PerpsAdded

July 24, 2023

0.1 Terrorism by Country

0.1.1 a. Countries with the Most Terrorism Over the Years

[3]:

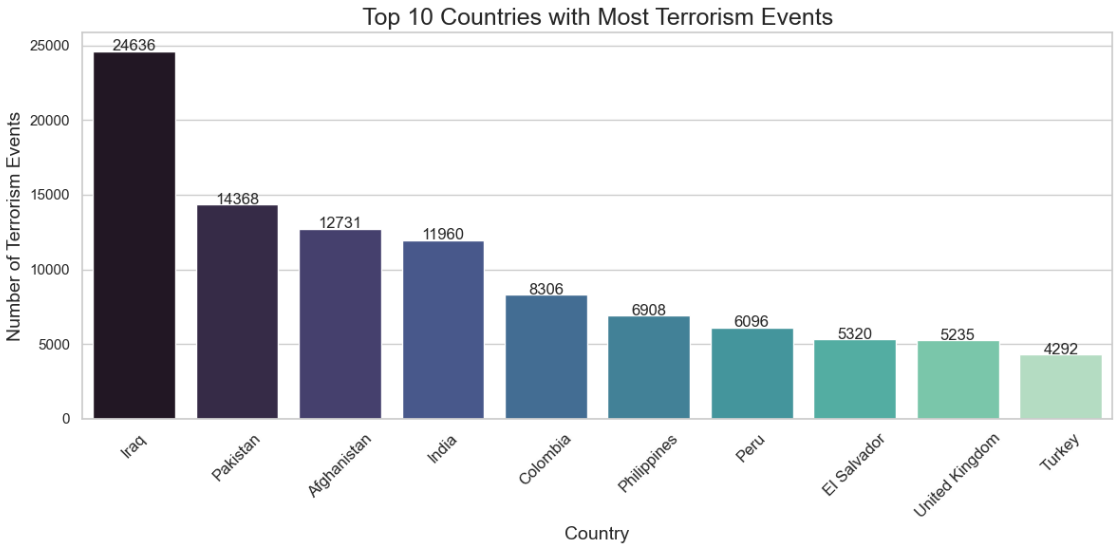
```
import pandas as pd

# Read the CSV file with a different encoding and set low_memory to False
data = pd.read_csv('PerpsAdded.csv', encoding='ISO-8859-1', low_memory=False,␣
 ↪sep=';')

# Group the data by 'country_txt' and count the number of events
country_counts = data['country_txt'].value_counts()

# Find the country with the most terrorism events
most_terrorism_country = country_counts.idxmax()
most_terrorism_count = country_counts.max()

print(f"The country with the most terrorism over the years is␣
 ↪{most_terrorism_country} with {most_terrorism_count} events.")
```

```
The country with the most terrorism over the years is Iraq with 24636 events.
```

[4]:

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

# Read the CSV file with a different encoding and set low_memory to False
data = pd.read_csv('PerpsAdded.csv', encoding='ISO-8859-1', low_memory=False,␣
 ↪sep=';')

# Group the data by 'country_txt' and count the number of events
country_counts = data['country_txt'].value_counts()

# Get the top 10 countries with the most terrorism events
top_10_countries = country_counts.head(10)

# Set the style of the plot
sns.set(style="whitegrid")
```

```
# Create the bar plot
plt.figure(figsize=(12, 6))
ax = sns.barplot(x=top_10_countries.index, y=top_10_countries.values,␣
 ↪palette="mako")

# Add labels and title
plt.xlabel('Country', fontsize=14)
plt.ylabel('Number of Terrorism Events', fontsize=14)
plt.title('Top 10 Countries with Most Terrorism Events', fontsize=18)

# Rotate x-axis labels for better readability
plt.xticks(rotation=45, fontsize=12)

# Add value labels above the bars
for i, v in enumerate(top_10_countries.values):
    ax.text(i, v + 25, str(v), horizontalalignment='center', fontsize=12)


plt.savefig('Top 10 Countries with Most Terrorism Events.png', dpi=300,␣
 ↪bbox_inches='tight')

# Display the plot
plt.tight_layout()
plt.show()

# Display a table with the findings
top_10_countries_table = top_10_countries.reset_index()
top_10_countries_table.columns = ['Country', 'Number of Terrorism Events']
print(top_10_countries_table)
```

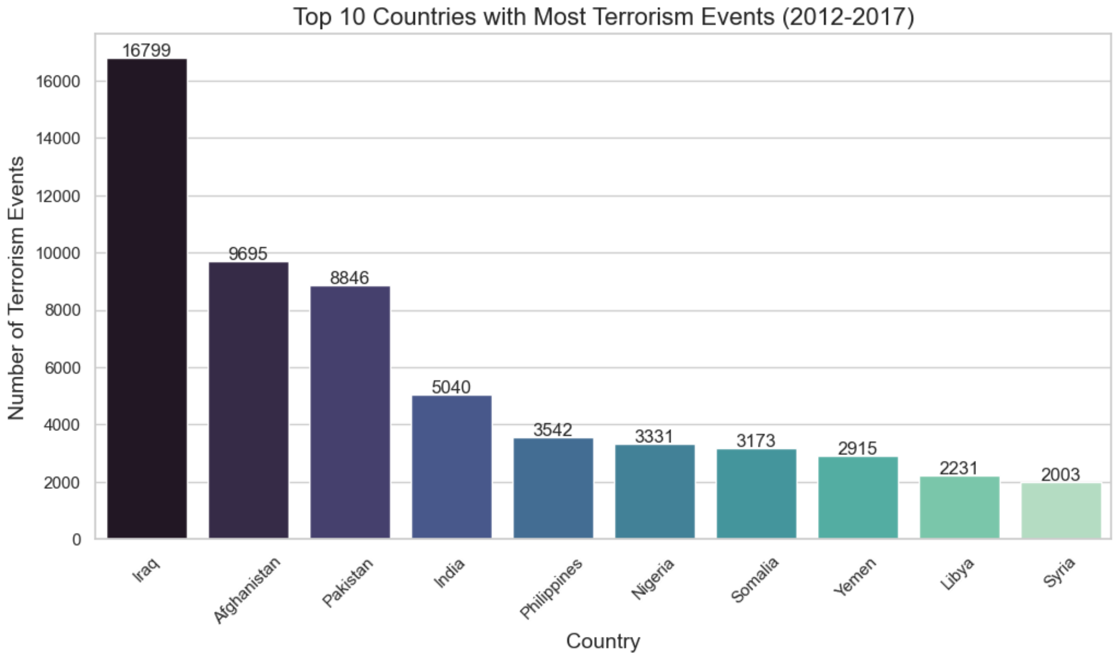


```
           Country  Number of Terrorism Events
0             Iraq                       24636
1         Pakistan                       14368
2      Afghanistan                       12731
3            India                       11960
4         Colombia                        8306
5      Philippines                        6908
6             Peru                        6096
7      El Salvador                        5320
8   United Kingdom                        5235
9           Turkey                        4292
```

0.1.2 b. Top 10 Countries with the Most Terrorism in Recent Times (2012-2017)

[5]:

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

data = pd.read_csv('PerpsAdded.csv', encoding='ISO-8859-1', low_memory=False,␣
 ↪sep=';')

# Convert the 'date' column to datetime format
data['date'] = pd.to_datetime(data['date'], format='%d.%m.%Y', errors='coerce')

# Filter the data for events between 2012 and 2017
recent_data = data[(data['date'].dt.year >= 2012) & (data['date'].dt.year <=␣
 ↪2017)]

# Group the data by 'country_txt' and count the number of events
recent_country_counts = recent_data['country_txt'].value_counts()

# Get the top 10 countries with the most terrorism events
top_10_recent_countries = recent_country_counts.head(10)

# Set the style of the plot
sns.set(style="whitegrid")

# Create the bar plot
plt.figure(figsize=(12, 6))
ax = sns.barplot(x=top_10_recent_countries.index, y=top_10_recent_countries.
 ↪values, palette="mako")

# Add labels and title
plt.xlabel('Country', fontsize=14)
plt.ylabel('Number of Terrorism Events', fontsize=14)
plt.title('Top 10 Countries with Most Terrorism Events (2012-2017)',␣
 ↪fontsize=16)

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

# Add values above the bars
for i, v in enumerate(top_10_recent_countries.values):
    ax.text(i, v + 50, str(v), horizontalalignment='center', fontsize=12)

plt.savefig('Top 10 Countries with Most Terrorism Events 2012-2017.png',␣
 ↪dpi=300, bbox_inches='tight')

# Display the plot
plt.show()

# Tabulate the findings
top_10_recent_countries_df = pd.DataFrame(top_10_recent_countries.reset_index())
top_10_recent_countries_df.columns = ['Country', 'Number of Terrorism Events']
print(top_10_recent_countries_df)
```



```
       Country  Number of Terrorism Events
0         Iraq                       16799
1  Afghanistan                        9695
2     Pakistan                        8846
3        India                        5040
```

