Terrorism : A Global Problem

Terrorism stands as one of the greatest threats the humanity has ever faced, impacting nations, communities, and individuals worldwide. In 2023, the world saw a troubling increase in the lethality of terrorist attacks, with **deaths rising by 22% to 8,352**(Institute for Economics & Peace, 2024). This is the highest since 2017. The increase in fatalities occurred even as the **overall number of terrorist incidents declined (by 22%).**

The average number of deaths per attack **rose from 1.6 in 2022 to 2.5 in 2023**. This rise uncovers the growing brutality and precision of terrorist operations. Terrorism is becoming increasingly **lethal, posing a significant threat not only to life but also to global stability** and security.

The geographic landscape of terrorism has also undergone significant shifts. **Sub-Saharan Africa has emerged as a new hotspot for terrorist activity**, accounting for nearly half of all terrorism-related deaths in 2023(Institute for Economics & Peace, 2024). For example, Burkina Faso saw a 68% increase in terrorism deaths. The region's instability has been amplified by factors such as political unrest, economic deprivation, and weak governance structures(UN Security Council, 2020). The terrorist groups are often found to be exploiting these aspects to enhance their 'influence'.

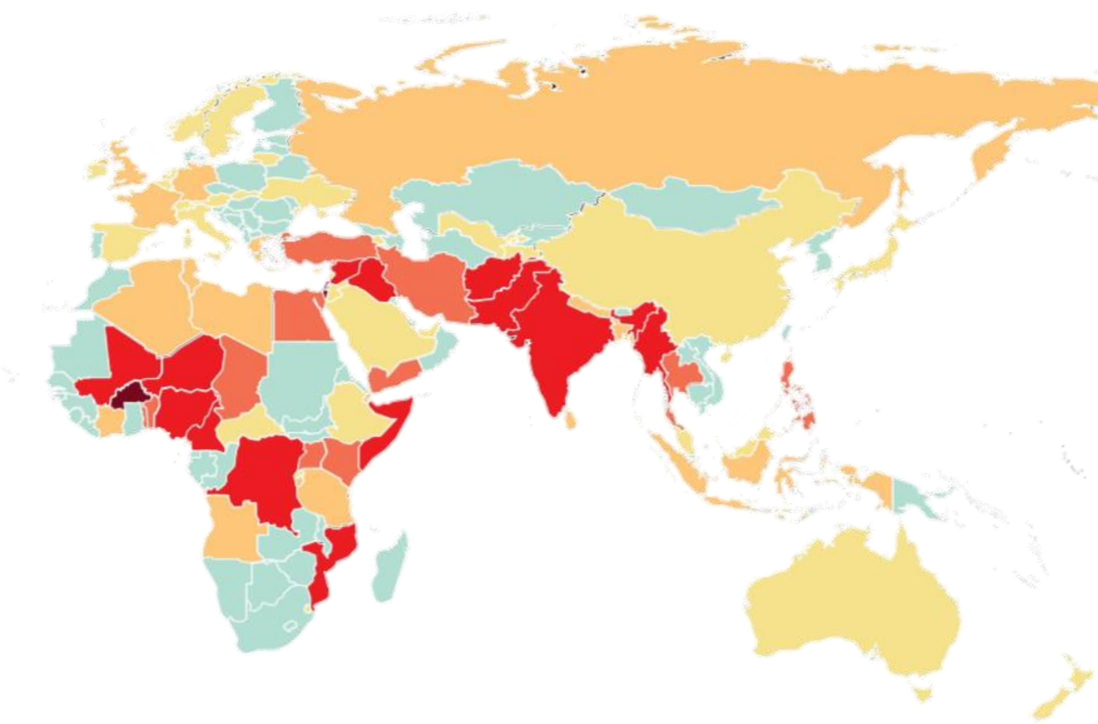


Fig 1 Global Terrorism Summary – Source: GTI 2024

Our Approach to the Problem

The broader impact of terrorism extends beyond the immediate loss of life and physical destruction. It weakens social cohesion, disrupts economic development, and creates long-lasting psychological trauma. The economic costs are also staggering. Between 2000 and 2018, the world lost **approximately $ 655 billion to terrorism**(Bardwell & Iqbal, 2021). The global economic impact of terrorism is estimated to **exceed $100 billion annually**, driven by losses in tourism, foreign investment, and increased security expenditures (Institute for Economics & Peace, 2024).

Furthermore, terrorism's unpredictable nature (and its capacity to incite fear) make it a uniquely potent threat, capable of destabilizing entire regions and complicating international relations. This volatility was particularly evident following the **October 7th attacks in Israel, the deadliest since 9/11.** A significantly heightened tensions were flet in the Middle East and beyond ((Institute for Economics & Peace, 2024)).

Addressing terrorism requires a nuanced understanding of its evolving dynamics and the socio-political contexts in which it thrives. It is highly critical **for the global community to enhance cooperation and improve counter-terrorism strategies**. The challenge is monumental. In this paper, we aim to **visualize and critically analyze the global terrorism dataset from 1970 – 2017**. The sole purpose of this analysis would be to understand the global incidence patterns, try and identify underlying trends and assist in creating a better understanding of this global illness for humanity.

At the core, we believe that **with coordinated efforts** and a commitment to peace, it is possible to mitigate the impact of terrorism and work towards a more secure world. The **first step begins with formulating a proper and better understanding of the problem.** Therefore, we visualize and analyze!

Introduction to the (Global Terrorism) Dataset

The ‘**Global Terrorism Database’ (GTD)** is an open-source database that collects information on terrorist attacks worldwide. The dataset has a comprehensive collection of records spanning from 1970 through 2017. This dataset is maintained by the ‘**National Consortium for the Study of Terrorism and Responses to Terrorism’** (START, 2024) at the University of Maryland.

Funded by various agencies, including the U.S. Department of State and the U.S. Department of Homeland Security, the GTD serves as an invaluable resource for researchers, and analysts seeking to understand the global patterns (causes, and consequences) of terrorism. The **dataset includes more than 180,000 terrorist incidents**, making it one of the most comprehensive sources of its kind.

Origin of the Dataset

This dataset’s origin dates to the early 1970s. The initial data collection was started by the ‘**Pinkerton Global Intelligence Service**’ (PGIS), a private security agency. Over time, the dataset was integrated with data collected by the ‘**Center for Terrorism and Intelligence Studies**’ (CETIS) and the ‘**Institute for the Study of Violent Groups**’ (ISVG). **START assumed full responsibility in 2011**. The dataset has evolved to become a critical tool for understanding the **evolution of terrorism over the past five decades**.

The GTD defines a terrorist attack as "the threatened or actual use of illegal force and violence by a non-state actor to attain a political, economic, religious, or social goal through fear, coercion, or intimidation"(START, 2024) . This definition, along with **inclusion criteria such as ‘intentionality’, ‘violence or threat of violence’, and ‘sub-national perpetrators’, ensures the dataset's focus remains on terrorist activities distinct from other forms** of political violence or crime.

Scope of the Dataset

The database includes over **100 variables** that cover a wide range of information, including the date and location of the incident, the types of weapons used, the groups or individuals responsible, the number of casualties among others. The inclusion of such granular data allows for **robust analyses** of terrorism's patterns, trends, and impacts globally.

The GTD covers incidents from **1970 to 2017, with a notable exclusion of data for the year 1993.** This gap resulted from a loss of the original data, which could not be fully recovered despite several attempts.

Methodology and Data Collection

The GTD relies on a rigorous data collection methodology, **combining automated data retrieval techniques with manual validation by subject matter experts**. Data sources include **over one million media articles published daily worldwide**(START, 2024), filtered through advanced 'natural language processing' (NLP) and 'machine learning algorithms' to identify relevant-incidents. The database prioritizes information from high-quality sources that are independent, externally verifiable, and primary.

Data collection for **the GTD is an ongoing process**, with efforts made to continually update and refine the dataset. The **transition to a fully START-based collection in 2011 marked a significant shift in the data's comprehensiveness** and internal consistency.

Data Types

The GTD dataset contains a mix of **categorical, numeric, and text data types**, each serving specific analytical purposes:

- **Categorical Data**: Used extensively for classification purposes, such as identifying **attack types, weapon types, target types, and perpetrator groups**. These variables allow categorization and comparison of incidents based on different criteria.
- **Numeric Data**: Includes date-related variables (**'year', 'month', 'day'**), as well as quantitative measures such as the number of casualties ('**nkill'**, '**nwound'**) and economic damage ('**propvalue'**). These variables are necessary for **quantitative analyses, trend analysis and predictive analytics**.
- **Text Data**: Provides descriptive information and qualitative context, such as incident summaries (**summary**), motives (**motive**), and additional notes (**addnotes**). Text data enriches the dataset by offering narrative details that complement quantitative analysis, providing a fuller picture of each incident.

Significance of the Dataset

The broad coverage of GTD dataset (47 years, from 1970 to 2017) allows **for longitudinal studies that can track changes in terrorism over time**. The inclusion of both domestic and international incidents **ensures a holistic view of terrorism's impact** across different regions and contexts. Its extensive coverage, rigorous methodology, and detailed coding provide a robust foundation for various analytical and research works.

