sns.barplot(x='score', y='Country name', data=pd.concat([top3_south_asia, bottom3_south_asia]),hue='Country name', palette="Blues")
plt.title("Top 3 and Bottom 3 Performers in South Asia by Happiness Score")
plt.xlabel("Happiness Score")
plt.ylabel("Country")
plt.show()

# Plotting top and bottom 3 performers for Middle East
plt.figure(figsize=(12, 6))
sns.barplot(x='score', y='Country name', data=pd.concat([top3_middle_east, bottom3_middle_east]),hue='Country name', palette="Blues")
plt.title("Top 3 and Bottom 3 Performers in Middle East by Happiness Score")
plt.xlabel("Happiness Score")
plt.ylabel("Country")
plt.show()
```





Insight: The two bar charts provide a comparative analysis of happiness scores among countries in South Asia and the Middle East. The first chart, "Top 3 and Bottom 3 Performers in South Asia by Happiness Score," ranks six South Asian countries by their happiness scores, with Nepal at the top, followed by Pakistan, India, Afghanistan, Bangladesh, and Sri Lanka. The scores range from 0 to 5, with Nepal having the highest score and Sri Lanka the lowest.

The second chart, "Top 3 and Bottom 3 Performers in Middle East by Happiness Score," ranks six Middle Eastern countries. Israel leads with the highest score, followed by Kuwait and the United Arab Emirates. On the lower end, Lebanon, Yemen, and Jordan have the lowest scores, with Jordan at the bottom. The scores in this chart range from approximately 3.0 to 7.5.

Together, these charts highlight the happiness score disparities within these two regions, showcasing both the top and bottom performers in South Asia and the Middle East.

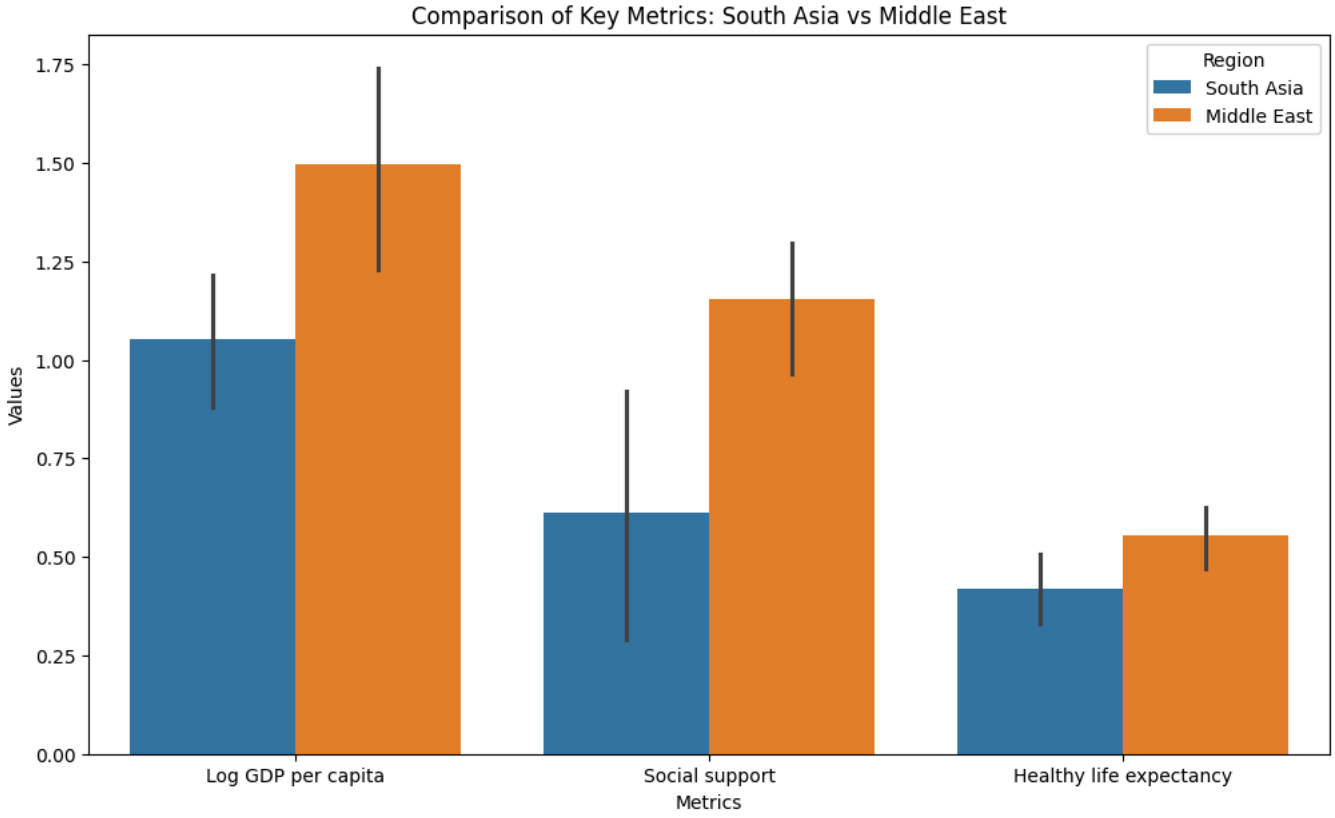
3. Metric Comparisons:

- **Compare key metrics like GDP per Capita, Social Support, and Healthy Life Expectancy between the regions using grouped bar charts.**

