Introduction

This project focuses on analyzing global safety and law and order indices using datasets derived from various countries. The main objective is to explore the relationship between perceived safety and law order scores, identify patterns, and detect outliers. By comparing these indices, we aim to uncover insights into global safety perceptions and law enforcement effectiveness across different nations.

To facilitate the analysis, we first preprocess the data by ranking countries based on their respective safety and law order scores. We then create interactive visualizations such as world maps and scatter plots to observe geographical trends and correlations between the two indices. Outlier analysis is also conducted to identify countries with unusual patterns—those with high safety but low law order scores and vice versa.

Finally, a correlation analysis is performed to quantify the relationship between safety and law order indices, highlighting regions or countries that stand out. This project provides a comprehensive view of how safety and law order scores vary across the globe, enabling further research into the factors influencing these metrics.

Dataset collected from: Gallup.com

Background Information: A recent Gallup global safety survey has classified Kuwait as the safest country in the world for 2024 regarding individuals walking alone in the streets, with an impressive 99% of respondents reporting they feel safe. Singapore follows in second place with 94%, while Norway and Saudi Arabia both score 92%, reports Al-Seyassah daily.

Kuwait also achieved the highest score on the Law and Order Index, receiving an impressive 98 out of 100, maintaining its position as one of the safest countries in the world since 2019. Additionally, the insurance company Hellosafe has released a "Travel Safety Index," which rates countries on a scale of 0 to 100, where 0 indicates the safest countries and 100 the least safe, according to a report by Le Soir. This index is based on 35 criteria, including the occurrence of natural disasters, societal violence, involvement in armed conflicts (both internal and external), and health infrastructure.

Importing Necessary Libraries:

In [1]:
```
pip install plotly
```

```
Requirement already satisfied: plotly in /Users/rising.volkan007/anaconda
3/lib/python3.11/site-packages (5.24.1)
Requirement already satisfied: tenacity>=6.2.0 in /Users/rising.volkan00
7/anaconda3/lib/python3.11/site-packages (from plotly) (8.2.3)
Requirement already satisfied: packaging in /Users/rising.volkan007/anaco
nda3/lib/python3.11/site-packages (from plotly) (24.1)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

This block imports the essential libraries used throughout the project:

- pandas: For handling data frames and performing analysis.
- plotly.express: For interactive visualizations (world maps, scatter plots).
- matplotlib.pyplot and seaborn: For additional static visualizations like scatter plots and heatmaps.

Loading and Preprocessing the Data:

In [3]:
```
#loading the dataset

safety= pd.read_csv('/Users/rising.volkan007/Downloads/Global_safety/Gall

law_order= pd.read_csv('/Users/rising.volkan007/Downloads/Global_safety/L
```

This block loads the safety and law order data from CSV files and previews the first few rows of each dataset. Replace the CSV filenames with actual filenames if different.

In [4]:
```
safety.head()
```

Out[4]:

	Country_name	Value
0	Ecuador	27 %
1	Liberia	30 %
2	South Africa	30 %
3	Botswana	32 %
4	Chile	36 %

In [5]:
```
law_order.head()
```

Out[5]:

	Country_name	Value
0	Albania	84
1	Argentina	65
2	Armenia	85
3	Australia	81
4	Austria	87

Data Preprocessing:

In [6]:
```
# Check the data types and basic information
print(safety.info())
print(law_order.info())

# Descriptive statistics
print(safety.describe())
print(law_order.describe())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 140 entries, 0 to 139
Data columns (total 2 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   Country_name  140 non-null    object
 1   Value         140 non-null    object
dtypes: object(2)
memory usage: 2.3+ KB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 140 entries, 0 to 139
Data columns (total 2 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   Country_name  140 non-null    object
 1   Value         140 non-null    int64 
dtypes: int64(1), object(1)
memory usage: 2.3+ KB
None
       Country_name Value
count           140   140
unique          140    57
top         Ecuador  72 %
freq              1     7
            Value
count  140.000000
mean    77.692857
std     10.095869
min     50.000000
25%     71.000000
50%     80.000000
75%     86.000000
max     98.000000
```

In [7]:
```
# Remove any non-breaking spaces and percentage sign, then convert to flo
safety['Value'] = safety['Value'].str.replace(r'\s*%', '', regex=True).as
```

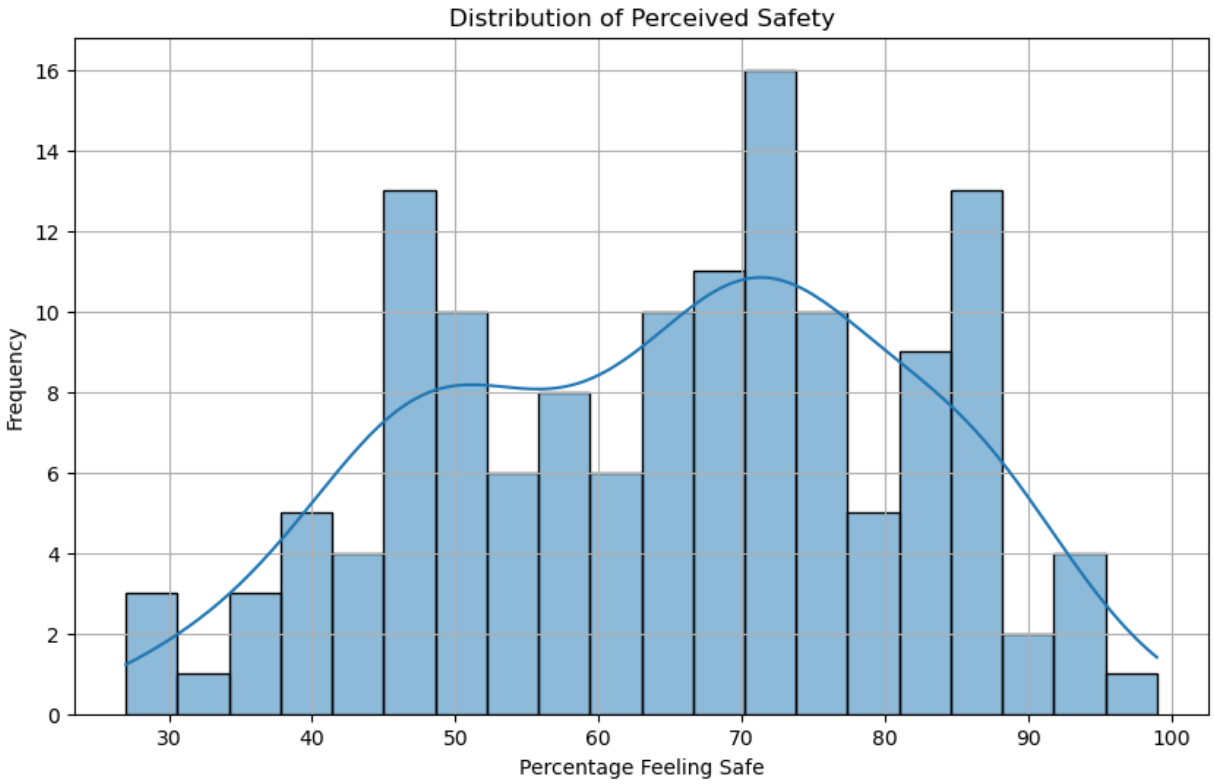
In [8]:
```
# Verify the data types
print(safety.dtypes)