Data Preprocessing

Dataset Composition

The GTD dataset is structured around a series of variables that capture various aspects of terrorist incidents. The composition of the data can be generalized into following categories

1. **Event Identification and Date Variables (numeric data)**
   - **'eventid'**:
     - A unique 12-digit identifier for each incident, formatted as **'yyyymmddxxxx'**.
     - 1st eight digits represent the **incident's date**, while the last four are a sequential number indicating the incident's order on that date.
   - **'iyear', 'imonth', 'iday'**:
     - represent the **year, month, and day** of the incident.
     - provide a precise temporal context for each incident
   - **'approxdate'**:
     - text variable, **used when the exact date is unknown**
     - offers an approximate time frame for the incident.
2. **Location Variables (categorical data)**
   - **'country', 'country_txt'**:
     - indicate the country where the incident occurred, both numerically and textually.
     - essential for geographic analyses and to examine regional patterns and hotspots of terrorist activity.
   - **region, region_txt**:
     - like the country variables, these categorize incidents by broader geographic regions.
     - divides **the world into 12 regions**, enabling macro-level analysis of terrorism's geographic-distribution.
   - **provstate, city**:
     - provide more granular location details, specifying the state, province and city where the incident occurred.
     - allow for highly localized studies**, identifying specific urban centers with high concentrations of terrorist activity**.
3. **Attack Information Variables (text and numeric data)**
   - **attacktype1, attacktype1_txt**:
     - **describe the type of attack** (e.g., assassination, assault, bombing etc) in both **numeric and textual formats**.
   - **weaptype1, weaptype1_txt**:

    - provide information about **the primary weapon type used** in the attack (e.g., firearms, explosives, incendiary devices).
    - useful for understanding **the methods employed by terrorist** groups and their access to various weaponry.

4. **Target and Perpetrator Variables (text and categorical data)**
    - **targtype1, targtype1_txt, gname**:
        - provide information on the target type (e.g., military, police, civilians) and the group or individual responsible for the attack.
        - assist in analyzing target preferences, **the strategic objectives** and their intended impact (message).
    - **gsubname, motive**:
        - **offer additional context about the perpetrators**, including subgroup names and stated motives.
        - helpful in **understanding intra-group dynamics** and the ideological or political drivers.
5. **Casualties and Consequences Variables (numeric data)**
    - **nkill, nwound**:
        - **record the number of fatalities** and injuries resulting from each attack.
        - critical for **assessing the human cost of terrorism** and identifying particularly lethal incidents.
    - **propvalue**:
        - the **estimated property damage in U.S. dollars**, providing a measure of the economic impact of terrorist activities.
        - valuable for **understanding the broader economic costs of terrorism**, beyond the immediate loss of life.
6. **Additional Information Variables (text data)**
    - **addnotes, scite1, scite2, scite3**:
        - **capture supplementary information** about the incident, including citations to source documents.
        - could be useful in deeper level analysis

Dataset Exploration

Loading the data is the first step in any data analysis process. It provides the initial dataset on which all subsequent operations will be performed. Dataset was loaded into Google Colab as the first step.

```
data = pd.read_csv('/content/drive/MyDrive/tsmdata/globaltsmdata.csv', encoding="ISO-8859-1");
data.head()
```

The original composition of the dataset was observed to be as below:

	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	country_txt	region	...	addnotes	scite1	scite2	scite3	dbsource	INT_LOG	INT_IDEO	INT_MISC	INT_ANY	related
0	197000000001	1970	7	2	NaN	0	NaN	58	Dominican Republic	2	...	NaN	NaN	NaN	NaN	PGIS	0	0	0	0	NaN
1	197000000002	1970	0	0	NaN	0	NaN	130	Mexico	1	...	NaN	NaN	NaN	NaN	PGIS	0	1	1	1	NaN
2	197001000001	1970	1	0	NaN	0	NaN	160	Philippines	5	...	NaN	NaN	NaN	NaN	PGIS	-9	-9	1	1	NaN
3	197001000002	1970	1	0	NaN	0	NaN	78	Greece	8	...	NaN	NaN	NaN	NaN	PGIS	-9	-9	1	1	NaN
4	197001000003	1970	1	0	NaN	0	NaN	101	Japan	4	...	NaN	NaN	NaN	NaN	PGIS	-9	-9	1	1	NaN

5 rows × 135 columns

Knowing the unique values helps **in identifying categorical variables, understanding data distribution, and detecting anomalies** or potential issues like inconsistent data entry(Yin et al., 2001). For each column of data in the data-frame, unique entries occurring on the dataset was explored using a simple looping statement.

```
for i in data.columns:
    print(i)
    print(f"{data[i].unique()}\n")
```

Before commencing data processing, the structure of the data was verified to reflect the one discussed in the above section.

Dataset Cleaning and Preprocessing

Dropping columns with excessive missing values **simplifies the dataset, reduces noise, and ensures that subsequent analyses are performed on cleaner, more reliable data**(Yin et al., 2001). Following the data validation, to ensure the analysis was focused on the relevant features, certain columns were dropped.

```
# Calculate the threshold
threshold = len(data) * 0.5
```

All columns **with over 50% null entry were identified** and dropped from the data-frame.

```
# Identify columns to drop
columns_to_drop = data.columns[data.isnull().mean() > 0.5]
```

A total **of 77 columns** were identified to contain sparse data (that is less than 50% entries). These columns were dropped from our dataset.

```
Columns dropped due to more than 50% missing values:
Index(['approxdate', 'resolution', 'location', 'alternative',
       'alternative_txt', 'attacktype2', 'attacktype2_txt', 'attacktype3',
       'attacktype3_txt', 'targtype2', 'targtype2_txt', 'targsubtype2',
       'targsubtype2_txt', 'corp2', 'target2', 'natlty2', 'natlty2_txt',
       'targtype3', 'targtype3_txt', 'targsubtype3', 'targsubtype3_txt',
       'corp3', 'target3', 'natlty3', 'natlty3_txt', 'gsubname', 'gname2',
       'gsubname2', 'gname3', 'gsubname3', 'motive', 'guncertain2',
       'guncertain3', 'claimmode', 'claimmode_txt', 'claim2', 'claimmode2',
       'claimmode2_txt', 'claim3', 'claimmode3', 'claimmode3_txt', 'compclaim',
       'weaptype2', 'weaptype2_txt', 'weapsubtype2', 'weapsubtype2_txt',
       'weaptype3', 'weaptype3_txt', 'weapsubtype3', 'weapsubtype3_txt',
       'weaptype4', 'weaptype4_txt', 'weapsubtype4', 'weapsubtype4_txt',
       'propextent', 'propextent_txt', 'propvalue', 'propcomment', 'nhostkid',
       'nhostkidus', 'nhours', 'ndays', 'divert', 'kidhijcountry', 'ransom',
       'ransomamt', 'ransomamtus', 'ransompaid', 'ransompaidus', 'ransomnote',
       'hostkidoutcome', 'hostkidoutcome_txt', 'nreleased', 'addnotes',
       'scite2', 'scite3', 'related'],
      dtype='object')
```

The new data-frame after the above procedure evolved to take the following form:

	eventid	iyear	imonth	iday	extended	country	country_txt	region	region_txt	provstate	city	latitude	longitude	specificity	vicinity	summary	crit1	crit2	crit3
0	197000000001	1970	7	2	0	58	Dominican Republic	2	Central America & Caribbean	NaN	Santo Domingo	18.456792	-69.951164	1.0	0	NaN	1	1	1
1	197000000002	1970	0	0	0	130	Mexico	1	North America	Federal	Mexico city	19.371887	-99.086624	1.0	0	NaN	1	1	1
2	197001000001	1970	1	0	0	160	Philippines	5	Southeast Asia	Tarlac	Unknown	15.478598	120.599741	4.0	0	NaN	1	1	1
3	197001000002	1970	1	0	0	78	Greece	8	Western Europe	Attica	Athens	37.997490	23.762728	1.0	0	NaN	1	1	1
4	197001000003	1970	1	0	0	101	Japan	4	East Asia	Fukouka	Fukouka	33.580412	130.396361	1.0	0	NaN	1	1	1

Next up, the headers for different **columns were renamed to give them a more descriptive form**.

```python
# Renaming the columns to be more descriptive
renamed_columns = {
    'iyear': 'year',
    'imonth': 'month',
    'iday': 'day',
    'country': 'country_code',
    'country_txt': 'country_name',
    'region': 'region_code',
    'region_txt': 'region_name',
    'attacktype1': 'attack_type_code',
    'attacktype1_txt': 'attack_type_name',
    'int_log': 'international_logistics',
    'int_ideo': 'international_ideological',
    'int_misc': 'international_miscellaneous',
    'int_any': 'international_any',
    'nkill':'killed',
    'nwound':'wounded',

}
```

The month and day data containing 0 were changed to 1 and replaced in a new 'date' column. Subsequently, the date field was segmented into day, month and year.

```
data_cleaned[['month','day']] = data_cleaned[['month','day']].replace(0,1)
data_cleaned['date'] = pd.to_datetime(data_cleaned[['year', 'month', 'day']])
```

Next, the entry value for year, month and day were encoded into numerical format. This ensured data consistency. Consistent data types **prevent errors in data manipulation and analysis**, particularly for date-related calculations(Yin et al., 2001).

```
# Data Type Conversion
# Convert year, month, and day to integers (if not already)
data_cleaned['year'] = data_cleaned['year'].astype(int)
data_cleaned['month'] = data_cleaned['month'].astype(int)
data_cleaned['day'] = data_cleaned['day'].astype(int)
```

After this, duplicate entries were removed.

```
# Handling Duplicates
data_cleaned = data_cleaned.drop_duplicates()
```

Managing outliers is essential for **maintaining the robustness and accuracy of data-driven** insights(Yin et al., 2001), especially when dealing with skewed data like casualties.

```
# Assuming we have a 'casualties' column
data_cleaned['killed'] = data_cleaned['killed'].clip(upper=data_cleaned['killed'].quantile(0.99))
```

Outliers can disproportionately affect statistical analyses and model performance. Clipping reduces **their influence, leading to more stable and reliable results**. Data values at the '**killed**' column were capped at 99th percentile to handle the outlier problem in given data-frame.