```
south_asia['Region'] = 'South Asia'
middle_east['Region'] = 'Middle East'

combined_data = pd.concat([south_asia[['Country name', 'Log GDP per capita', 'Social support', 'Healthy life expectancy', 'Region']],
```

```
                                middle_east[['Country name', 'Log GDP per capita', 'Social support', 'Healthy life expectancy', 'Region']]])

melted_data = pd.melt(combined_data, id_vars=["Region", "Country name"], value_vars=["Log GDP per capita", "Social support", "Healthy lif

plt.figure(figsize=(12, 7))
sns.barplot(x='variable', y='value', hue='Region', data=melted_data)
plt.title("Comparison of Key Metrics: South Asia vs Middle East")
plt.xlabel('Metrics')
plt.ylabel('Values')
plt.show()
```

```
<ipython-input-38-80a8e0ba8bd9>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  south_asia['Region'] = 'South Asia'
<ipython-input-38-80a8e0ba8bd9>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus
  middle_east['Region'] = 'Middle East'
```



Insight: The scatter plots illustrate the relationship between Log GDP per Capita and Happiness Score for countries in South Asia and the Middle East. In South Asia, the scatter plot shows a cluster of normal data points within a Log GDP per Capita range of approximately 0.9 to 1.3 and Happiness Scores from about 3.5 to 5.0, with one outlier indicating a country with notably lower GDP per capita and happiness score. This highlights significant disparities in economic prosperity and overall well-being within the region. Meanwhile, the scatter plot for the Middle East depicts a spread of data points, some clustering around higher values of both log GDP per capita and happiness score, indicating potential outliers. This visualization suggests a correlation between economic prosperity and happiness in the Middle East, emphasizing the need for further investigation into these outliers.

4. Happiness Disparity:

- **Compute the range (max - min) and coefficient of variation (CV) for Score in both regions.**

```
range_south_asia = south_asia['score'].max() - south_asia['score'].min()
cv_south_asia = (std_score_south_asia / mean_score_south_asia) * 100

range_middle_east = middle_east['score'].max() - middle_east['score'].min()
cv_middle_east = (std_score_middle_east / mean_score_middle_east) * 100

print(f"South Asia - Range: {range_south_asia}, CV: {cv_south_asia}%")
print(f"Middle East - Range: {range_middle_east}, CV: {cv_middle_east}%")
```