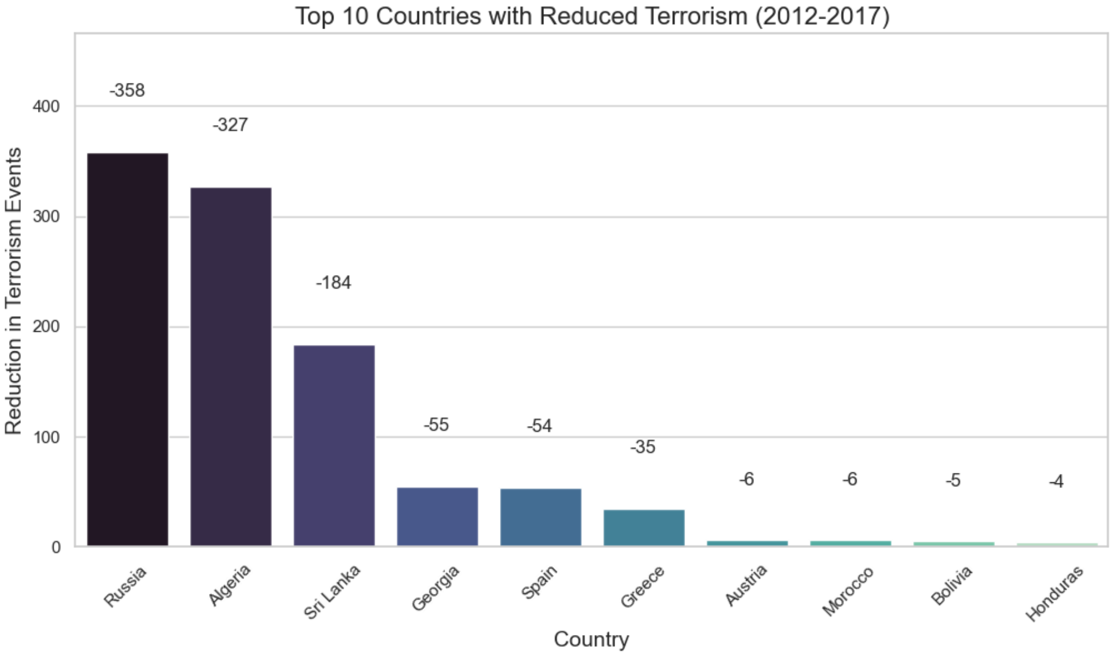
```
4  Philippines                      3542
5      Nigeria                      3331
6      Somalia                      3173
7        Yemen                      2915
8        Libya                      2231
9        Syria                      2003
```

0.1.3 c. Countries with Reduced Terrorism in Recent Times

[6]:

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

# Read the CSV file with a different encoding and set low_memory to False
data = pd.read_csv('PerpsAdded.csv', encoding='ISO-8859-1', low_memory=False,␣
 ↪sep=';')

# Convert the 'date' column to datetime format
data['date'] = pd.to_datetime(data['date'], format='%d.%m.%Y', errors='coerce')

# Filter the data for the period 2007-2011
data_2007_2011 = data[(data['date'].dt.year >= 2007) & (data['date'].dt.year <=␣
 ↪2011)]

# Group the data by 'country_txt' and count the number of events for 2007-2011
country_counts_2007_2011 = data_2007_2011['country_txt'].value_counts()

# Filter the data for the period 2012-2017
data_2012_2017 = data[(data['date'].dt.year >= 2012) & (data['date'].dt.year <=␣
 ↪2017)]

# Group the data by 'country_txt' and count the number of events for 2012-2017
country_counts_2012_2017 = data_2012_2017['country_txt'].value_counts()

# Calculate the difference between the two periods
difference = country_counts_2007_2011 - country_counts_2012_2017

# Drop NaN values and sort the difference in descending order
difference = difference.dropna().sort_values(ascending=False)

# Get the top 10 countries with the most reduced terrorism events
top_10_reduced_countries = difference.head(10)

# Set the style of the plot
sns.set(style="whitegrid")

# Create the bar plot
```

```
plt.figure(figsize=(12, 6))
ax = sns.barplot(x=top_10_reduced_countries.index, y=top_10_reduced_countries.
 ↪values, palette="mako")

# Add labels and title
plt.xlabel('Country', fontsize=14)
plt.ylabel('Reduction in Terrorism Events', fontsize=14)
plt.title('Top 10 Countries with Reduced Terrorism (2012-2017)', fontsize=16)

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

# Set the ylim to accommodate larger values
ax.set_ylim(0, max(top_10_reduced_countries.values) * 1.3)

# Add negative values above the bars
for i, v in enumerate(top_10_reduced_countries.values):
    ax.text(i, v + 50, f'-{int(v)}', horizontalalignment='center', fontsize=12)

plt.savefig('Top 10 Countries with Reduced Terrorism 2012-2017.png', dpi=300,␣
 ↪bbox_inches='tight')

# Display the plot
plt.show()

# Tabulate the findings
top_10_reduced_countries_df = pd.DataFrame(top_10_reduced_countries).
 ↪reset_index()
top_10_reduced_countries_df.columns = ['Country', 'Reduction in Terrorism␣
 ↪Events']
print(top_10_reduced_countries_df)
```



```
     Country  Reduction in Terrorism Events
0     Russia                          358.0
1    Algeria                          327.0
2  Sri Lanka                          184.0
3    Georgia                           55.0
4      Spain                           54.0
5     Greece                           35.0
6    Austria                            6.0
7    Morocco                            6.0
8    Bolivia                            5.0
9   Honduras                            4.0
```

0.2 Weapon Types and Terrorism

0.2.1 a. Most Common Weapon Types in Different Regions

[7]:
```
import pandas as pd

# Read the CSV file with a different encoding and set low_memory to False
data = pd.read_csv('PerpsAdded.csv', encoding='ISO-8859-1', low_memory=False,␣
 ↪sep=';')

# Group the data by 'region_txt' and 'weaptype1_txt', count the number of␣
 ↪events, and reset the index
weapons_by_region = data.groupby(['region_txt', 'weaptype1_txt']).size().
 ↪reset_index(name='count')
```

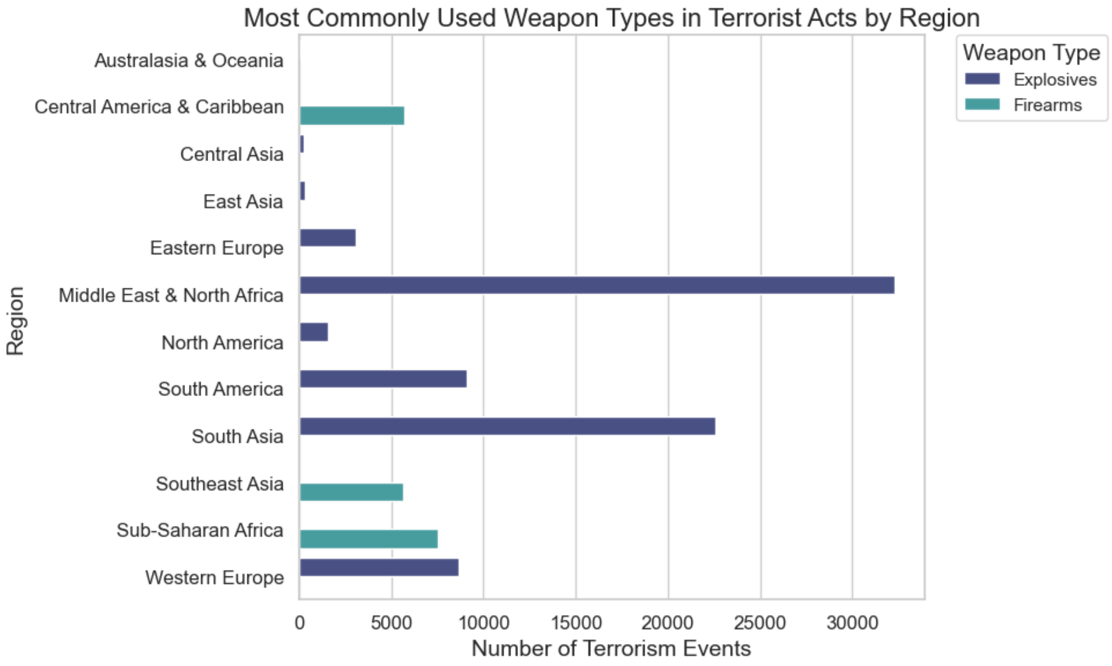
```
# Find the weapon type with the highest count in each region
idx = weapons_by_region.groupby(['region_txt'])['count'].transform(max) ==␣
 ↪weapons_by_region['count']
most_common_weapons = weapons_by_region[idx]