Before getting started with visual analysis of the data set, the data frame was checked to see if it contained any missing values. Identification of missing values in critical column can significantly corrupt the results.

```
data_cleaned[['country_name','year','country_code','city','killed','wounded']].isnull().sum()
```

	0
country_name	0
year	0
country_code	0
city	435
killed	10313
wounded	16311

The result indicated that some columns, ‘city’, ‘killed’ and ‘wounded’, contained null entries. These null entries were then handled in the next set of steps.

```
# Impute missing values in 'killed' based on the mean within each 'attack_type_name' group
data_cleaned['killed'] = data_cleaned.groupby('attack_type_name')['killed'].transform(lambda x: x.fillna(x.mean()))
data_cleaned['wounded'] = data_cleaned.groupby('attack_type_name')['wounded'].transform(lambda x: x.fillna(x.mean()))
```

Missing values in the killed and wounded columns **were filled with the mean value** within each attack type group, and **missing city values were replaced with 'Unknown'**. Imputation was **performed to preserve data** by replacing missing values with estimates based on other similar observations.

```
data_cleaned['city'].fillna('Unknown', inplace=True)
```

‘Group-based imputation’ was used to maintain the distribution within each group. This approach **reduced data loss and allowed in-depth analyses**, retaining the dataset's representativeness. The one final check verified that all null data in our dataset were handled properly.

	0
country_name	0
year	0
country_code	0
city	0
killed	0
wounded	0

Data Visualization

V1: Casualty Severity by Country

A '**choropleth map**' was utilized to visualize the total casualties resulting from terrorist activities worldwide. This covered the counts of both wounded as well as killed. The map employed a **color gradient ranging from light to dark red**, where darker shades represented higher numbers of casualties. The use of a color gradient effectively **communicated the severity of casualties**.

This visualization was specifically chosen to provide **a geospatial representation of the impact of terrorism** in terms of casualties. The purpose of this choropleth map was to allow viewers to quickly identify which countries experienced the most significant impact by visualizing the data geographically. This type of visualization is **considered particularly useful for policymakers, researchers, and analysts aiming to understand the geographic distribution** of terrorism's impact, and to prioritize resources and interventions accordingly.

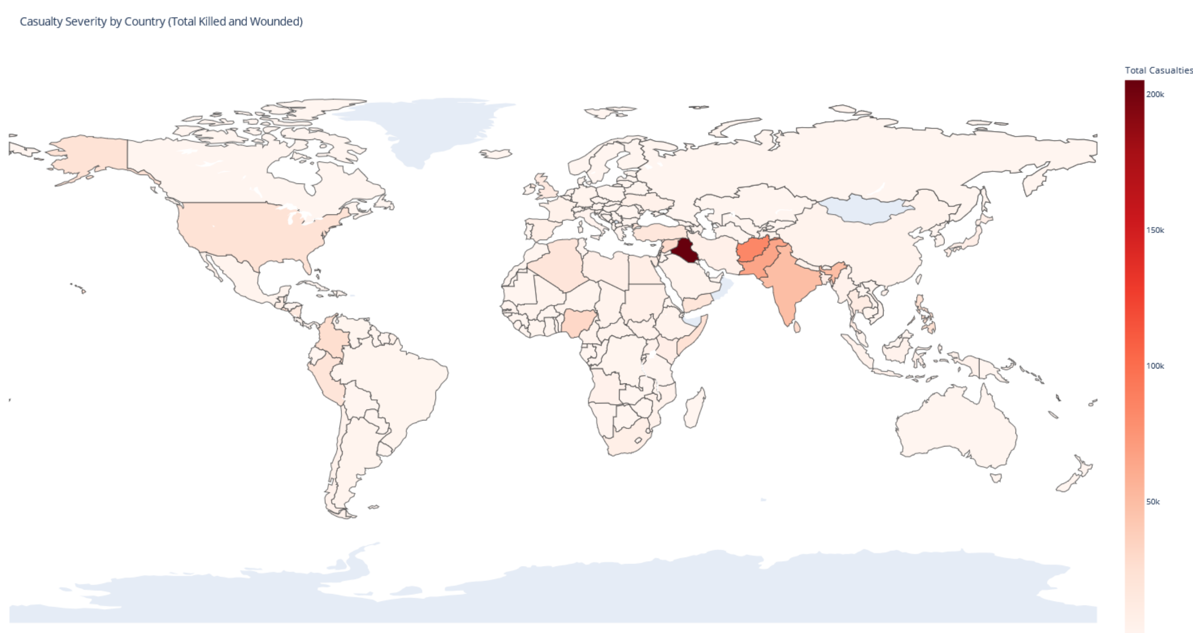


Fig 2 Visualization of Casualty Severity by Country

In the above plot, the darkest regions, indicating the highest number of casualties**, were primarily located in the Middle East and South Asia (with countries like Iraq, Afghanistan, and Pakistan)**. Countries such as **India and Nigeria were represented by medium red shades**. This implies that these nations also exhibited significant numbers of casualties, but were less severe. In contrast, **regions like North America, Western Europe,**

and parts of East Asia were shaded in very light red or white, suggesting fewer casualties. Some regions, such as Central Africa and Eastern Europe, showed very light or no color, potentially **indicating a lack of terrorist activity or insufficient data coverage**.

The code snippet used to generate this plot is shown below:

```
import plotly.express as px
# Calculate total casualties
data['total_casualties'] = data['killed'] + data['wounded']
casualties_by_country = data.groupby('country_name')['total_casualties'].sum().reset_index()

# Create the choropleth map
fig = px.choropleth(casualties_by_country,
                    locations='country_name',
                    locationmode='country names',
                    color='total_casualties',
                    hover_name='country_name',
                    color_continuous_scale='Reds',
                    title='Casualty Severity by Country (Total Killed and Wounded)',
                    labels={'total_casualties': 'Total Casualties'})
fig.update_layout(geo=dict(showframe=False, showcoastlines=False))
fig.show()
```

The use of countries as the geographical unit of analysis **provided a general overview but lacked granularity**. The map **did not account for population density**. The absence of a temporal component limited the ability to observe trends over time.

The choropleth map effectively highlighted the global distribution of terrorism-induced casualties. However, it **could be enhanced with additional charts showing temporal changes, regional details, and per capita adjustments.** This could provide a **more nuanced understanding** of global terrorism patterns and their impact.

V2: Attack Types Vs Regions

A **hive plot** with a circular layout was employed to visualize the connections between attack types and regions in the global terrorism dataset. Nodes represented categories like attack types (e.g., Bombing/Explosion) and regions (e.g., Middle East & North Africa), while edges connected these nodes, indicating where specific types of attacks occurred.

```
# Step 2: Build the Graph
G = nx.Graph()

# Add nodes for attack types and regions
attack_types = df_subset['attack_type_name'].unique()
regions = df_subset['region_name'].unique()

G.add_nodes_from(attack_types, bipartite=0)
G.add_nodes_from(regions, bipartite=1)
```

The purpose of this visualization was to **simplify the complex relationships** in the dataset, showing how different terrorist tactics are distributed across regions. The circular layout **allowed for a clear representation of these connections.** This makes it easier to identify prevalent attack types and their corresponding regions.