# Display the first few rows of the cleaned DataFrame
print(safety.head())
```

```
Country_name    object
Value            int64
dtype: object
   Country_name  Value
0       Ecuador     27
1       Liberia     30
2  South Africa     30
3      Botswana     32
4         Chile     36
```

In [9]:
```
print(safety['Value'].describe())
```

```
count    140.000000
mean      64.864286
std       16.454957
min       27.000000
25%       51.000000
50%       67.000000
75%       77.000000
max       99.000000
Name: Value, dtype: float64
```

In [10]:
```
plt.figure(figsize=(10, 6))
sns.histplot(safety['Value'], bins=20, kde=True)
plt.title('Distribution of Perceived Safety')
plt.xlabel('Percentage Feeling Safe')
plt.ylabel('Frequency')
plt.grid()
plt.show()
```



Analysis and Visualization:

In [11]:
```
outliers_low = safety[safety['Value'] < 40]
outliers_high = safety[safety['Value'] > 80]
print("Countries with low perceived safety:\n", outliers_low)
print("Countries with high perceived safety:\n", outliers_high)
```

Countries with low perceived safety:

	Country_name	Value
0	Ecuador	27
1	Liberia	30
2	South Africa	30
3	Botswana	32
4	Chile	36
5	Eswatini	37
6	Namibia	37
7	The Gambia	38
8	Zimbabwe	38

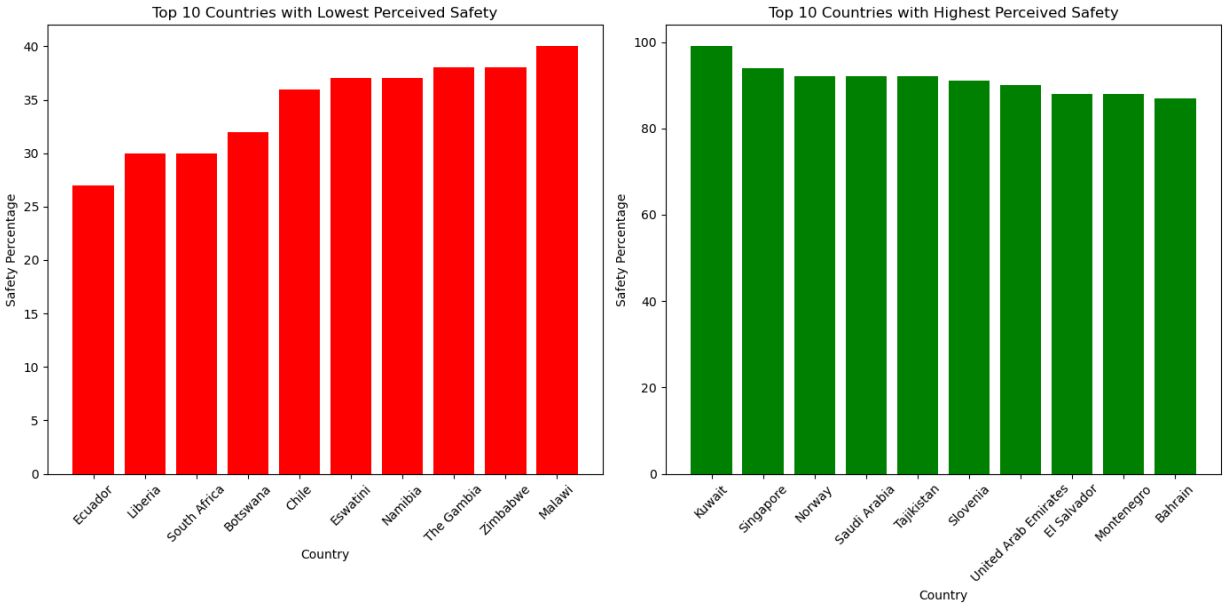
Countries with high perceived safety:

	Country_name	Value
111	Egypt	81
112	Somalia	81
113	Austria	82
114	Spain	82
115	Indonesia	83
116	Netherlands	83
117	Portugal	83
118	Taiwan, Province of China	84
119	Uzbekistan	84
120	Estonia	85
121	Finland	85
122	Hong Kong (S.A.R. of China)	85
123	Armenia	86
124	China	86
125	Denmark	86
126	Bahrain	87
127	Iceland	87
128	Kosovo	87
129	Luxembourg	87
130	Switzerland	87
131	El Salvador	88
132	Montenegro	88
133	United Arab Emirates	90
134	Slovenia	91
135	Norway	92
136	Saudi Arabia	92
137	Tajikistan	92
138	Singapore	94