# Display the results
print(most_common_weapons)
```

```
                      region_txt weaptype1_txt  count
1          Australasia & Oceania    Explosives     80
10    Central America & Caribbean      Firearms   5679
17                  Central Asia    Explosives    254
25                     East Asia    Explosives    333
36                Eastern Europe    Explosives   3089
47    Middle East & North Africa    Explosives  32283
58                 North America    Explosives   1557
70                 South America    Explosives   9098
81                    South Asia    Explosives  22568
93                Southeast Asia      Firearms   5634
103           Sub-Saharan Africa      Firearms   7499
112               Western Europe    Explosives   8657
```

[8]:

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

# Read the CSV file with a different encoding and set low_memory to False
data = pd.read_csv('PerpsAdded.csv', encoding='ISO-8859-1', low_memory=False,␣
 ↪sep=';')

# Group by region and weapon type, and count the number of events
weapon_counts_by_region = data.groupby(['region_txt', 'weaptype1_txt']).size().
 ↪reset_index(name='count')

# Find the most commonly used weapon type for each region
most_common_weapons = weapon_counts_by_region.loc[weapon_counts_by_region.
 ↪groupby('region_txt')['count'].idxmax()]

# Set the style of the plot
sns.set(style="whitegrid")

# Create the horizontal bar plot
plt.figure(figsize=(10, 6))
ax = sns.barplot(x='count', y='region_txt', hue='weaptype1_txt',␣
 ↪data=most_common_weapons, palette='mako')
```

```
# Add labels and title
plt.xlabel('Number of Terrorism Events', fontsize=14)
plt.ylabel('Region', fontsize=14)
plt.title('Most Commonly Used Weapon Types in Terrorist Acts by Region',␣
 ↪fontsize=16)

# Adjust y-axis labels for better readability
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)

# Move the legend outside the plot
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0., title="Weapon␣
 ↪Type", title_fontsize=14)

plt.savefig('Most Commonly Used Weapon Types in Terrorist Acts by Region.png',␣
 ↪dpi=300, bbox_inches='tight')

# Display the plot
plt.tight_layout()
plt.show()
```

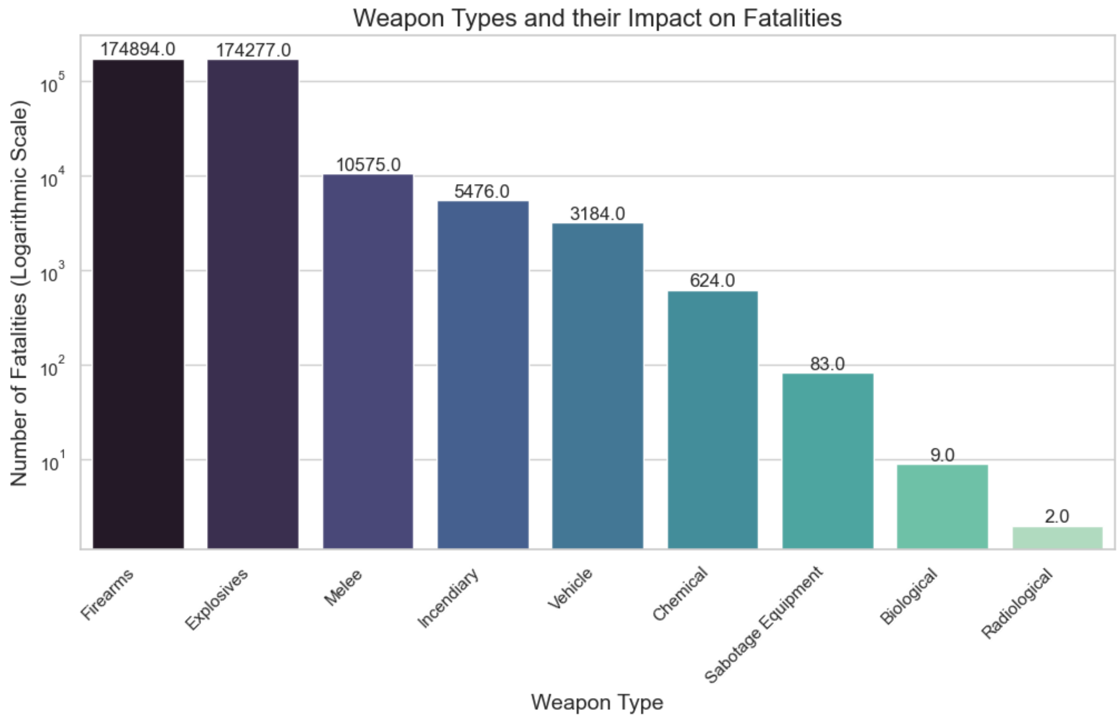


0.2.2 b. Weapon types and their impact on fatalities

[9]:

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

# Read the CSV file with a different encoding and set low_memory to False
data = pd.read_csv('PerpsAdded.csv', encoding='ISO-8859-1', low_memory=False,␣
 ↪sep=';')

# Calculate the number of fatalities for each weapon type
weapon_fatalities = data.groupby('weaptype1_txt')['nkill'].sum().
 ↪sort_values(ascending=False)

# Remove "Unknown", "Other", and "Fake Weapons"
weapon_fatalities = weapon_fatalities.drop(["Unknown", "Other", "Fake Weapons"])

# Rename 'Vehicle (not to include vehicle-borne explosives, i.e., car or truck␣
 ↪bombs)' to 'Vehicle'
weapon_fatalities.index = weapon_fatalities.index.map(lambda x: 'Vehicle' if x␣
 ↪== 'Vehicle (not to include vehicle-borne explosives, i.e., car or truck␣
 ↪bombs)' else x)

# Set the style of the plot
sns.set(style="whitegrid")

# Create the bar plot
plt.figure(figsize=(12, 6))
ax = sns.barplot(x=weapon_fatalities.index, y=weapon_fatalities.values,␣
 ↪palette="mako", log=True)

# Add labels and title
plt.xlabel('Weapon Type', fontsize=14)
plt.ylabel('Number of Fatalities (Logarithmic Scale)', fontsize=14)
plt.title('Weapon Types and their Impact on Fatalities', fontsize=16)

# Rotate x-axis labels for better readability
plt.xticks(rotation=45, ha='right')

# Display the values above the bars
for index, value in enumerate(weapon_fatalities.values):
    plt.text(index, value, str(value), ha='center', va='bottom', fontsize=12)

# Save the plot as a PNG file
plt.savefig('weapon_fatalities_modified.png', dpi=300, bbox_inches='tight')

# Display the plot
```

```
plt.show()

# Tabulate the findings
weapon_fatalities_table = pd.DataFrame({"Weapon Type": weapon_fatalities.index,
                                        "Number of Fatalities":␣
 ↪weapon_fatalities.values})
print(weapon_fatalities_table)
```