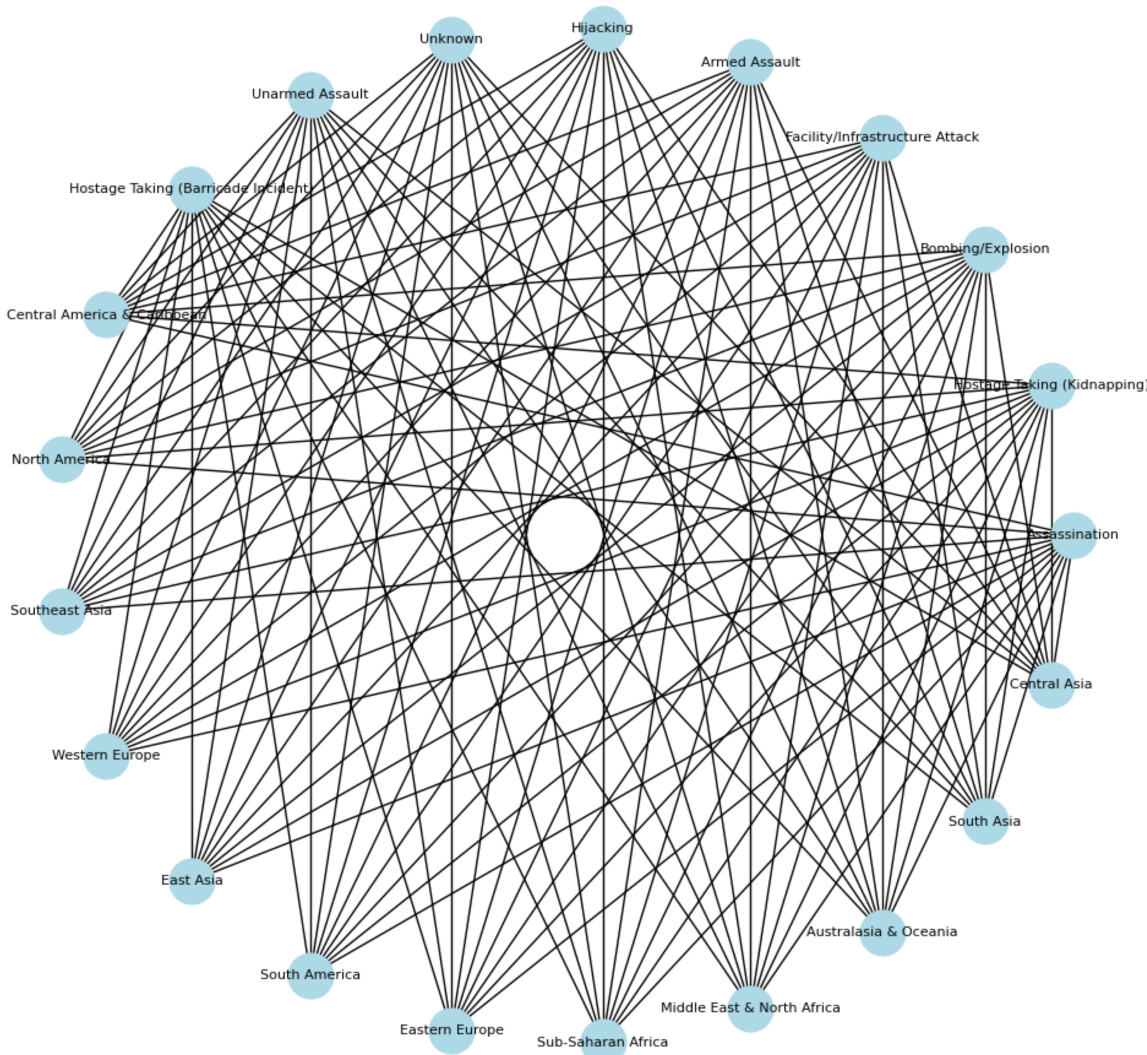


Fig 3 Visualization of Attack Types vs Regions

The plot revealed several insights: a hub-and-spoke pattern indicated that **many attack types were widely distributed across multiple regions**, underscoring the global nature of terrorism. More connected nodes, such as **Bombing/Explosion and Armed Assault, suggested these were common attack types**, especially in regions like the Middle East & North Africa and South Asia. Less connected nodes (Hijacking) were more region-specific.

However, the visualization **lacked quantitative data on the frequency of connections**, and the circular layout led to overlapping nodes, which could obscure detailed relationships.

The hive plot effectively displayed the global distribution of terrorist attack types and their regional connections. **While it provided valuable insights, enhancements like edge weights, color coding, and interactivity could improve the visualization's utility** for analysts and decision-makers. The code snippet for the above visualization is as follows:

```
plt.figure(figsize=(10, 10))
pos = nx.circular_layout(G)  # Circular layout for a structured appearance
nx.draw(G, pos, with_labels=True, node_size=700, node_color='lightblue', font_size=8, font_color='black')
plt.title('Hive Plot: Attack Types and Regions in Global Terrorism Dataset')
plt.show()
```

V3: Most Common Targets by Region

The **heatmap** was utilized to visualize the frequency of various target types in terrorist attacks across global regions. The primary aim of the heatmap was to offer an intuitive and comprehensive view of which target types were most frequently targeted in various regions.

By presenting the data in a matrix format, the **heatmap allowed for quick identification of patterns, outliers, and trends**, highlighting areas of both high and low terrorist activity. Created using the **Seaborn library in Python**, the heatmap enabled a straightforward visual analysis of the data. Heatmap was created with the aid of the following code snippet:

```
# Pivot the data to create a matrix suitable for a heatmap
target_by_region_pivot = target_by_region.pivot(index='targtype1_txt', columns='region_name', values='count').fillna(0)

# Step 3: Visualization
plt.figure(figsize=(14, 10))
sns.heatmap(target_by_region_pivot, cmap='YlGnBu', annot=True, fmt='.0f')
plt.title('Most Common Targets by Region')
plt.xlabel('Region')
plt.ylabel('Target Type')
plt.show()
```

Key observations from the chart revealed that **Private Citizens & Property was the most targeted** category globally, with notable frequencies in **South Asia** (15,257 incidents) and the **Middle East & North Africa** (10,491 incidents).

The **Military was also a frequent target, particularly in the Middle East & North Africa** (9,269 incidents) and South Asia (5,696 incidents),**revealing high-conflict zones where terrorist groups often engage military forces**. Additionally, regions such as **South Asia and the Middle East & North Africa** exhibited high concentrations of attacks across multiple target types, **marking these areas as hotspots for diverse terrorist activities**.

Conversely, **Western Europe and North America showed lower numbers of attacks**. This could possibly indicate fewer terrorist activities or more effective protective measures. Less common target categories, such **as Abortion-Related and Telecommunications, were rarely attacked,** as evidenced by lighter shades and lower frequencies.

An interesting outlier was the relatively **high number of attacks on Government (General) in Western Europe (2,157 incidents**) compared to other targets in the same region, which might indicate **regional conflicts or political unrest**.

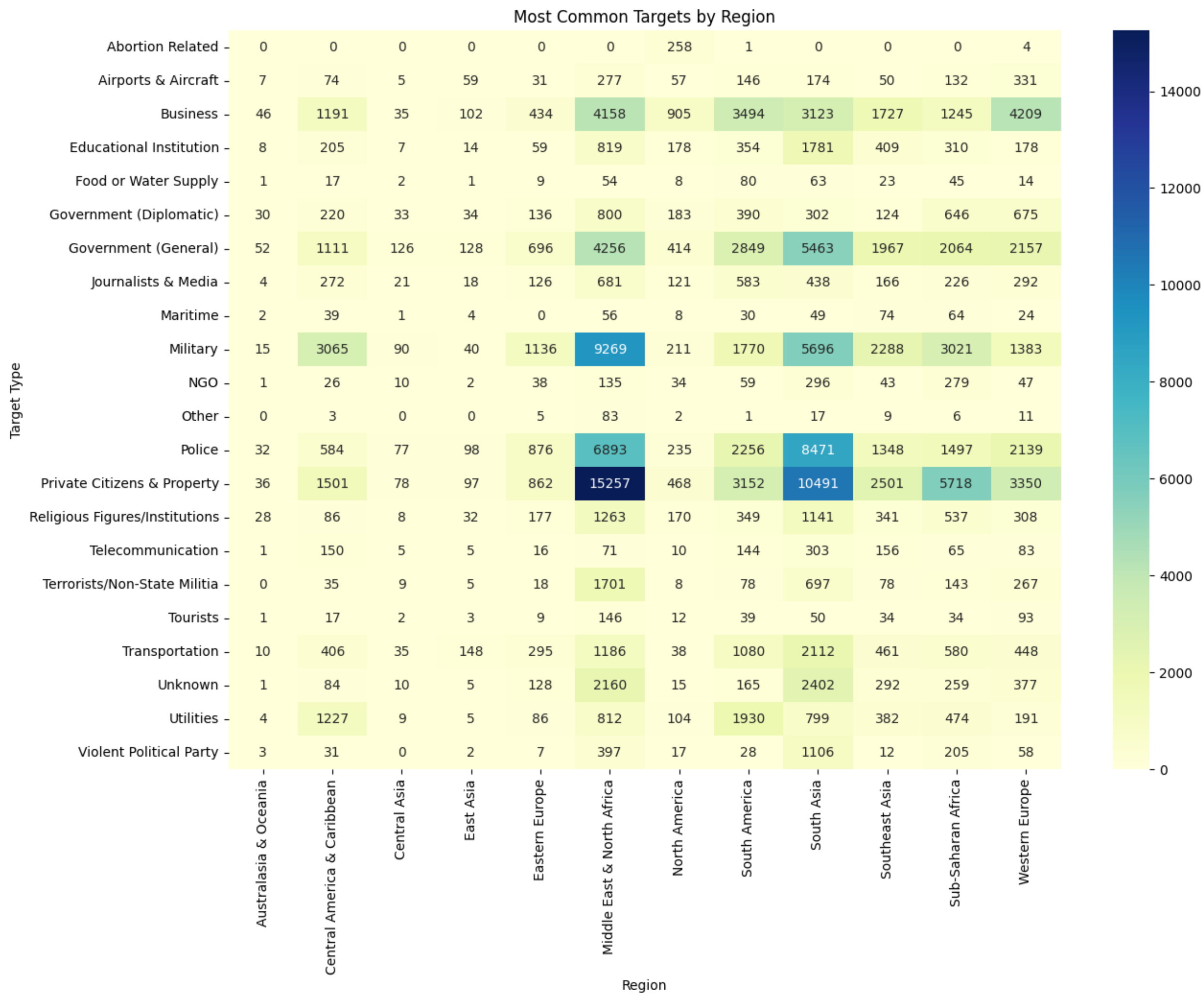


Fig 4 Visualization of Most Common Targets by Region

The heatmap's complexity could potentially overwhelm some viewers. **Simplifying the categories or focusing on specific areas of interest might enhance clarity.** Additionally, the absence of a temporal dimension in the visualization meant that changes over time were not accounted for. Incorporating a time component, such as an animated sequence of heatmaps over different periods, **could provide valuable insights into the evolution of targeting strategies.**