139	Kuwait	99

In [12]:

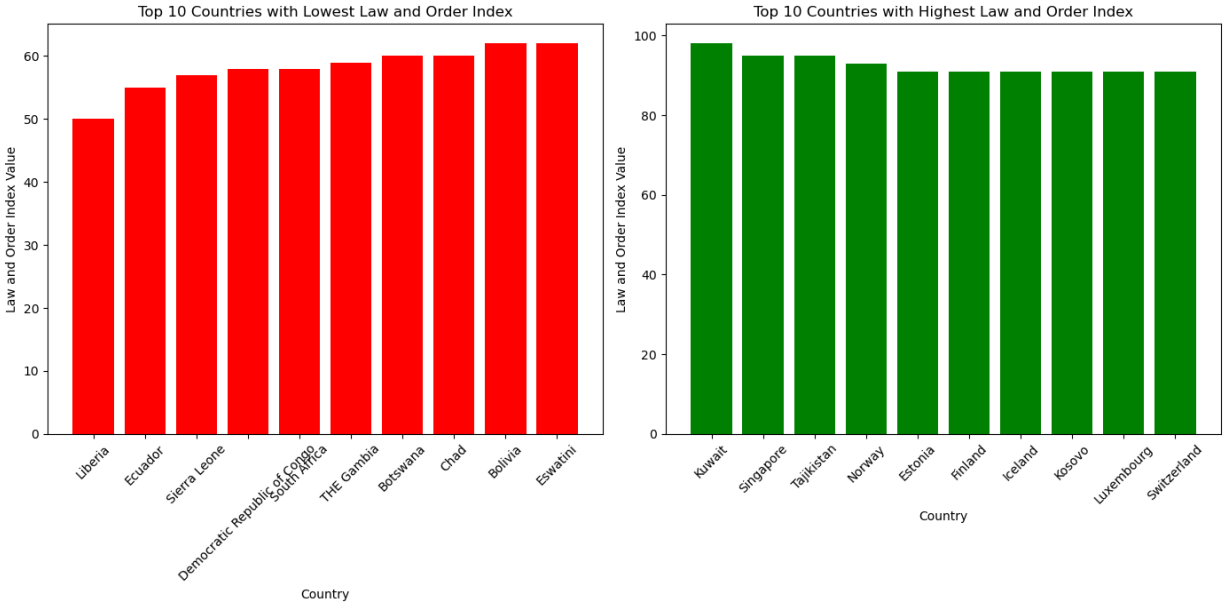
```
# Get the top 10 countries with the lowest and highest perceived safety
top_10_low_safety = safety.nsmallest(10, 'Value')
top_10_high_safety = safety.nlargest(10, 'Value')

# Get the top 10 countries with the lowest and highest Law and Order Inde
top_10_low_law_order = law_order.nsmallest(10, 'Value')
top_10_high_law_order = law_order.nlargest(10, 'Value')

# Set up the figure and axes for safety bar charts
fig, axes = plt.subplots(1, 2, figsize=(14, 7))

# Bar chart for low perceived safety
axes[0].bar(top_10_low_safety['Country_name'], top_10_low_safety['Value']
axes[0].set_title('Top 10 Countries with Lowest Perceived Safety')
axes[0].set_xlabel('Country')
axes[0].set_ylabel('Safety Percentage')
axes[0].set_xticklabels(top_10_low_safety['Country_name'], rotation=45)

# Bar chart for high perceived safety
axes[1].bar(top_10_high_safety['Country_name'], top_10_high_safety['Value
axes[1].set_title('Top 10 Countries with Highest Perceived Safety')
axes[1].set_xlabel('Country')
axes[1].set_ylabel('Safety Percentage')
axes[1].set_xticklabels(top_10_high_safety['Country_name'], rotation=45)

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

```
/var/folders/3j/tlpxgbnn03s1zb0rcz4xx9j80000gn/T/ipykernel_71198/14701517
95.py:17: UserWarning: set_ticklabels() should only be used with a fixed
number of ticks, i.e. after set_ticks() or using a FixedLocator.
  axes[0].set_xticklabels(top_10_low_safety['Country_name'], rotation=45)
/var/folders/3j/tlpxgbnn03s1zb0rcz4xx9j80000gn/T/ipykernel_71198/14701517
95.py:24: UserWarning: set_ticklabels() should only be used with a fixed
number of ticks, i.e. after set_ticks() or using a FixedLocator.
  axes[1].set_xticklabels(top_10_high_safety['Country_name'], rotation=4
5)
```