```
          Weapon Type  Number of Fatalities
0            Firearms              174894.0
1          Explosives              174277.0
2               Melee               10575.0
3          Incendiary                5476.0
4             Vehicle                3184.0
5            Chemical                 624.0
6  Sabotage Equipment                  83.0
7          Biological                   9.0
8        Radiological                   2.0
```

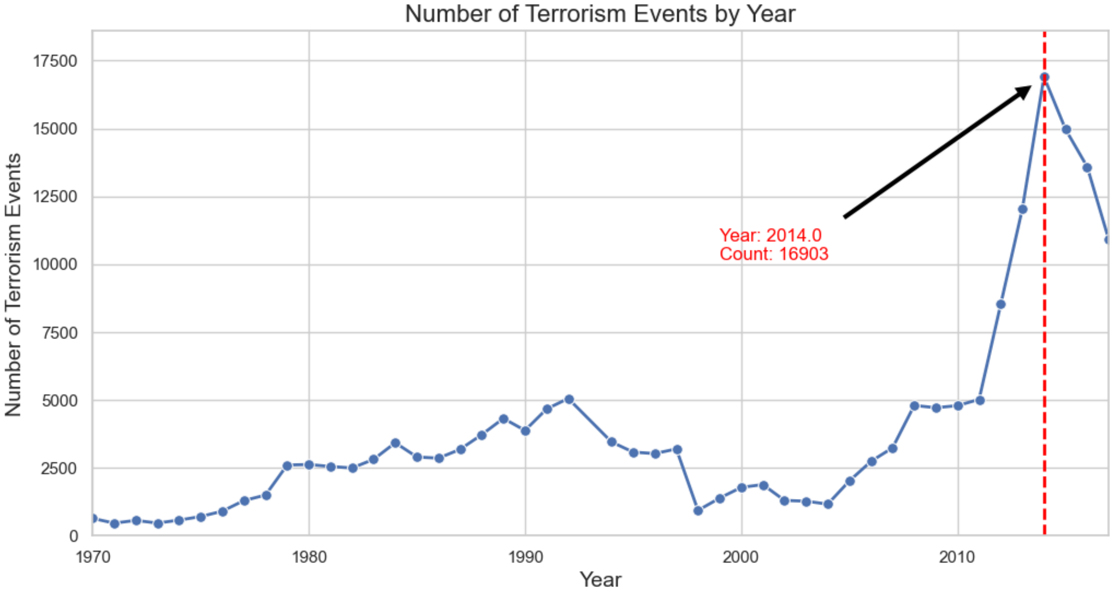
0.3 Terrorism Trends and Patterns

0.3.1 a. Worst Year for Terrorism

[10]:

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

# Read the CSV file with a different encoding and set low_memory to False
data = pd.read_csv('PerpsAdded.csv', encoding='ISO-8859-1', low_memory=False,␣
 ↪sep=';')

# Convert 'date' column to datetime objects
data['date'] = pd.to_datetime(data['date'], format='%d.%m.%Y', errors='coerce')

# Group the data by year and count the number of events
year_counts = data['date'].dt.year.value_counts().sort_index()

# Set the style of the plot
sns.set(style="whitegrid")

# Create the line plot
plt.figure(figsize=(12, 6))
ax = sns.lineplot(x=year_counts.index, y=year_counts.values, marker="o",␣
 ↪linewidth=2, markersize=7)

# Add labels and title
plt.xlabel('Year', fontsize=14)
plt.ylabel('Number of Terrorism Events', fontsize=14)
plt.title('Number of Terrorism Events by Year', fontsize=16)

# Set x-axis and y-axis limits
plt.xlim(year_counts.index.min(), year_counts.index.max())
plt.ylim(0, year_counts.values.max() * 1.1)

# Highlight the year with the highest number of terrorism events
max_year = year_counts.idxmax()
max_count = year_counts[max_year]
plt.axvline(max_year, color='red', linestyle='--', lw=2)
plt.annotate(f"Year: {max_year}\nCount: {max_count}", xy=(max_year, max_count),␣
 ↪xytext=(max_year - 15, max_count * 0.6),
             arrowprops=dict(facecolor='black', shrink=0.05), fontsize=12,␣
 ↪color='red')

plt.savefig('Number of Terrorism Events by Year.png', dpi=300,␣
 ↪bbox_inches='tight')

# Display the plot
```

```
plt.show()
```



[11]:

```
import pandas as pd

# Read the CSV file with a different encoding and set low_memory to False
data = pd.read_csv('PerpsAdded.csv', encoding='ISO-8859-1', low_memory=False,␣
  ↪sep=';')

# Convert 'date' column to datetime objects
data['date'] = pd.to_datetime(data['date'], format='%d.%m.%Y', errors='coerce')

# Group the data by year and count the number of events
year_counts = data['date'].dt.year.value_counts().sort_index()

# Create a DataFrame with the year_counts data
year_counts_df = pd.DataFrame({'Year': year_counts.index, 'Number of Terrorism␣
  ↪Events': year_counts.values})

# Display the DataFrame
print(year_counts_df)
```

```
      Year  Number of Terrorism Events
0   1970.0                         636
1   1971.0                         461
2   1972.0                         561
3   1973.0                         463
4   1974.0                         572
```

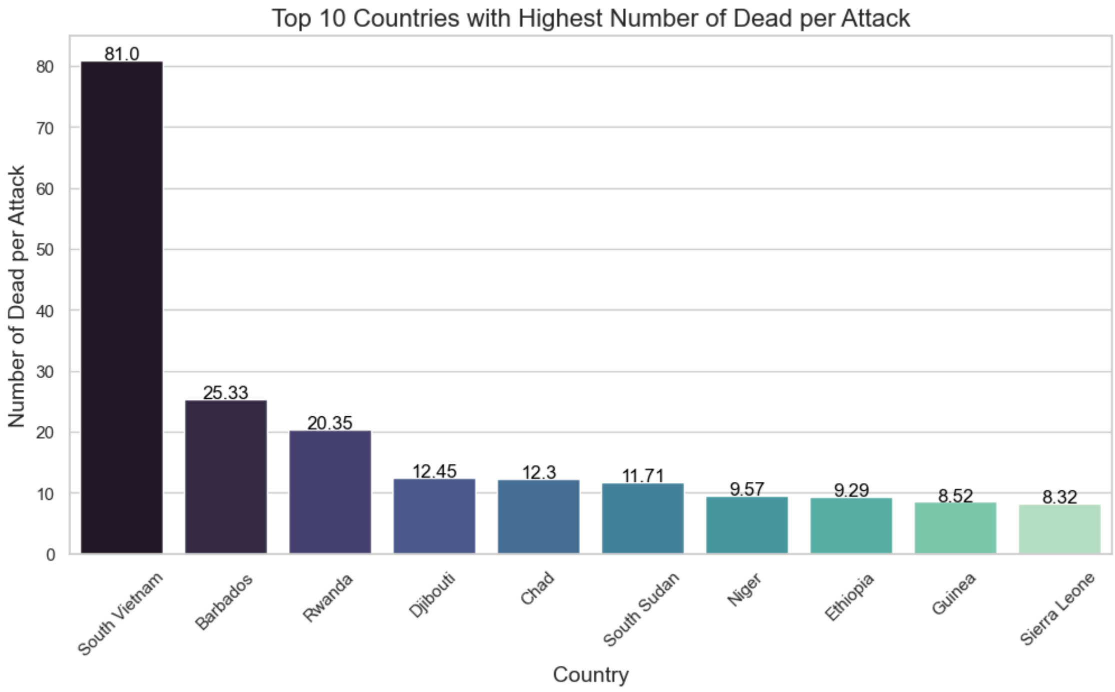
```
5   1975.0                        703
6   1976.0                        900
7   1977.0                       1292
8   1978.0                       1493
9   1979.0                       2601
10  1980.0                       2620
11  1981.0                       2548
12  1982.0                       2492
13  1983.0                       2810
14  1984.0                       3422
15  1985.0                       2897
16  1986.0                       2854
17  1987.0                       3179
18  1988.0                       3720
19  1989.0                       4319
20  1990.0                       3876
21  1991.0                       4666
22  1992.0                       5053
23  1994.0                       3452
24  1995.0                       3077
25  1996.0                       3023
26  1997.0                       3194
27  1998.0                        930
28  1999.0                       1382
29  2000.0                       1778
30  2001.0                       1881
31  2002.0                       1298
32  2003.0                       1269
33  2004.0                       1161
34  2005.0                       2015
35  2006.0                       2736
36  2007.0                       3237
37  2008.0                       4797
38  2009.0                       4712
39  2010.0                       4789
40  2011.0                       5018
41  2012.0                       8522
42  2013.0                      12036
43  2014.0                      16903
44  2015.0                      14965
45  2016.0                      13587
46  2017.0                      10900
```

0.3.2 b. Number of Deaths per Attack

[12]:

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

# Read the CSV file with a different encoding and set low_memory to False
data = pd.read_csv('PerpsAdded.csv', encoding='ISO-8859-1', low_memory=False,␣
 ↪sep=';')

# Calculate total number of deaths per country
deaths_per_country = data.groupby('country_txt')['nkill'].sum()

# Calculate total number of attacks per country
attacks_per_country = data['country_txt'].value_counts()

# Calculate number of dead per attack for each country
dead_per_attack = deaths_per_country / attacks_per_country

# Get the top 10 countries with the highest number of dead per attack
top_10_countries = dead_per_attack.sort_values(ascending=False).head(10)