V4: Trends of Global Terrorist Attacks over time

A **line chart** was used to depict trends in global terrorist incidents over time, spanning from 1970 to 2017. Three distinct lines were presented, each showing **the number of terrorist incidents**, **the total number of people killed**, and **the total number of people wounded**. Each line was color-coded for clarity.

```
# Aggregate data by year
annual_data = data.groupby('year').agg(
    incidents=('eventid', 'count'),
    total_killed=('killed', 'sum'),
    total_wounded=('wounded', 'sum')
).reset_index()
```

This visualization aimed to illustrate **the temporal evolution of terrorism's impact globally**, allowing viewers to understand changes in the frequency and severity of attacks over several decades.

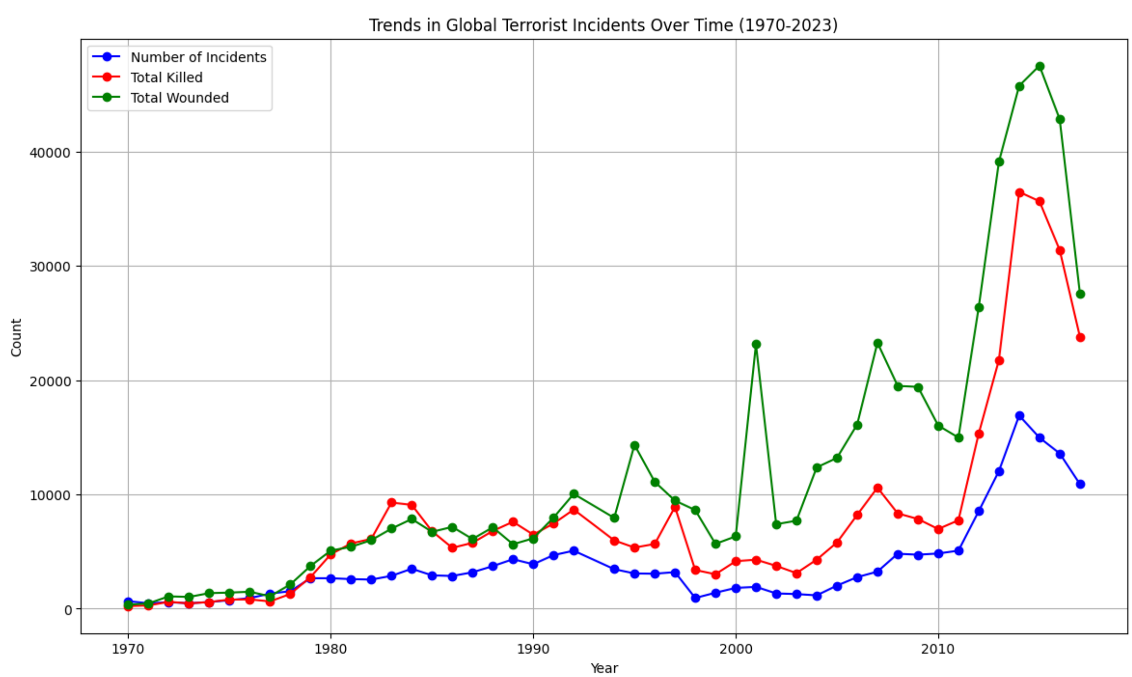


Fig 5 Visualization of Trends of Global Terrorist Attacks over time

This rather simpler visualization highlighted multiple patterns. First, a steady increase in all three metrics was observed from the early 1970s to the early 2010s, with **a particularly sharp rise starting around 2005**. The **number of incidents and casualties peaked around 2014**, suggesting a **period of heightened global terrorist activity**. Following this peak, there was a noticeable decline in both the total number of people killed and wounded, although the number of incidents showed a more gradual decrease. This trend indicated **that while**

terrorist activity remained relatively high, the lethality and impact of these attacks might have reduced slightly post-2014.

The chart also revealed that the total n**umber of wounded consistently exceeded the number of people killed**. This suggested that many terrorist attacks, while causing significant injuries**, did not always result in a proportionately high number of fatalities**. The peaks in the green and red lines corresponded with periods of significant terrorist incidents, indicating that **major attacks during these times resulted in both high casualties and fatalities**. Additionally, from around 1995 to 2005, a noticeable fluctuation in the **number of people wounded was observed, indicating periods of intensified violence or mass casualty events**.

Following code snippet was able to generate the above line chart

```
# Plotting the trend of incidents over time
plt.figure(figsize=(14, 8))
plt.plot(annual_data['year'], annual_data['incidents'], marker='o', linestyle='-', color='b', label='Number of Incidents')
plt.plot(annual_data['year'], annual_data['total_killed'], marker='o', linestyle='-', color='r', label='Total Killed')
plt.plot(annual_data['year'], annual_data['total_wounded'], marker='o', linestyle='-', color='g', label='Total Wounded')
# Adding titles and labels
plt.title('Trends in Global Terrorist Incidents Over Time (1970-2017)')
plt.xlabel('Year')
plt.ylabel('Count')
plt.legend()
# Display grid for better readability
plt.grid(True)
# Show plot
plt.show()
```

The visualization effectively communicated the trends in global terrorism, showing both the **frequency and human cost of terrorist attacks over time**. However, it could have been enhanced by adding **annotations to highlight major global events or terrorist attacks that corresponded with the peaks in the chart.** This would provide context to the observed trends.

V5: Trends of Attacks on Various Regions

The global trend of terrorist attack was further drilled down to analyze for each region present in the data set. This was achieved by the following python script:

```
# Adding traces for each region
for region in year_attacks_region['region_name'].unique():
    region_data = year_attacks_region[year_attacks_region['region_name'] == region]
    fig.add_trace(go.Scatter(
        x=region_data['year'],
        y=region_data['count'],
        mode='lines',
        name=region
    ))
```

The **Middle East & North Africa** and **South Asia** regions experienced the most significant increases in terrorist attacks, particularly after the early 2000s. Both regions showed sharp upward trends, peaking around 2014, which reflected a period of intense terrorist activity. This **pattern suggested that geopolitical instability, ongoing conflicts, and the rise of extremist groups in these areas (might have)** contributed significantly to the high number of attacks.

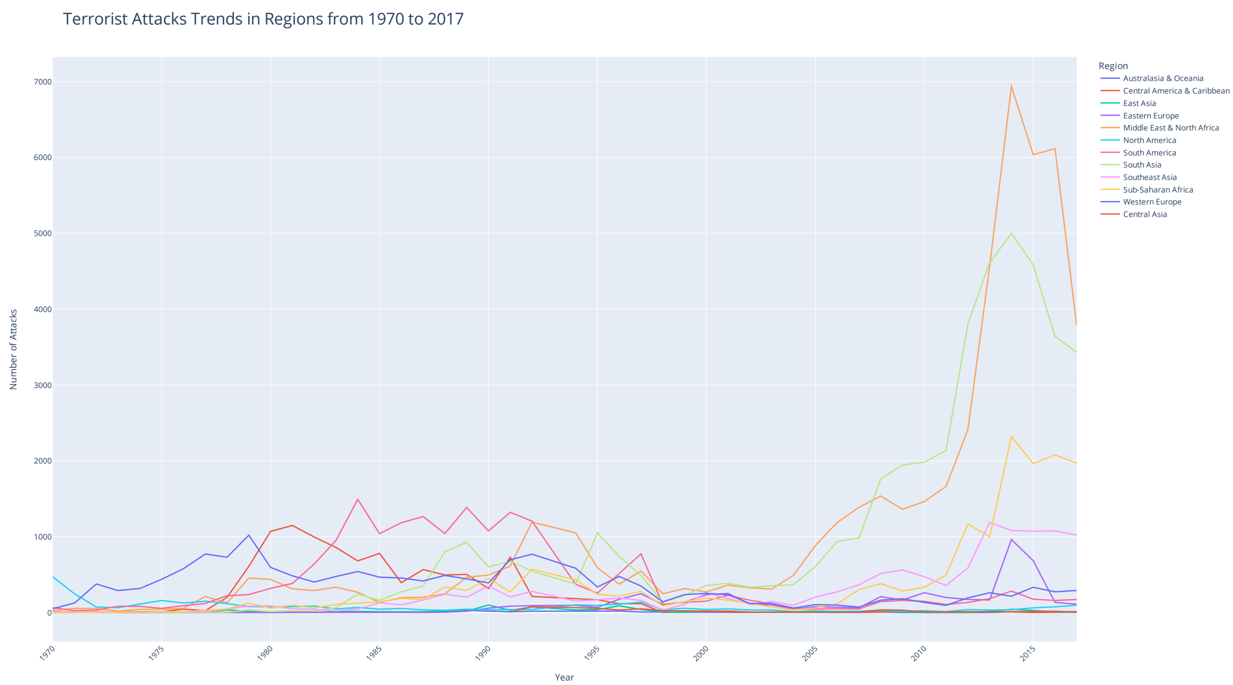


Fig 6 Visualization of Trends of Attacks in Various Regions

In contrast, other regions, such as **Western Europe** and **North America**, exhibited more modest fluctuations in the number of terrorist incidents, indicating a relatively stable or controlled environment with fewer large-scale terrorist activities.