In [13]:
```
# Set up the figure and axes for law order bar charts
fig, axes = plt.subplots(1, 2, figsize=(14, 7))

# Bar chart for low law order index
axes[0].bar(top_10_low_law_order['Country_name'], top_10_low_law_order['V
axes[0].set_title('Top 10 Countries with Lowest Law and Order Index')
axes[0].set_xlabel('Country')
axes[0].set_ylabel('Law and Order Index Value')
axes[0].set_xticklabels(top_10_low_law_order['Country_name'], rotation=45

# Bar chart for high law order index
axes[1].bar(top_10_high_law_order['Country_name'], top_10_high_law_order[
axes[1].set_title('Top 10 Countries with Highest Law and Order Index')
axes[1].set_xlabel('Country')
axes[1].set_ylabel('Law and Order Index Value')
axes[1].set_xticklabels(top_10_high_law_order['Country_name'], rotation=4

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

```
/var/folders/3j/tlpxgbnn03s1zb0rcz4xx9j80000gn/T/ipykernel_71198/13652554
06.py:9: UserWarning: set_ticklabels() should only be used with a fixed n
umber of ticks, i.e. after set_ticks() or using a FixedLocator.
  axes[0].set_xticklabels(top_10_low_law_order['Country_name'], rotation=
45)
/var/folders/3j/tlpxgbnn03s1zb0rcz4xx9j80000gn/T/ipykernel_71198/13652554
06.py:16: UserWarning: set_ticklabels() should only be used with a fixed
number of ticks, i.e. after set_ticks() or using a FixedLocator.
  axes[1].set_xticklabels(top_10_high_law_order['Country_name'], rotation
=45)
```



In [14]:

```python
# Get the top 20 countries for safety
top_safety = safety.nlargest(20, 'Value').copy()

# Get the top 20 countries for law order
top_law_order = law_order.nlargest(20, 'Value').copy()

# Reset index for both DataFrames
top_safety.reset_index(drop=True, inplace=True)
top_law_order.reset_index(drop=True, inplace=True)

# Display the top 20 countries for each metric
print("Top 20 Countries by Perceived Safety:")
print(top_safety)

print("\nTop 20 Countries by Law and Order Index:")
print(top_law_order)
```

Top 20 Countries by Perceived Safety:

	Country_name	Value
0	Kuwait	99
1	Singapore	94
2	Norway	92
3	Saudi Arabia	92
4	Tajikistan	92
5	Slovenia	91
6	United Arab Emirates	90
7	El Salvador	88
8	Montenegro	88
9	Bahrain	87
10	Iceland	87
11	Kosovo	87
12	Luxembourg	87
13	Switzerland	87
14	Armenia	86
15	China	86
16	Denmark	86
17	Estonia	85
18	Finland	85
19	Hong Kong (S.A.R. of China)	85

Top 20 Countries by Law and Order Index:

	Country_name	Value
0	Kuwait	98
1	Singapore	95
2	Tajikistan	95
3	Norway	93
4	Estonia	91
5	Finland	91
6	Iceland	91
7	Kosovo	91
8	Luxembourg	91
9	Switzerland	91
10	Denmark	90
11	United Arab Emirates	90
12	Vietnam	90
13	Bahrain	89
14	El Salvador	89
15	Indonesia	89
16	Portugal	89
17	Saudi Arabia	89
18	Slovenia	89
19	Uzbekistan	89

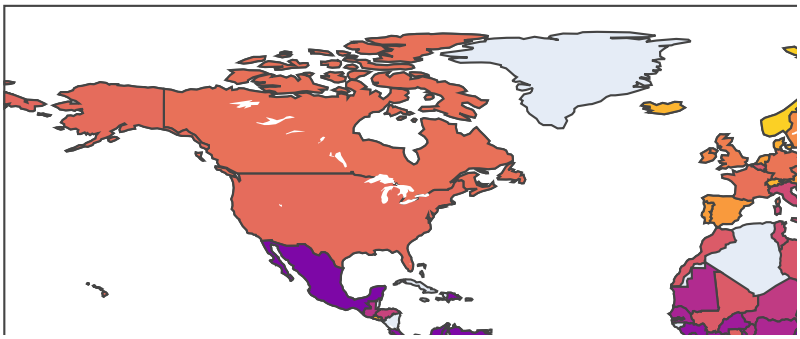
In [15]:
```
# Rank all countries in the safety dataset based on their values
safety['Rank'] = safety['Value'].rank(ascending=False, method='min')
safety_sorted = safety.sort_values(by='Rank').reset_index(drop=True)

# Rank all countries in the law order dataset based on their values
law_order['Rank'] = law_order['Value'].rank(ascending=False, method='min'
law_order_sorted = law_order.sort_values(by='Rank').reset_index(drop=True

# Select the specified countries from the ranked datasets
countries_to_check = ['Germany', 'United States of America', 'Bangladesh'

# Get the rank and value for the specified countries in safety
safety_selected = safety_sorted[safety_sorted['Country_name'].isin(countr

# Get the rank and value for the specified countries in law order
law_order_selected = law_order_sorted[law_order_sorted['Country_name'].is