# Set the style of the plot
sns.set(style="whitegrid")

# Create the bar plot
plt.figure(figsize=(12, 6))
ax = sns.barplot(x=top_10_countries.index, y=top_10_countries.values,␣
 ↪palette="mako")

# Add labels and title
plt.xlabel('Country', fontsize=14)
plt.ylabel('Number of Dead per Attack', fontsize=14)
plt.title('Top 10 Countries with Highest Number of Dead per Attack',␣
 ↪fontsize=16)

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

# Add values on top of the bars
for i, v in enumerate(top_10_countries.values):
    ax.text(i, v + 0.1, str(round(v, 2)), ha='center', fontsize=12,␣
 ↪color='black')


plt.savefig('Top 10 Countries with Highest Number of Dead per Attack.png',␣
 ↪dpi=300, bbox_inches='tight')
```

```python
# Display the plot
plt.show()

# Tabulate the findings
top_10_countries_table = top_10_countries.reset_index()
top_10_countries_table.columns = ['Country', 'Number of Dead per Attack']
print(top_10_countries_table)
```

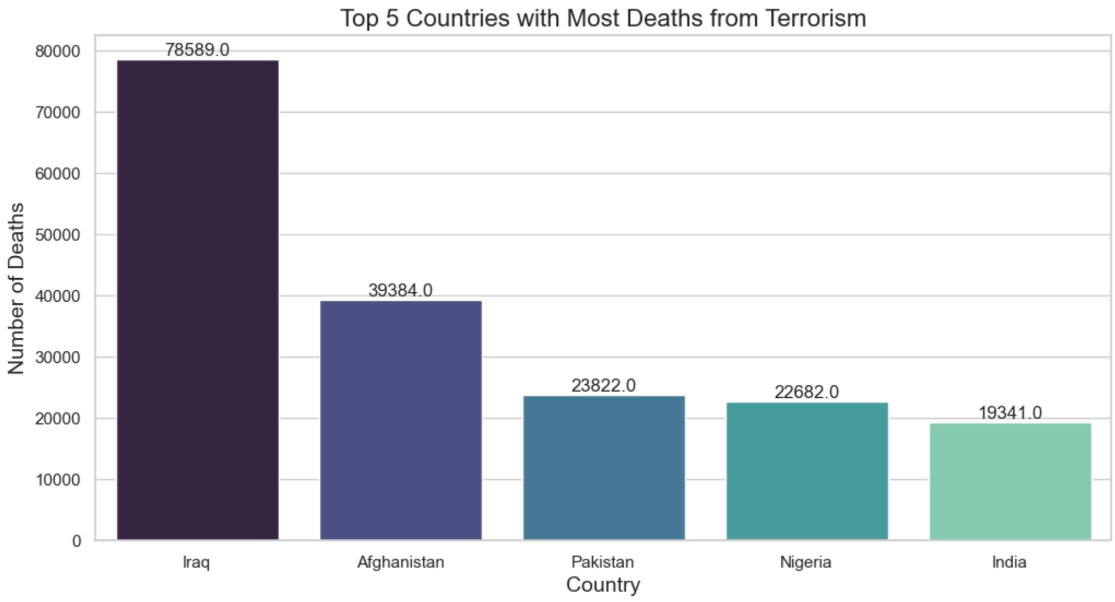


```
         Country  Number of Dead per Attack
0  South Vietnam                  81.000000
1       Barbados                  25.333333
2         Rwanda                  20.352201
3       Djibouti                  12.454545
4           Chad                  12.296703
5    South Sudan                  11.706667
6          Niger                   9.571429
7       Ethiopia                   9.289474
8         Guinea                   8.520000
9   Sierra Leone                   8.316832
```

0.3.3 c. Top 5 Countries with the Most Deaths

[13]:

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

# Read the CSV file with a different encoding and set low_memory to False
data = pd.read_csv('PerpsAdded.csv', encoding='ISO-8859-1', low_memory=False,␣
 ↪sep=';')

# Calculate the top 5 countries with the most deaths from terrorism
top_5_countries_deaths = data.groupby('country_txt')['nkill'].sum().nlargest(5)

# Now you can use the previous code to create the bar plot and save it as a PNG␣
 ↪file
# Set the style of the plot
sns.set(style="whitegrid")

# Create the bar plot
plt.figure(figsize=(12, 6))
ax = sns.barplot(x=top_5_countries_deaths.index, y=top_5_countries_deaths.
 ↪values, palette="mako")

# Add labels and title
plt.xlabel('Country', fontsize=14)
plt.ylabel('Number of Deaths', fontsize=14)
plt.title('Top 5 Countries with Most Deaths from Terrorism', fontsize=16)

# Display the values above the bars
for index, value in enumerate(top_5_countries_deaths.values):
    plt.text(index, value, str(value), ha='center', va='bottom', fontsize=12)

# Save the plot as a PNG file
plt.savefig('top_5_countries_deaths.png', dpi=300, bbox_inches='tight')

# Display the plot
plt.show()

# Display findings in a tabular format
top_5_deaths_table = pd.DataFrame({"Country": top_5_countries_deaths.index,
                                   "Number of Deaths": top_5_countries_deaths.
 ↪values})
print(top_5_deaths_table)
```

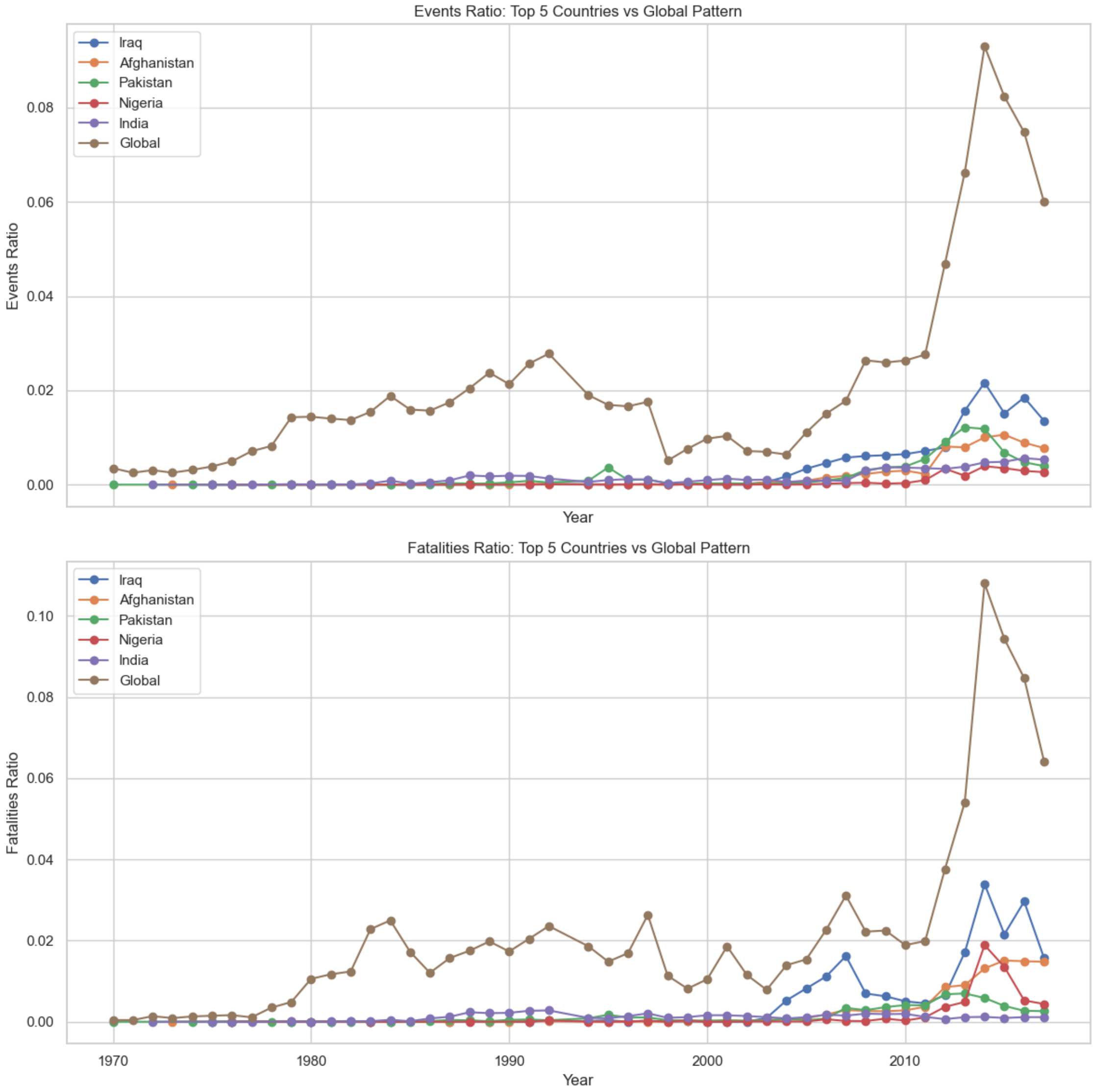


```
       Country  Number of Deaths
0         Iraq           78589.0
1  Afghanistan           39384.0
2     Pakistan           23822.0
3      Nigeria           22682.0
4        India           19341.0
```