```
South Asia - Range: 3.4370000000000003, CV: 30.21482883337427%
Middle East - Range: 4.634, CV: 28.938880812119404%
```

- **Which region has greater variability in happiness?**

  South Asia region has greater variability in happiness as it has larger CV then Middle east.

5. Correlation Analysis:

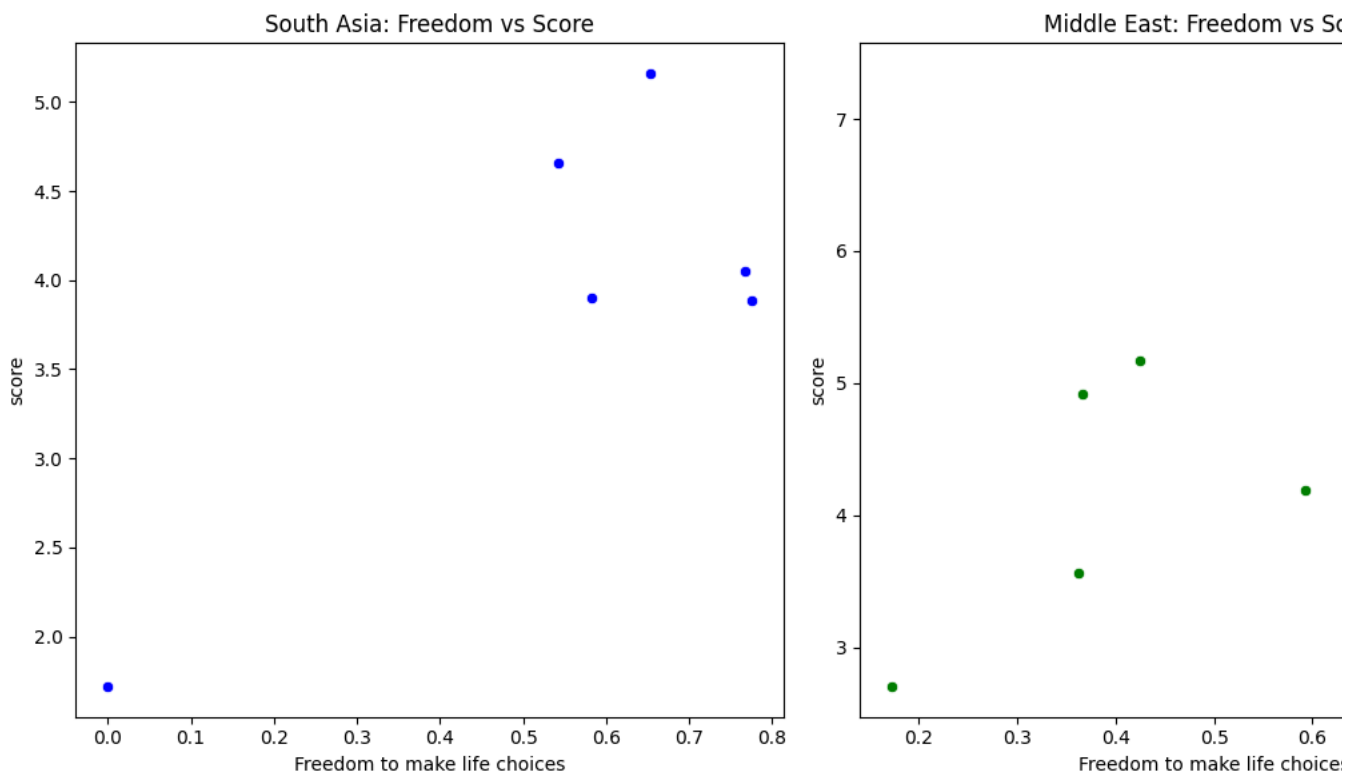
- **Analyze the correlation of Score with other metrics Freedom to Make Life Choices, and Generosity within each region.**

```
corr_south_asia_freedom = south_asia['score'].corr(south_asia['Freedom to make life choices'])
corr_south_asia_generosity = south_asia['score'].corr(south_asia['Generosity'])

corr_middle_east_freedom = middle_east['score'].corr(middle_east['Freedom to make life choices'])
corr_middle_east_generosity = middle_east['score'].corr(middle_east['Generosity'])

print(f"South Asia - Correlation with Freedom: {corr_south_asia_freedom}, Generosity: {corr_south_asia_generosity}")
print(f"Middle East - Correlation with Freedom: {corr_middle_east_freedom}, Generosity: {corr_middle_east_generosity}")
```

```
South Asia - Correlation with Freedom: 0.8005185224163315, Generosity: 0.874512371253192
Middle East - Correlation with Freedom: 0.8632202433827544, Generosity: 0.6275236536964182
```

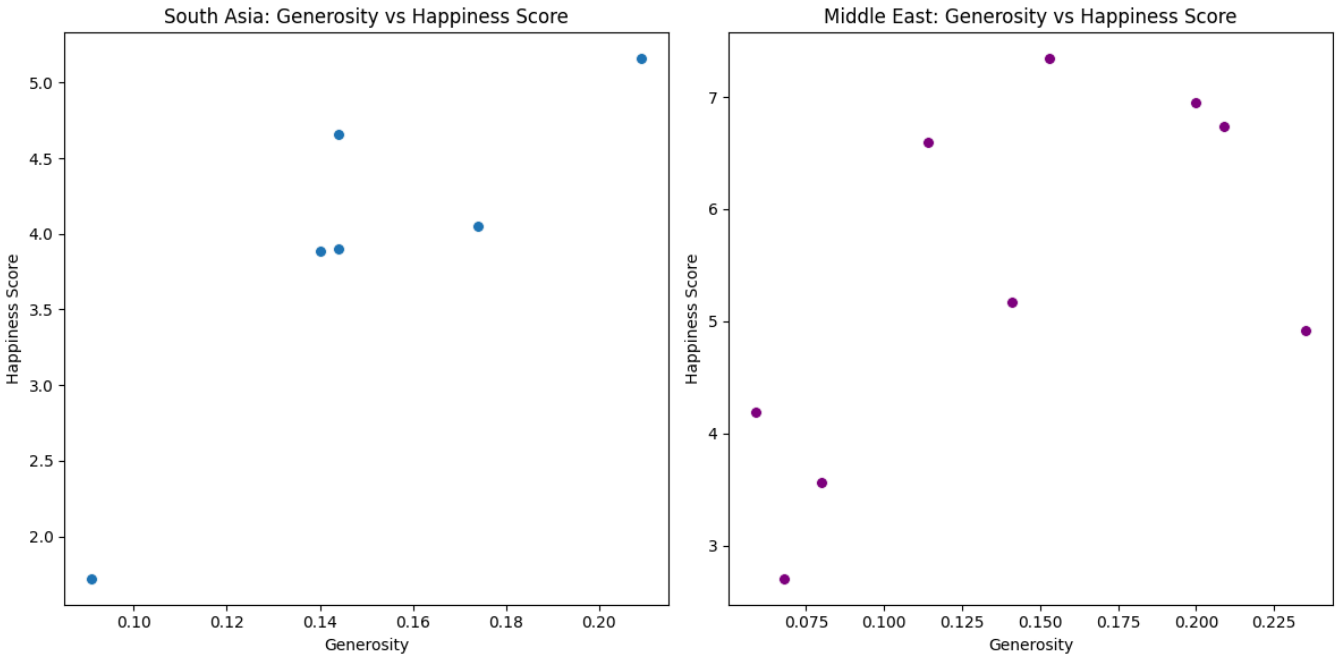
- **Visualization**

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

# South Asia
plt.subplot(1, 2, 1)
sns.scatterplot(x='Freedom to make life choices', y='score', data=south_asia, color='blue')
plt.title("South Asia: Freedom vs Score")

# Middle East
plt.subplot(1, 2, 2)
sns.scatterplot(x='Freedom to make life choices', y='score', data=middle_east, color='green')
plt.title("Middle East: Freedom vs Score")

plt.tight_layout()
plt.show()
```