The data also showed that certain regions, like **Sub-Saharan Africa** and **Southeast Asia**, experienced noticeable spikes in terrorist incidents, especially towards the late 2000s and early 2010s. Meanwhile, regions like **Australasia & Oceania** and **East Asia** maintained consistently low levels of terrorist activity throughout the entire period, suggesting a lower impact of terrorism or more effective counter-terrorism measures.

```python
# Updating the layout of the plot
fig.update_layout(
    title=dict(text='Terrorist Attacks Trends in Regions from 1970 to 2017', font=dict(size=25)),
    xaxis_title='Year',
    yaxis_title='Number of Attacks',
    xaxis=dict(tickangle=-45),
    legend=dict(title='Region')
)
```

This visualization too, has limitations like the previous one. Overall, the line chart **provided a clear and informative overview of regional trends in global terrorism**, illustrating the diverse patterns of terrorist activity worldwide.

V6: Factors Affecting the Success of Attacks

A **parallel coordinates plot** displayed the relationships between several factors affecting the success of terrorist attacks. The plot included four key dimensions: **success** (whether an attack was successful or not), **attack type**, **region**, and **weapon type**. This visualization aimed to explore how these different variables interacted and influenced the outcome of terrorist activities across various contexts globally.

A clear observation was that **Bombing/Explosion** and **Armed Assault** were the most common attack types (successful-or-not) across multiple regions. This suggested that **these attack types were frequently employed regardless of regional or tactical differences.** The lines representing these attack types were heavily concentrated in regions such as **South Asia**, **Middle East & North Africa**, and **Sub-Saharan Africa**, indicating that these areas **experienced a higher frequency of these types of attacks**.

Additionally, the use of **Explosives** emerged as a dominant weapon choice, closely associated with successful attacks, particularly in the regions. This pattern implied a preference for explosives due to their potential for causing significant damage and increasing the likelihood of a successful outcome.

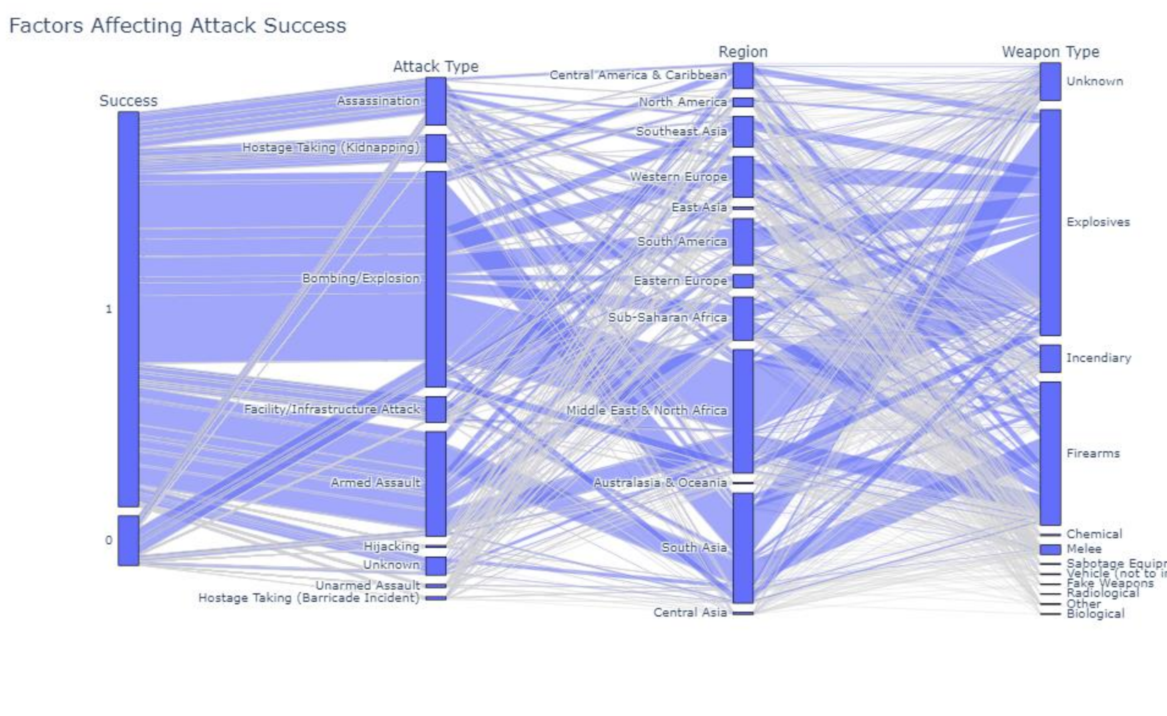


Fig 7 Visualization of Factors Affecting the Success of Attacks

Further analysis of the plot showed that certain regions, like **North America** and **Western Europe**, exhibited **a more diverse range of attack types and weapon usage** but with fewer connections overall. In contrast, regions like **South Asia** and the **Middle East & North Africa** showed a **more consistent and concentrated pattern, suggesting ongoing and frequent terrorist activities** using similar tactics and weapons. This **uniformity could point to the presence of established terrorist groups** employing well-defined strategies in these regions.

This visualization was generated with the aid of the following python script.

```python
# Prepare data for parallel coordinates plot
df_filtered = data_cleaned[['success', 'attack_type_name', 'region_name', 'weaptype1_txt']].dropna()

# Convert 'success' to a string to treat it as a categorical variable
df_filtered['success'] = df_filtered['success'].astype(str)

# Create a parallel coordinates plot using Plotly
fig = px.parallel_categories(df_filtered, dimensions=['success', 'attack_type_name', 'region_name', 'weaptype1_txt'],
                             labels={'success': 'Success', 'attack_type_name': 'Attack Type',
                                     'region_name': 'Region', 'weaptype1_txt': 'Weapon Type'},
                             title='Factors Affecting Attack Success')
```

The parallel coordinates plot **provided a valuable perspective on the factors affecting the success of terrorist attacks globally**. It highlighted the importance of understanding **regional contexts and tactical choices** in formulating effective counter-terrorism strategies. However, it could be **enhanced by incorporating additional dimensions, such as target type or the number of casualties,** to provide a more comprehensive understanding of the dynamics at play.

Adding interactivity, such as the ability to filter by specific regions or attack types, would allow exploration of specific patterns more deeply and gain insights tailored to analytical needs.

V7: Trend of Success Rate for Assassination

In this analysis, a specific type of attack is measured for its success rate. A **line chart** was used to depict the success rate of assassination attempts over the years, highlighting how the effectiveness of these attacks changed from 1970 to the 2010s.

Initially, **the success rate of assassinations was relatively high, often exceeding 85%,** indicating that a significant majority of these attempts were successful during the early years. However, the data revealed notable fluctuations, with **the success rate dropping sharply at certain points, such as the mid-1970s and early 2000s, reaching lows below 70%.** These drops suggested periods when assassination attempts were less effective, **potentially due to improved security measures and tactics**.