0.3.4 d. Comparative Analysis of Terrorism Intensity: Top 5 Countries and Global Patterns

[14]:

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

data['date'] = pd.to_datetime(data['date'], format='%d.%m.%Y')  # Convert the␣
 ↪'date' column to datetime format

global_events = data['ï»¿eventid'].count()
global_fatalities = data['nkill'].sum()

global_events_ratio = data.groupby(data['date'].dt.year)['ï»¿eventid'].count() /
 ↪ global_events
global_fatalities_ratio = data.groupby(data['date'].dt.year)['nkill'].sum() /␣
 ↪global_fatalities

global_ratios = pd.DataFrame({'Year': global_events_ratio.index,
                              'Country': 'Global',
                              'Events Ratio': global_events_ratio.values,
                              'Fatalities Ratio': global_fatalities_ratio.
 ↪values})

top_5_ratios = pd.DataFrame(columns=['Year', 'Country', 'Events Ratio',␣
 ↪'Fatalities Ratio'])

for country in top_5_countries_deaths.index:
    country_data = data[data['country_txt'] == country]
    country_events = country_data.groupby(country_data['date'].dt.
 ↪year)['ï»¿eventid'].count()
    country_fatalities = country_data.groupby(country_data['date'].dt.
 ↪year)['nkill'].sum()

    country_events_ratio = country_events / global_events
    country_fatalities_ratio = country_fatalities / global_fatalities

    country_ratios = pd.DataFrame({'Year': country_events_ratio.index,
                                   'Country': country,
                                   'Events Ratio': country_events_ratio.values,
                                   'Fatalities Ratio': country_fatalities_ratio.
 ↪values})
    top_5_ratios = pd.concat([top_5_ratios, country_ratios], ignore_index=True)

all_ratios = pd.concat([top_5_ratios, global_ratios], ignore_index=True)

fig, axes = plt.subplots(2, 1, figsize=(12, 12), sharex=True)

for country in all_ratios['Country'].unique():
    country_data = all_ratios[all_ratios['Country'] == country]
    axes[0].plot(country_data['Year'], country_data['Events Ratio'],␣
 ↪marker='o', label=country)
    axes[1].plot(country_data['Year'], country_data['Fatalities Ratio'],␣
 ↪marker='o', label=country)

axes[0].set(title='Events Ratio: Top 5 Countries vs Global Pattern',␣
 ↪ylabel='Events Ratio')
axes[1].set(title='Fatalities Ratio: Top 5 Countries vs Global Pattern',␣
 ↪xlabel='Year', ylabel='Fatalities Ratio')
axes[0].set_xlabel('Year')
axes[1].set_xlabel('Year')

axes[0].legend(loc='upper left')
axes[1].legend(loc='upper left')

plt.tight_layout()
plt.savefig('terrorism_ratios_comparison.png', dpi=300, bbox_inches='tight')
plt.show()
```



```
[15]: # Tabulate the findings
      events_ratio_table = top_5_ratios.pivot_table(index=['Year'],␣
       ↪columns='Country', values='Events Ratio', margins=True)
      events_ratio_table.fillna(0, inplace=True)

      fatalities_ratio_table = top_5_ratios.pivot_table(index=['Year'],␣
       ↪columns='Country', values='Fatalities Ratio', margins=True)
      fatalities_ratio_table.fillna(0, inplace=True)

      # Display the tables
```

```
print("Top 5 Countries and Global Patterns\n")
print("Events Ratio:\n", events_ratio_table)
print("\nFatalities Ratio:\n", fatalities_ratio_table)
```

```
Top 5 Countries and Global Patterns

Events Ratio:
 Country  Afghanistan     India      Iraq   Nigeria  Pakistan       All
Year
1970.0       0.000000  0.000000  0.000000  0.000000  0.000006  0.000006
1972.0       0.000000  0.000006  0.000000  0.000000  0.000000  0.000006
1973.0       0.000006  0.000000  0.000000  0.000000  0.000000  0.000006
1974.0       0.000000  0.000000  0.000000  0.000000  0.000011  0.000011
1975.0       0.000000  0.000006  0.000006  0.000000  0.000006  0.000006
1976.0       0.000000  0.000006  0.000017  0.000006  0.000017  0.000011
1977.0       0.000000  0.000006  0.000000  0.000000  0.000000  0.000006
1978.0       0.000000  0.000000  0.000000  0.000000  0.000011  0.000011
1979.0       0.000017  0.000110  0.000011  0.000000  0.000039  0.000044
1980.0       0.000000  0.000055  0.000028  0.000006  0.000006  0.000023
1981.0       0.000000  0.000088  0.000011  0.000000  0.000022  0.000040
1982.0       0.000000  0.000066  0.000028  0.000000  0.000022  0.000039
1983.0       0.000000  0.000259  0.000017  0.000017  0.000050  0.000085
1984.0       0.000000  0.000859  0.000011  0.000000  0.000017  0.000295
1985.0       0.000000  0.000215  0.000000  0.000000  0.000011  0.000113
1986.0       0.000000  0.000528  0.000000  0.000000  0.000132  0.000330
1987.0       0.000006  0.000914  0.000017  0.000000  0.000330  0.000316
1988.0       0.000061  0.001970  0.000022  0.000011  0.000242  0.000461
1989.0       0.000055  0.001778  0.000022  0.000000  0.000248  0.000526
1990.0       0.000011  0.001910  0.000000  0.000000  0.000479  0.000800
1991.0       0.000165  0.001866  0.000017  0.000017  0.000826  0.000578
1992.0       0.000198  0.001304  0.000193  0.000061  0.000462  0.000444
1994.0       0.000050  0.000589  0.000099  0.000044  0.000848  0.000326
1995.0       0.000033  0.000985  0.000094  0.000006  0.003666  0.000957
1996.0       0.000022  0.001172  0.000066  0.000061  0.000991  0.000462
1997.0       0.000006  0.001062  0.000116  0.000110  0.001134  0.000485
1998.0       0.000006  0.000336  0.000039  0.000011  0.000204  0.000119
1999.0       0.000050  0.000616  0.000066  0.000099  0.000215  0.000209
2000.0       0.000077  0.000969  0.000055  0.000033  0.000270  0.000281
2001.0       0.000077  0.001282  0.000017  0.000028  0.000292  0.000339
2002.0       0.000204  0.001002  0.000033  0.000033  0.000253  0.000305
2003.0       0.000550  0.001079  0.000561  0.000050  0.000160  0.000480
2004.0       0.000484  0.000594  0.001756  0.000033  0.000369  0.000647
2005.0       0.000853  0.000804  0.003390  0.000050  0.000424  0.001104
2006.0       0.001541  0.000908  0.004585  0.000204  0.000897  0.001627
2007.0       0.001866  0.000820  0.005757  0.000336  0.001431  0.002042
2008.0       0.002273  0.002934  0.006087  0.000418  0.003110  0.002964
2009.0       0.002757  0.003699  0.006258  0.000231  0.003671  0.003323
2010.0       0.002983  0.003627  0.006484  0.000347  0.003853  0.003459
```

```
2011.0      0.002290  0.003506  0.007166  0.000963  0.005471  0.003879
2012.0      0.008085  0.003363  0.007909  0.003390  0.009103  0.006370
2013.0      0.007942  0.003820  0.015697  0.001904  0.012191  0.008311
2014.0      0.010039  0.004733  0.021647  0.003930  0.011839  0.010438
2015.0      0.010611  0.004865  0.015141  0.003511  0.006841  0.008194
2016.0      0.008900  0.005641  0.018493  0.002934  0.004755  0.008145
2017.0      0.007782  0.005317  0.013572  0.002664  0.003957  0.006659
All         0.002187  0.001563  0.003662  0.000717  0.001834  0.002019