# Display the results
print("Rank of Selected Countries in Safety DataFrame:")
print(safety_selected)

print("\nRank of Selected Countries in Law Order DataFrame:")
print(law_order_selected)
```

```
Rank of Selected Countries in Safety DataFrame:
                Country_name  Value  Rank
33                 Bangladesh     78  32.0
34                      Japan     77  35.0
43                    Germany     74  44.0
54  United States of America     72  51.0
60                      India     70  61.0

Rank of Selected Countries in Law Order DataFrame:
                Country_name  Value  Rank
32                      Japan     86  32.0
36                    Germany     86  32.0
49                      India     83  48.0
61  United States of America     81  60.0
62                 Bangladesh     81  60.0
```

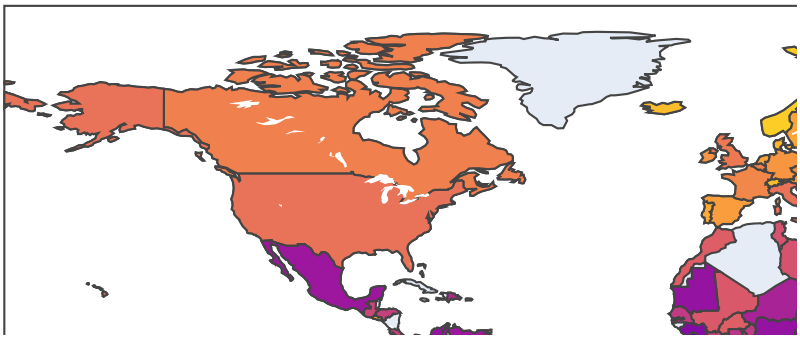
In [16]:

```
# Rank the countries based on their values
safety = safety.sort_values(by='Value', ascending=False).reset_index(drop
safety['Rank'] = safety.index + 1  # Add rank starting from 1

law_order = law_order.sort_values(by='Value', ascending=False).reset_inde
law_order['Rank'] = law_order.index + 1  # Add rank starting from 1

# Create a map for Perceived Safety
fig_safety = px.choropleth(safety,
    locations='Country_name',
    locationmode='country names',
    color='Value',
    color_continuous_scale=px.colors.sequential.Plasma,
    labels={'Value':'Perceived Safety'},
    title='World Map of Perceived Safety'
)

# Create a map for Law Order Index
fig_law_order = px.choropleth(law_order,
    locations='Country_name',
    locationmode='country names',
    color='Value',
    color_continuous_scale=px.colors.sequential.Plasma,
    labels={'Value':'Law Order Index'},
    title='World Map of Law Order Index'
)

# Update hover information to show country name, value, and rank
for df, fig in zip([safety, law_order], [fig_safety, fig_law_order]):
    fig.update_traces(hovertemplate="<b>%{location}</b><br>" +
                      "Score: %{z}<br>" +
                      "Rank: " + df['Rank'].astype(str) + "<extra></extra

# Show the figures
fig_safety.show()
fig_law_order.show()
```

World Map of Perceived Safety



World Map of Law Order Index

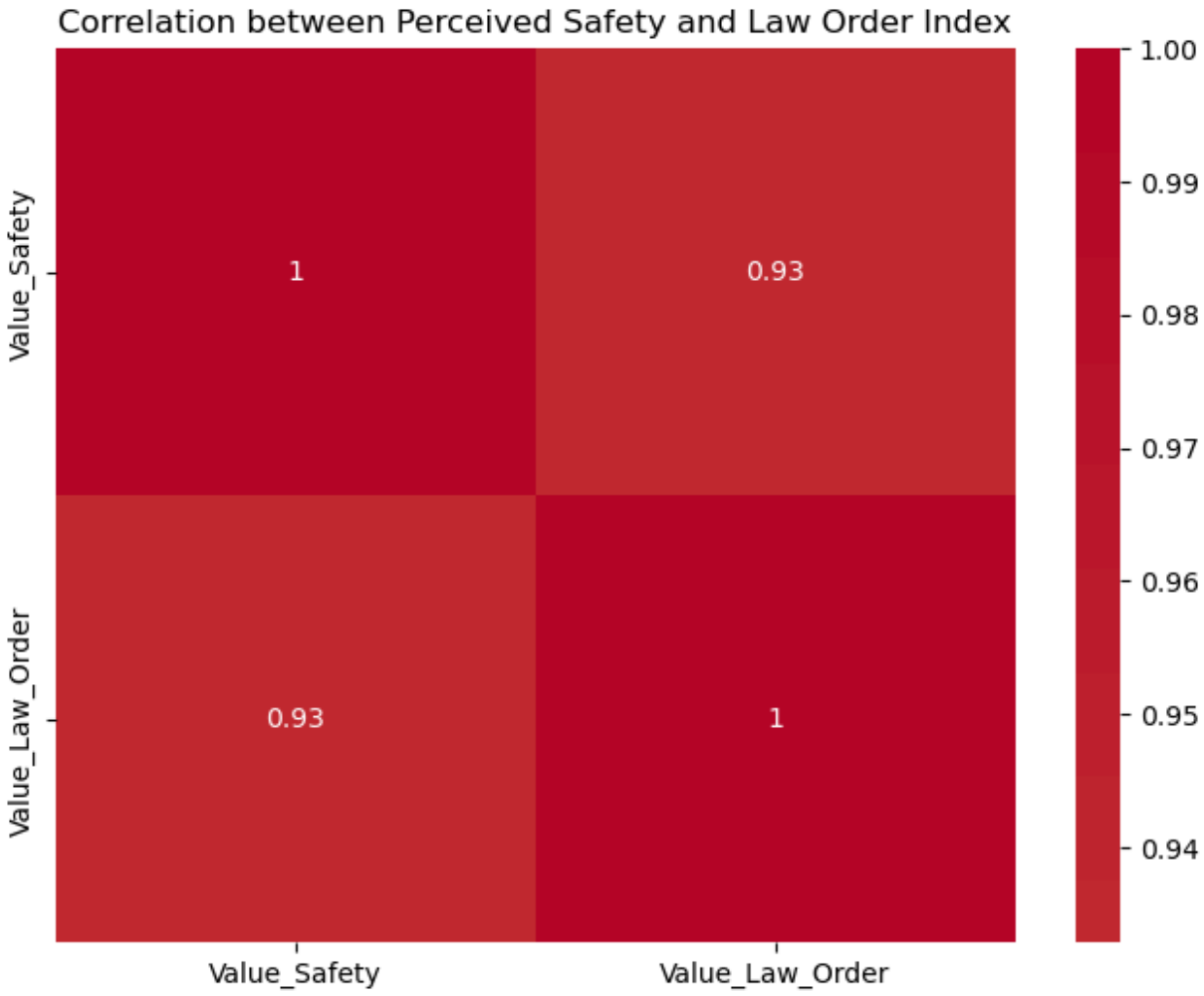


In [17]:
```
# Merge the datasets on 'Country_name'
merged_data = pd.merge(safety, law_order, on='Country_name', suffixes=('_
```

In [18]:
```
# Calculate correlation
correlation = merged_data[['Value_Safety', 'Value_Law_Order']].corr()
print(correlation)
```