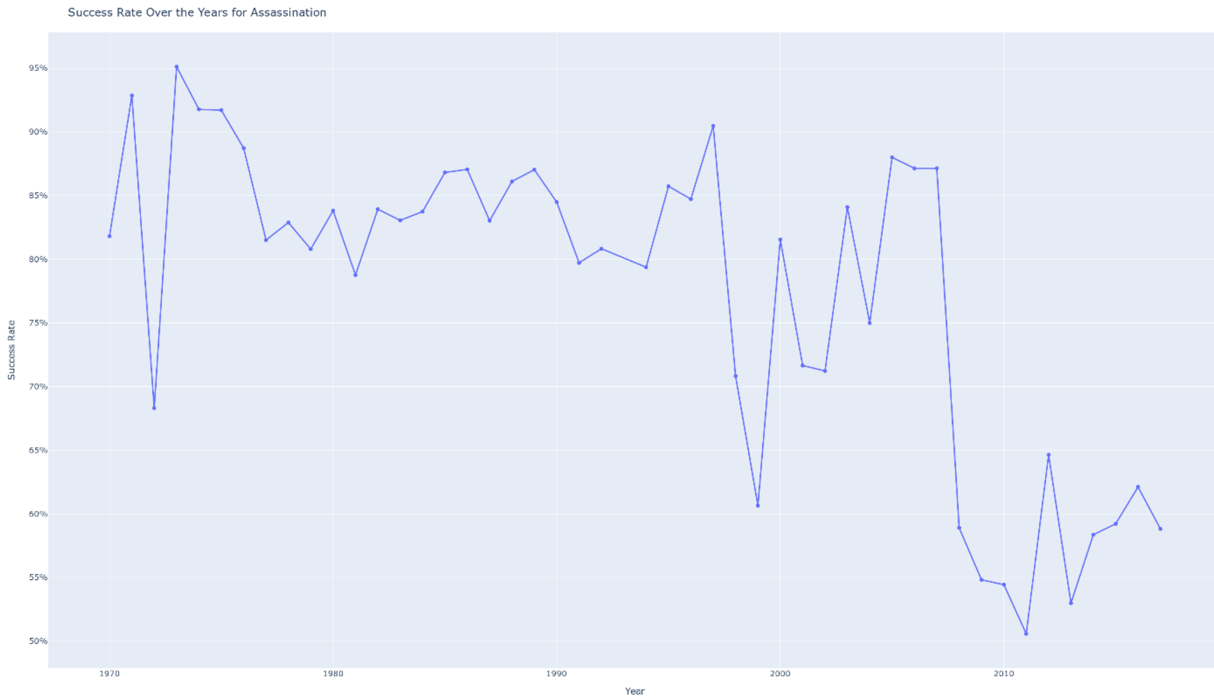


Fig 8 Visualization of Trend of Success Rate for Assassination

In the 1990s, a slight increase in the success rate was observed, stabilizing around 80-85% before another significant decline in the late 1990s and early 2000s. This decline marked one of the lowest points in the chart**, indicating a challenging period for assassinations**. The subsequent years showed a mix of recovery and further drops, reflecting an unstable pattern in the effectiveness of assassination attempts.

The analysis suggested that while assassinations remained a tactic used by terrorist groups, **their effectiveness had become less consistent**, **possibly due to better preparation and response by potential targets and security agencies**. However, the chart could be enhanced by including **comparative data for other attack types would also provide a broader** perspective on how assassination success rates compared with other forms of terrorist activities.

Overall, the line chart served as a valuable tool for analyzing the trends in the success rates of assassinations over several decades. The code snippet:

```
# Filter data for the least success rate attack type
df_least_success = data_cleaned[data_cleaned['attack_type_name'] == least_success_attack_type]
# Prepare data for plotting
df_least_success_plot = df_least_success.groupby(['year']).agg({'success': 'mean'}).reset_index()
# Create a line plot using Plotly to visualize the success rate over the years
fig = px.line(df_least_success_plot, x='year', y='success',
              title=f'Success Rate Over the Years for {least_success_attack_type}',
              labels={'success': 'Success Rate', 'year': 'Year'},
              markers=True)
```

V8: Trend for Suicide Attacks over time

The **line chart** depicted trends in suicide attacks and their impact over time. It included three lines each representing **total number of suicide attacks**, total **killed**, and **total wounded**. The chart's objective was to **visualize the temporal evolution of suicide attacks and their severity in terms of fatalities and injuries**. The required visualization was achieved using the following code snippet:

```
# Create a line plot to analyze trends in suicide attacks over time
fig = px.line(
    suicide_trends,
    x='year',
    y=['num_attacks', 'total_killed', 'total_wounded'],
    title='Trends in Suicide Attacks and Their Impact Over Time',
    labels={'value': 'Count', 'variable': 'Metric'},
    markers=True
)
```

It was observed that, initially, there were relatively few suicide attacks, with minimal casualties recorded through the 1980s and 1990s. A notable increase began around the early 2000s, particularly with a sharp spike in total wounded around 2001, reaching its peak. This spike indicated a **significant incident or a series of incidents that caused many injuries, reflecting a period of heightened violence**.

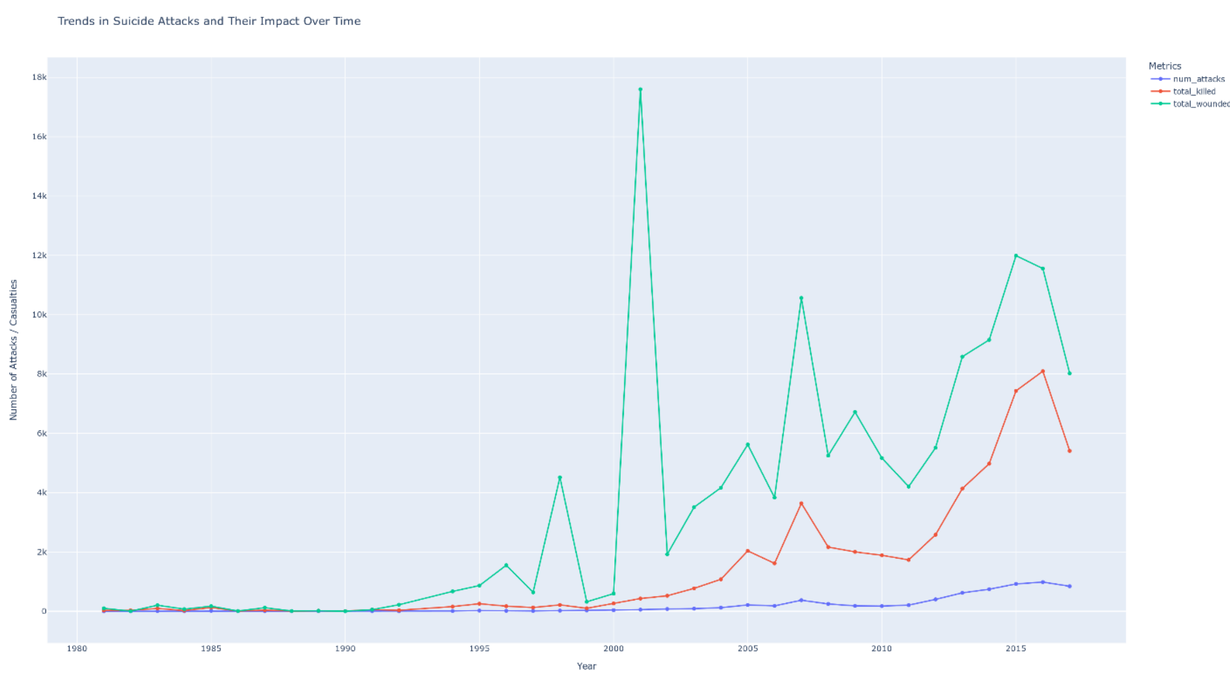


Fig 9 Visalization of Trend for Suicide Attacks over time

Following this peak, the numbers showed a fluctuating pattern, with subsequent spikes in both total killed and wounded, particularly noticeable around 2005 and the early 2010s. The chart also showed that the **total number of wounded consistently exceeded the number of killed** throughout the observed period, suggesting that **many suicide attacks resulted in injuries rather than fatalities**.

Furthermore, the overall upward trend in the number of attacks indicated that suicide tactics became increasingly common over the years, likely due **to their perceived effectiveness in causing mass casualties and drawing media attention**.

While the visualization effectively communicated the general trends in suicide attacks and their impact, it **could be enhanced by adding contextual information**, such as annotations highlighting significant events that correlated with the peaks in the data. Additionally**, normalizing the data by the global population could offer a more accurate picture** of the relative impact of suicide attacks over time, accounting for population growth.

V9: Regional Preference for Specific Attack Types

The **stacked bar chart** illustrated regional preferences for specific attack types, depicting how different regions favored certain types of terrorist attacks from 1970 to 2017. Each bar represented a region, and the segments within each bar were color-coded to show different attack types, such as Armed Assault, Bombing/Explosion etc. The plot for this chart was generated using the following code snippet

```
# Group data by region and attack type to get the count of each attack type per region
region_attack_type = df.groupby(['region_name', 'attack_type_name']).size().reset_index(name='count')

# Create a bar plot to explore the relationship between region and attack type
fig = px.bar(
    region_attack_type,
    x='region_name',
    y='count',
    color='attack_type_name',
    title='Regional Preferences for Specific Attack Types',
    labels={'count': 'Number of Attacks', 'region_name': 'Region', 'attack_type_name': 'Attack Type'},
    barmode='stack'
)
```

The **Middle East & North Africa** and **South Asia** regions exhibited the highest total number of attacks, with a **predominant use of Bombing/Explosion and Armed Assault**. This trend suggested a preference for these tactics in regions experiencing prolonged conflict and political instability. In **Sub-Saharan Africa** and **Southeast Asia**, a similar pattern was observed, although with slightly lower total attack numbers. This indicated a significant presence of terrorist activities but potentially with different strategic or operational constraints.

Regions like **Western Europe** and **North America** showed **a more diverse distribution of attack types,** with significant segments for **Hijacking** and **Hostage Taking**, particularly in Western Europe. In contrast, regions like **Australasia & Oceania** and **Central Asia** had relatively low numbers of attacks overall, suggesting that these areas were less affected by terrorism **or had more effective counter-terrorism measures in place**.