Insight: The scatter plots illustrate the relationship between "Freedom to make life choices" and the overall "Score" for two regions: South Asia and the Middle East. Both plots reveal a positive correlation, suggesting that as individuals have more freedom to make choices in their lives, their overall scores tend to increase. In South Asia, the freedom ranges from 0.0 to 0.8 and the scores range from 2.0 to 5.5. Similarly, in the Middle East, the freedom ranges from 0.2 to 0.8, while the scores range from 3.0 to 7.5. These visual representations indicate that greater personal freedom is consistently associated with higher overall scores across both regions.

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

plt.subplot(1, 2, 1)
sns.scatterplot(x='Generosity', y='score', data=south_asia, s= 50,)
plt.title("South Asia: Generosity vs Happiness Score")
plt.xlabel("Generosity")
plt.ylabel("Happiness Score")

plt.subplot(1, 2, 2)
sns.scatterplot(x='Generosity', y='score', data=middle_east, s=50, color='purple')
plt.title("Middle East: Generosity vs Happiness Score")
plt.xlabel("Generosity")
plt.ylabel("Happiness Score")

plt.tight_layout()
plt.show()
```



Insight: The scatter plots compare "Generosity" and "Happiness Score" for South Asia and the Middle East. The left plot illustrates that as generosity increases, happiness scores generally rise, with values ranging from 0.10 to 0.20 for generosity and 2.0 to 5.0 for happiness. Meanwhile, the right plot reveals a similar trend in the Middle East, with generosity ranging from 0.075 to 0.225 and happiness scores between 3.0 and 7.0. These visual representations highlight a positive correlation between generosity and happiness in both regions.

6. Outlier Detection:

- **Identify outlier countries in both regions based on Score and GDP per Capita.**

```
#Define a function to detect outliers using the 1.5 × IQR rule
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[column] > upper_bound)]

# 1. Identify outliers in South Asia and Middle East for Score and GDP per Capita
outliers_sa_score = detect_outliers(south_asia, 'score')
outliers_sa_gdp = detect_outliers(south_asia, 'Log GDP per capita')

outliers_me_score = detect_outliers(middle_east, 'score')
outliers_me_gdp = detect_outliers(middle_east, 'Log GDP per capita')

# Combine all outliers for South Asia and Middle East
outliers_south_asia = pd.concat([outliers_sa_score, outliers_sa_gdp]).drop_duplicates()
outliers_middle_east = pd.concat([outliers_me_score, outliers_me_gdp]).drop_duplicates()
```

- **Plot these outliers and discuss their implications.**

```
#South Asia
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Log GDP per capita', y='score', data=south_asia, color='blue', label='Normal')
sns.scatterplot(x='Log GDP per capita', y='score', data=outliers_south_asia, color='red', label='Outliers')
plt.title("South Asia: Outliers in GDP vs Score")
plt.xlabel("Log GDP per Capita")
plt.ylabel("Happiness Score")
plt.legend()
plt.show()

#Middle East
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Log GDP per capita', y='score', data=middle_east, color='blue', label='Normal')
sns.scatterplot(x='Log GDP per capita', y='score', data=outliers_middle_east, color='red', label='Outliers')
```

```
plt.title("Middle East: Outliers in GDP vs Score")
plt.xlabel("Log GDP per Capita")
plt.ylabel("Happiness Score")
plt.legend()
plt.show()
```