```
                Value_Safety  Value_Law_Order
Value_Safety        1.000000         0.932855
Value_Law_Order     0.932855         1.000000
```

In [19]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Set up the matplotlib figure
plt.figure(figsize=(8, 6))

# Create a heatmap
sns.heatmap(correlation, annot=True, cmap='coolwarm', center=0)

# Show the plot
plt.title('Correlation between Perceived Safety and Law Order Index')
plt.show()
```



The heatmap shows the correlation between perceived safety and the law order index. The correlation coefficient is a measure of the strength and direction of the linear relationship between two variables. A value of 1 indicates a perfect positive correlation, meaning that the two variables increase or decrease together perfectly. A value of -1 indicates a perfect negative correlation, meaning that one variable increases as the other decreases perfectly. A value of 0 indicates no correlation.

In this heatmap, the correlation coefficient between perceived safety and the law order index is 0.93. This indicates a strong positive correlation between the two variables. This means that as perceived safety increases, the law order index also tends to increase, and vice versa. This suggests that people who perceive their area to be safer also tend to perceive the law and order in their area to be better.

In [20]:
```
# Create a new column to categorize countries
merged_data['Category'] = 'Medium'
merged_data.loc[merged_data['Value_Safety'] >= 80, 'Category'] = 'High Sa
merged_data.loc[merged_data['Value_Safety'] < 50, 'Category'] = 'Low Safe

# Group by category and analyze
grouped_analysis = merged_data.groupby('Category')[['Value_Safety', 'Valu
print(grouped_analysis)
```

```
              Value_Safety  Value_Law_Order
Category
High Safety      86.928571        89.607143
Low Safety       42.387097        63.677419
Medium           65.974684        79.075949
```

In [21]:

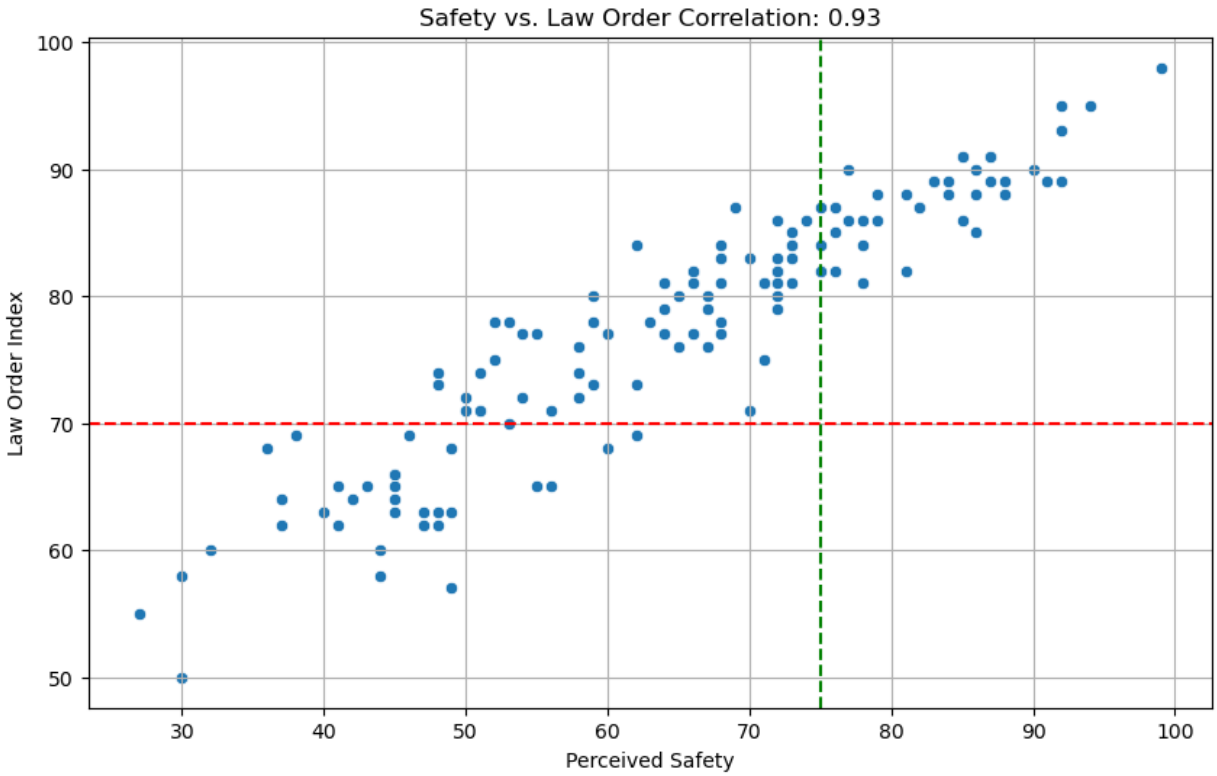
```python
# Define thresholds for high safety and low law order
high_safety_threshold = 80  # Example threshold for high safety
low_law_order_threshold = 50  # Example threshold for low law order

# Create a DataFrame combining safety and law order data
combined_data = pd.merge(safety, law_order, on='Country_name', suffixes=(

# Identify outliers
high_safety_low_law_order = combined_data[
    (combined_data['Value_Safety'] >= high_safety_threshold) &
    (combined_data['Value_Law_Order'] < low_law_order_threshold)
]

low_safety_high_law_order = combined_data[
    (combined_data['Value_Safety'] < high_safety_threshold) &
    (combined_data['Value_Law_Order'] >= low_law_order_threshold)
]

# Display results
print("Countries with High Safety but Low Law Order:")
print(high_safety_low_law_order[['Country_name', 'Value_Safety', 'Value_L

print("\nCountries with Low Safety but High Law Order:")
print(low_safety_high_law_order[['Country_name', 'Value_Safety', 'Value_L
```

```
Countries with High Safety but Low Law Order:
Empty DataFrame
Columns: [Country_name, Value_Safety, Value_Law_Order]
Index: []

Countries with Low Safety but High Law Order:
     Country_name  Value_Safety  Value_Law_Order
28         Sweden            79               88
29        Georgia            79               86
30        Ireland            78               86
31     Bangladesh            78               81
32         Serbia            78               84
..             ...           ...              ...
133         Chile            36               68
134      Botswana            32               60
135  South Africa            30               58
136       Liberia            30               50
137       Ecuador            27               55

[110 rows x 3 columns]
```

Summary of Findings

1. Countries with High Safety but Low Law Order:
• There are no countries identified in this category. This suggests a strong relationship where countries perceived as safe tend to also have higher law order indices.
2. Countries with Low Safety but High Law Order:
• A significant number of countries, 110 in total, fall into this category. Here are a few notable examples:
• Sweden: Safety Score = 79, Law Order Score = 88
• Georgia: Safety Score = 79, Law Order Score = 86
• Ireland: Safety Score = 78, Law Order Score = 86
• Bangladesh: Safety Score = 78, Law Order Score = 81
• Serbia: Safety Score = 78, Law Order Score = 84
• Additional countries with varying levels of safety and law order include Chile, Botswana, South Africa, Liberia, and Ecuador, which have low safety perceptions.

In [22]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Calculate correlation
correlation = combined_data[['Value_Safety', 'Value_Law_Order']].corr().i