Fatalities Ratio:
 Country  Afghanistan     India      Iraq   Nigeria  Pakistan       All
Year
1970.0      0.000000  0.000000  0.000000  0.000000  0.000010  0.000010
1972.0      0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
1973.0      0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
1974.0      0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
1975.0      0.000000  0.000010  0.000000  0.000000  0.000002  0.000004
1976.0      0.000000  0.000000  0.000029  0.000007  0.000002  0.000010
1977.0      0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
1978.0      0.000000  0.000000  0.000000  0.000000  0.000012  0.000012
1979.0      0.000129  0.000075  0.000002  0.000000  0.000015  0.000055
1980.0      0.000000  0.000041  0.000015  0.000000  0.000041  0.000024
1981.0      0.000000  0.000058  0.000000  0.000000  0.000010  0.000023
1982.0      0.000000  0.000143  0.000233  0.000000  0.000007  0.000128
1983.0      0.000000  0.000143  0.000002  0.000002  0.000024  0.000043
1984.0      0.000000  0.000469  0.000000  0.000000  0.000029  0.000166
1985.0      0.000000  0.000124  0.000000  0.000000  0.000000  0.000062
1986.0      0.000000  0.000826  0.000000  0.000000  0.000146  0.000486
1987.0      0.000000  0.001229  0.000000  0.000000  0.000444  0.000418
1988.0      0.000311  0.002345  0.000104  0.000010  0.000330  0.000620
1989.0      0.000024  0.002122  0.000000  0.000000  0.000136  0.000571
1990.0      0.000029  0.002200  0.000000  0.000000  0.000456  0.000895
1991.0      0.000165  0.002702  0.000002  0.000024  0.000561  0.000691
1992.0      0.000119  0.002797  0.000284  0.000328  0.000362  0.000778
1994.0      0.000053  0.000944  0.000182  0.000036  0.000859  0.000415
1995.0      0.000012  0.000876  0.000379  0.000002  0.001729  0.000600
1996.0      0.000075  0.001382  0.000027  0.000058  0.001027  0.000514
1997.0      0.000010  0.002071  0.000257  0.000260  0.001076  0.000735
1998.0      0.000019  0.000966  0.000046  0.000022  0.000367  0.000284
1999.0      0.000095  0.001139  0.000078  0.000325  0.000308  0.000389
2000.0      0.000092  0.001615  0.000024  0.000000  0.000286  0.000404
2001.0      0.000422  0.001602  0.000022  0.000007  0.000265  0.000464
2002.0      0.000180  0.001413  0.000024  0.000068  0.000255  0.000388
2003.0      0.000396  0.001146  0.000949  0.000068  0.000289  0.000570
2004.0      0.000668  0.000811  0.005266  0.000100  0.000738  0.001517
2005.0      0.000891  0.001131  0.008214  0.000046  0.000369  0.002130
2006.0      0.001775  0.001738  0.011152  0.000617  0.000762  0.003209
2007.0      0.002892  0.001520  0.016187  0.000199  0.003414  0.004842
```

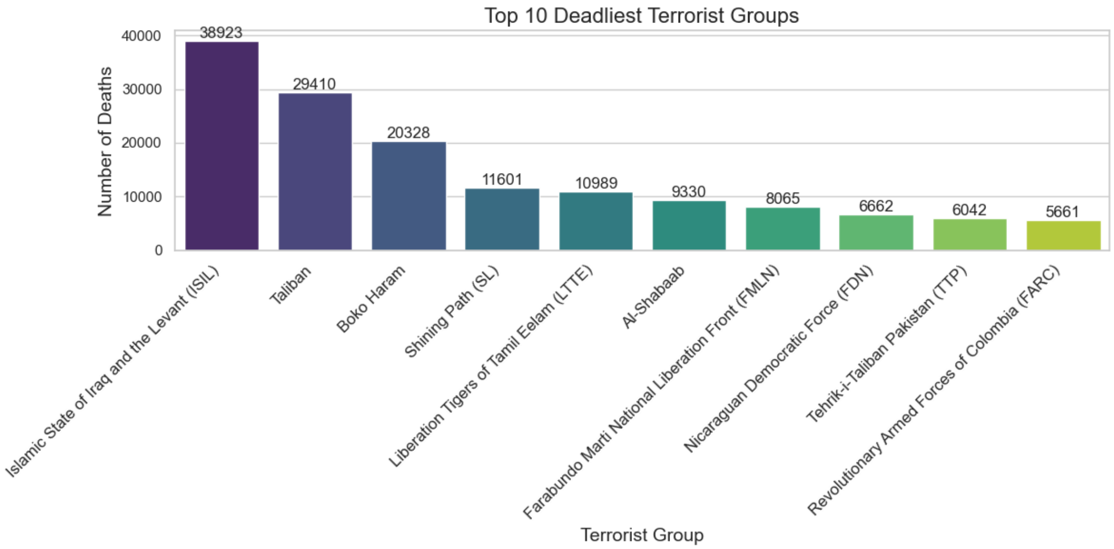
```
2008.0      0.002651  0.001998  0.006954  0.000175  0.002855  0.002927
2009.0      0.002583  0.001879  0.006276  0.000767  0.003610  0.003023
2010.0      0.002809  0.001945  0.005033  0.000284  0.004113  0.002837
2011.0      0.003703  0.001197  0.004533  0.001085  0.004038  0.002911
2012.0      0.008549  0.000641  0.006522  0.003661  0.006759  0.005226
2013.0      0.009005  0.001134  0.017095  0.004890  0.006980  0.007821
2014.0      0.013145  0.001190  0.033906  0.018892  0.005859  0.014598
2015.0      0.015092  0.000940  0.021572  0.013497  0.003904  0.011001
2016.0      0.014913  0.001134  0.029638  0.005257  0.002702  0.010729
2017.0      0.014791  0.001129  0.015723  0.004382  0.002612  0.007728
All         0.002987  0.001115  0.005155  0.001836  0.001343  0.002424
```

0.4 Deadliest Terrorist Groups

[20]:

```
# Dropping the rows where 'gname' is Unknown
data = data[data['gname'] != 'Unknown']

# Group the dataset
grouped_data = data.groupby('gname')['nkill'].sum()

# Sort the data in descending order
top_deadliest_groups = grouped_data.sort_values(ascending=False).head(10)

# Set a style for the plot
sns.set_style("whitegrid")

# Plot the bar chart
plt.figure(figsize=(12, 6))
ax = sns.barplot(x=top_deadliest_groups.index, y=top_deadliest_groups.values,␣
 ↪palette='viridis')
plt.title('Top 10 Deadliest Terrorist Groups', fontsize=16)
plt.xlabel('Terrorist Group', fontsize=14)
plt.ylabel('Number of Deaths', fontsize=14)
plt.xticks(rotation=45, ha='right', fontsize=12)

# Add values on top of each bar
for i, p in enumerate(ax.patches):
    ax.annotate(f"{p.get_height():.0f}", (p.get_x() + p.get_width() / 2., p.
 ↪get_height()), ha='center', va='bottom', fontsize=12)

plt.tight_layout()
plt.savefig('Top 10 Deadliest Terrorist Groups.png', dpi=300,␣
 ↪bbox_inches='tight')
plt.show()

# Tabulate the findings
```

```
top_deadliest_groups_df = top_deadliest_groups.reset_index().
 ↪rename(columns={'nkill': 'Number of Deaths'})
print(top_deadliest_groups_df)
```