# Create a scatter plot
plt.figure(figsize=(10, 6))
sns.scatterplot(data=combined_data, x='Value_Safety', y='Value_Law_Order'
plt.title(f'Safety vs. Law Order Correlation: {correlation:.2f}')
plt.xlabel('Perceived Safety')
plt.ylabel('Law Order Index')
plt.grid()
plt.axhline(y=70, color='r', linestyle='--')  # Example threshold line
plt.axvline(x=75, color='g', linestyle='--')  # Example threshold line
plt.show()
```



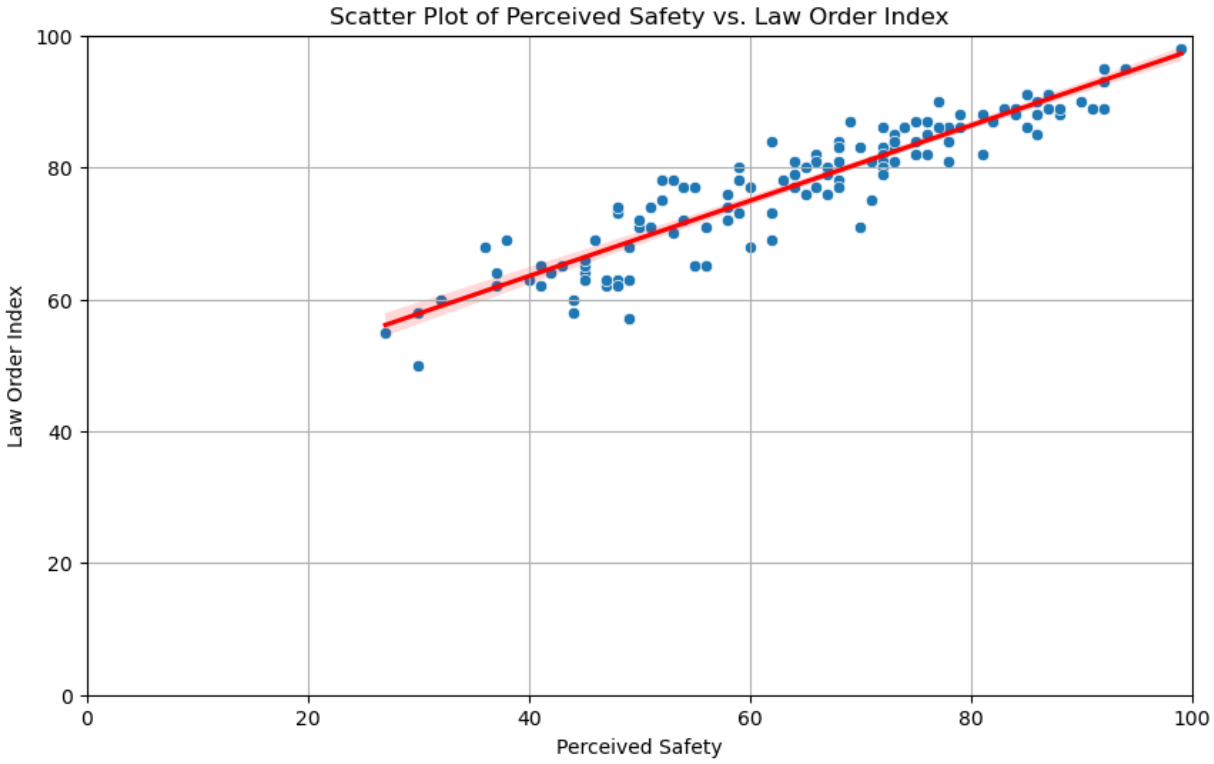
The scatter plot shows the relationship between perceived safety and the law order index. Each dot on the graph represents a data point, where the x-coordinate represents the perceived safety and the y-coordinate represents the law order index.

The title of the graph, "Safety vs. Law Order Correlation: 0.93," indicates a strong positive correlation between the two variables. This means that as perceived safety increases, the law order index also tends to increase, and vice versa.

The green vertical line at x=75 represents the average perceived safety. The red horizontal line at y=70 represents the average law order index. These lines help to visualize the distribution of the data points around the average values.

Overall, the scatter plot shows a clear upward trend, indicating a strong positive relationship between perceived safety and the law order index. This suggests that people who perceive their area to be safer also tend to perceive the law and order in their area to be better.

In [23]:
```
# Create a scatter plot
plt.figure(figsize=(10, 6))
sns.scatterplot(data=combined_data, x='Value_Safety', y='Value_Law_Order'

# Add a regression line
sns.regplot(data=combined_data, x='Value_Safety', y='Value_Law_Order', sc

# Customize the plot
plt.title('Scatter Plot of Perceived Safety vs. Law Order Index')
plt.xlabel('Perceived Safety')
plt.ylabel('Law Order Index')
plt.grid()
plt.xlim(0, 100)  # Set limits for better visibility
plt.ylim(0, 100)
plt.show()
```



The scatter plot shows the relationship between perceived safety and the law order index, with a regression line fitted to the data.

Key elements of the chart:

Scatter plot: The blue dots represent individual data points, where the x-coordinate is the perceived safety and the y-coordinate is the law order index. Regression line: The red line is a linear regression line that best fits the data points. It represents the trend between the two variables.

Confidence interval: The shaded area around the regression line represents the confidence interval, which indicates the range of values within which the true relationship between the two variables is likely to fall.

Axes: The x-axis represents perceived safety, and the y-axis represents the law order index. Interpretation:

The scatter plot shows a clear upward trend, indicating a positive relationship between perceived safety and the law order index. This suggests that as perceived safety increases, the law order index also tends to increase. The regression line further confirms this positive relationship, providing a linear model that can be used to predict the law order index based on perceived safety. The confidence interval shows the uncertainty associated with the regression line. A narrower confidence interval indicates a more precise prediction, while a wider confidence interval indicates more uncertainty. Overall, the chart suggests a strong positive relationship between perceived safety and the law order index. This information can be useful for understanding the factors that contribute to perceptions of safety and law and order in a community.

Conclusion:

In this project, we explored global safety and law and order indices to better understand the relationships between these two critical metrics. Through data preprocessing, ranking, and visualizations, we identified trends across countries and highlighted how these indices vary around the world.

Key insights were gained by analyzing the correlation between perceived safety and law order scores. While many countries showed a strong positive correlation between the two, we also identified several outliers, such as countries with high safety but lower law order scores and vice versa. These outliers provide valuable information for further investigation, as they may reveal socio-political, economic, or cultural factors that influence the perceived safety and law order enforcement.

Interactive visualizations, including world maps and scatter plots, provided a geographical perspective on global safety and law order dynamics. These visual tools enabled a clearer understanding of how these metrics differ by region and facilitated deeper exploration of specific countries.

Overall, this project demonstrated the power of data-driven analysis in understanding global safety perceptions and law enforcement effectiveness. The insights gained can inform policymakers, researchers, and international organizations in their efforts to improve safety and law order in different regions of the world. The project serves as a foundation for future work in exploring the underlying causes of these variations and how they relate to broader social and economic factors.