```
                                              gname  Number of Deaths
0       Islamic State of Iraq and the Levant (ISIL)           38923.0
1                                           Taliban           29410.0
2                                        Boko Haram           20328.0
3                                 Shining Path (SL)           11601.0
4           Liberation Tigers of Tamil Eelam (LTTE)           10989.0
5                                        Al-Shabaab            9330.0
6  Farabundo Marti National Liberation Front (FMLN)            8065.0
7                 Nicaraguan Democratic Force (FDN)            6662.0
8                   Tehrik-i-Taliban Pakistan (TTP)            6042.0
9     Revolutionary Armed Forces of Colombia (FARC)            5661.0
```

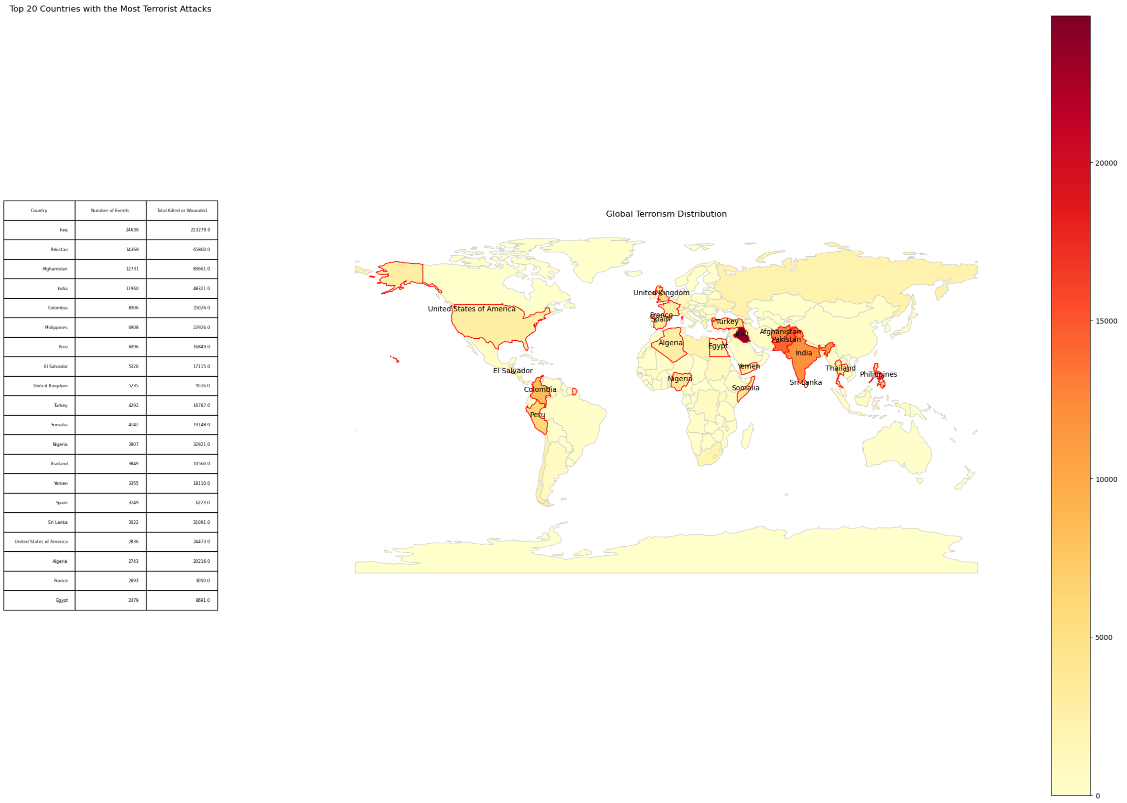
0.5 Map

[2]:

```
import geopandas as gpd
import matplotlib.pyplot as plt
import pandas as pd

# Read the CSV file with a different encoding and set low_memory to False
df = pd.read_csv('PerpsAdded.csv', encoding='ISO-8859-1', low_memory=False,␣
 ↪sep=';')

def map_country_names(df):
    country_mappings = {
        'East Germany (GDR)': 'Germany',
        'West Germany (FRG)': 'Germany',
        'South Vietnam': 'Vietnam',
        'Rhodesia': 'Zimbabwe',
        'Soviet Union': 'Russia',
        'North Yemen': 'Yemen',
        'South Yemen': 'Yemen',
        "People's Republic of the Congo": 'Dem. Rep. Congo',
        'Republic of the Congo': 'Dem. Rep. Congo',
        'Zaire': 'Dem. Rep. Congo',
        'Democratic Republic of the Congo': 'Dem. Rep. Congo',
        'Czechoslovakia': 'Czech Republic',
        'Serbia-Montenegro': 'Serbia',
        'United States': 'United States of America'
    }
    df['country_txt'] = df['country_txt'].replace(country_mappings)
    return df

df = map_country_names(df)


# Aggregate the number of terror events per country
country_terror_counts = df['country_txt'].value_counts().reset_index()
country_terror_counts.columns = ['Country', 'Number of Events']

country_killed_wounded = df.groupby('country_txt')[['nkill', 'nwound']].sum().
 ↪reset_index()
country_killed_wounded['Total Killed or Wounded'] =␣
 ↪country_killed_wounded['nkill'] + country_killed_wounded['nwound']
country_killed_wounded.drop(['nkill', 'nwound'], axis=1, inplace=True)
country_killed_wounded.columns = ['Country', 'Total Killed or Wounded']

# Merge the terror event counts and killed/wounded counts per country
country_stats = pd.merge(country_terror_counts, country_killed_wounded,␣
 ↪on='Country')

# Load the world map dataset
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))

# Merge the terror data with the world map dataset
world_terror = world.merge(country_terror_counts, left_on='name',␣
 ↪right_on='Country', how='left')

# Fill missing values with 0
world_terror['Number of Events'].fillna(0, inplace=True)

top_20 = country_terror_counts.head(20)
top_20_map = world_terror[world_terror['Country'].isin(top_20['Country'])]
```

```
# Normalize the data for visualization
world_terror['count_normalized'] = world_terror['Number of Events'] /␣
 ↪world_terror['Number of Events'].max()

# Define the colormap
cmap = plt.get_cmap('YlOrRd')

# Create a subplot with a table and a map
fig, (ax1, ax2) = plt.subplots(1, 2, gridspec_kw={'width_ratios': [1, 4]},␣
 ↪figsize=(30, 20))

# Plot the world map with terror events
world_terror.plot(column='count_normalized', cmap=cmap, linewidth=0.8,␣
 ↪edgecolor='0.8', ax=ax2)

# Add colorbar
sm = plt.cm.ScalarMappable(cmap=cmap, norm=plt.Normalize(vmin=0,␣
 ↪vmax=world_terror['Number of Events'].max()))
sm._A = []
cbar = fig.colorbar(sm, ax=ax2)

# Set the title
ax2.set_title('Global Terrorism Distribution')

# Remove axes
ax2.set_axis_off()

# Find the midpoint of the yellow color range in the colormap
yellow_midpoint = (cmap(0.5)[0] + cmap(0.5)[1] + cmap(0.5)[2]) / 3

# Add the names of the top 20 countries to the map
for idx, row in top_20_map.iterrows():
    plt.annotate(text=row['Country'], xy=row['geometry'].centroid.coords[0],␣
 ↪ha='center', fontsize=10, color='black')

# Plot the top 20 countries in a different color and larger size
top_20_map.plot(ax=ax2, color='none', edgecolor='red', linewidth=1.1,␣
 ↪markersize=30)

top_20 = country_stats.head(20)
table = ax1.table(cellText=top_20.values, colLabels=top_20.columns,␣
 ↪loc='center')

# Set the font size of the cells in the table
table_cells = table.get_celld()
```

```python
for i in range(1, len(top_20) + 1):
    for j in range(len(top_20.columns)):
        cell = table_cells[(i, j)]
        cell.set_text_props(fontsize=40)
        cell.set_height(0.025)

# Set the header cell height to match the other cells
for j in range(len(top_20.columns)):
    header_cell = table_cells[(0, j)]
    header_cell.set_height(0.025)
    header_cell.set_text_props(fontsize=40)

# Set the title of the table
ax1.set_title('Top 20 Countries with the Most Terrorist Attacks')

# Remove axes
ax1.axis('off')

# Save the figure
plt.savefig('global_terrorism_map_with_country_names_and_table.png', dpi=300)

plt.show()
```



Top 20 Countries with the Most Terrorist Attacks

Country	Number of Events	Total Killed or Wounded
Iraq	24636	213279.0
Pakistan	14368	65860.0
Afghanistan	12731	83661.0
India	11960	48321.0
Colombia	8306	25026.0
Philippines	6908	22926.0
Peru	6096	16849.0
El Salvador	5320	17115.0
United Kingdom	5235	9516.0
Turkey	4292	16787.0
Somalia	4142	19148.0
Nigeria	3907	32921.0
Thailand	3849	10560.0
Yemen	3355	18110.0
Spain	3249	6223.0
Sri Lanka	3022	31091.0
United States of America	2836	24473.0
Algeria	2743	20216.0
France	2693	3050.0
Egypt	2479	8691.0

Global Terrorism Distribution

[ ]:

```
```