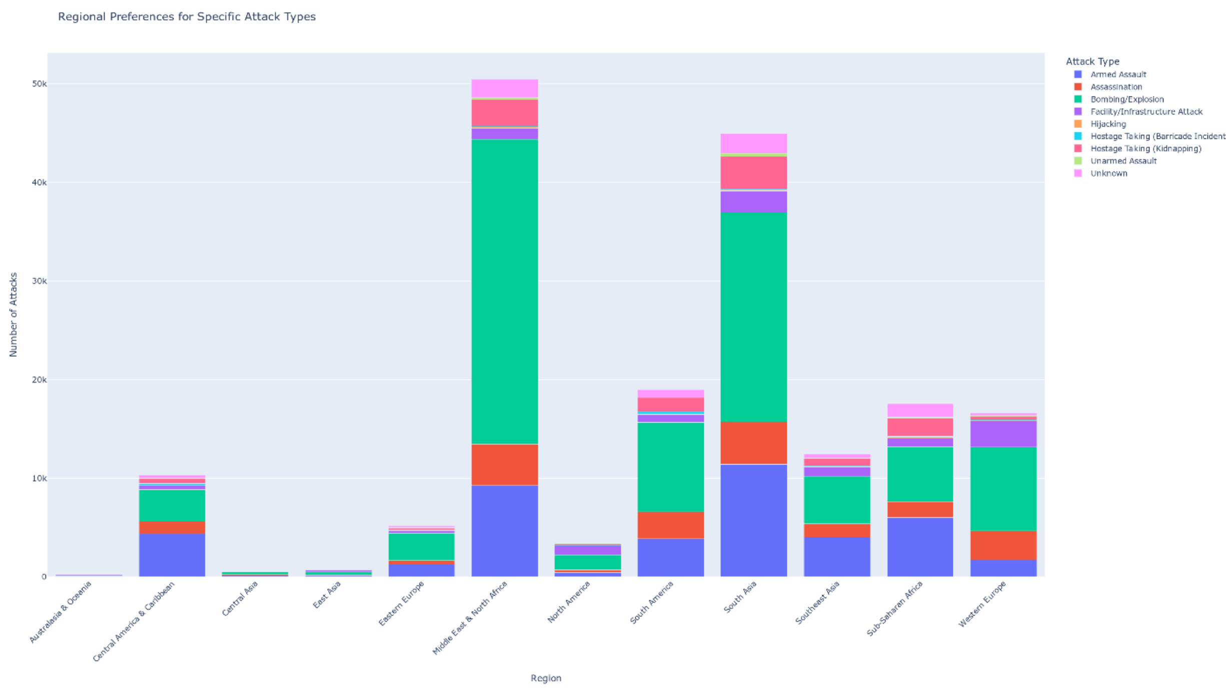


Fig 10 Visualization for Regional Preference for Specific Attack Types

In the above visualization, **normalizing the data by population size or geographical area could provide a clearer picture** of the relative impact of these attacks. It could offer insights into how concentrated or widespread terrorist activities were within each region. Overall, the stacked bar chart provided a comprehensive overview of regional preferences for specific attack types, illustrating the diversity and distribution of terrorist tactics across the globe

V10: Attack Incidents per County

The **horizontal bar chart** was used to display the total number of terrorist incidents per country from 1970 to 2017. Each bar represented a country, with the length indicating the total number of terrorist incidents recorded during this period.

Several key observations emerged from the visualization. **Iraq topped the list** with the highest number of terrorist incidents**, surpassing 25,000**, followed closely by **Pakistan** and **Afghanistan**, both **exceeding 10,000 incidents**. This concentration of incidents in these countries suggested **prolonged periods of conflict and instability**, likely driven by war. **India**, **Colombia**, and the **Philippines** also **featured prominently,** indicating that these countries experienced significant terrorist activities over the decades.

The chart also highlighted several countries in the Western Hemisphere, such as **El Salvador** and **Colombia**, and in Asia, like **Sri Lanka** and **Thailand**, which experienced a high frequency of terrorist incidents, reflecting regional insurgencies or prolonged internal conflicts. Conversely, countries like the **United States** and **United Kingdom** appeared on the list, but with comparatively fewer incidents.

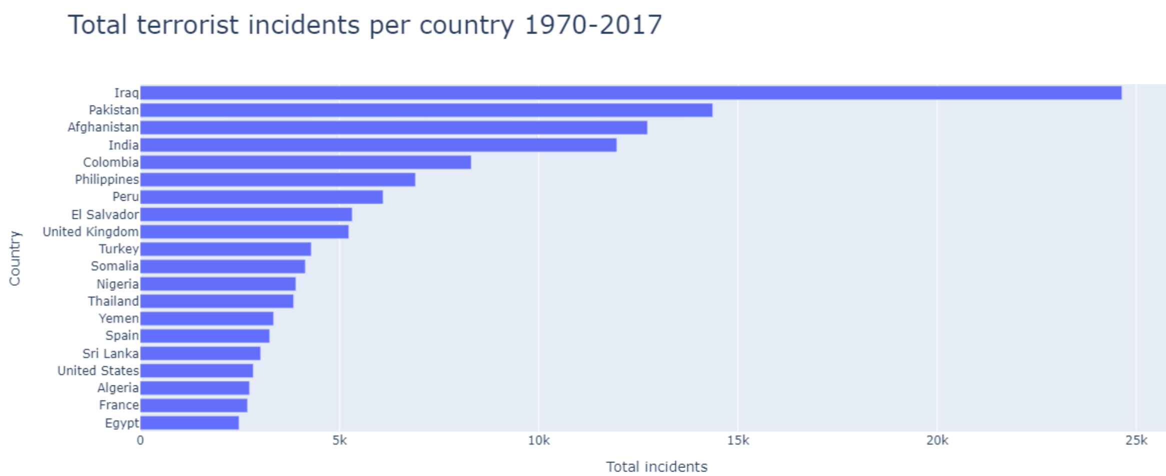


Fig 11 Visualization for Attack Incidents per Country

The visualization in the form of horizontal bar charts was achieved using the following code:

```
head_countries = countries.head(20)[::-1] #reversing the order

fig = go.Figure(go.Bar(
            y=head_countries.country_name,
            x=head_countries.incidents,
            orientation='h'))

fig.update_layout(title=dict(text='Total terrorist incidents per country 1970-2017', font=dict(size=25)),
                xaxis_title="Total incidents",
                yaxis_title="Country",)
```

The above visualization could be enhanced by **incorporating additional contextual information**. Moreover, **normalizing the data by population size or land area could offer a clearer understanding of the relative impact** of terrorism in each country, rather than just the total number of incidents.

V11: Comparison of Attacks Across Top 5 Regions

The **radar chart** provided a visual comparison of the prevalence of different attack types across the top five regions most affected by terrorism. The chart's circular format allowed **for a multi-dimensional comparison,** with each axis representing a specific attack type. Each region was represented by **a unique color-coded line and shaded area,** allowing for easy comparison of the concentration of attack types.

Bombing/Explosion was the most prevalent attack type in most regions, especially in the **Middle East & North Africa** and **South Asia**, as indicated by the longer extensions on the corresponding axes. **Armed Assault** was another common attack type across most regions, reflecting its versatility and effectiveness in various conflict scenarios.

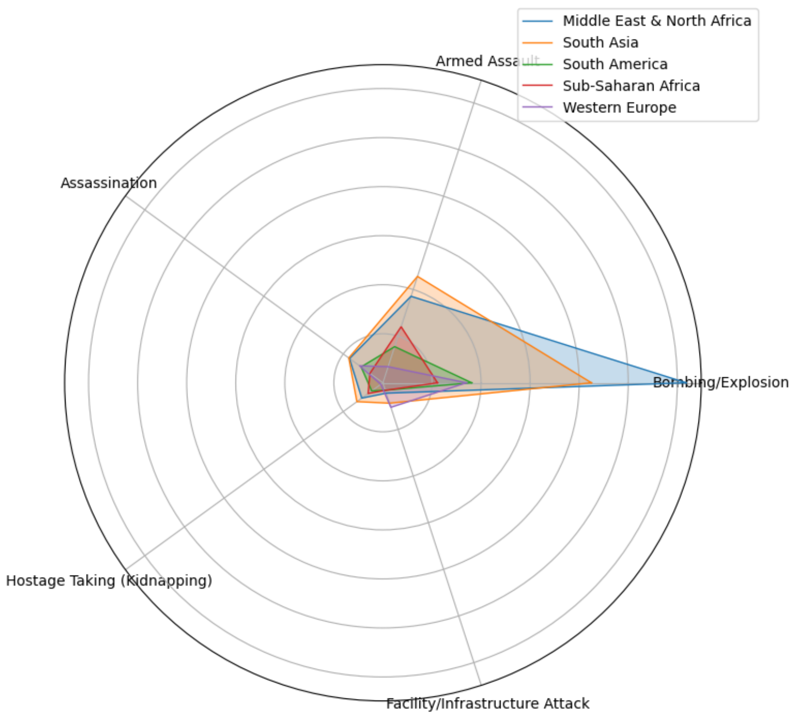


Fig 12 Visualization for Comparison of Attacks Across Top 5 Regions

Interestingly, **Western Europe** showed a more balanced distribution across different attack types, with no single type dominating the region. **Sub-Saharan Africa** and **South America** displayed unique profiles, with distinct peaks in attack types that differed from those in the Middle East and Asia, suggesting region-specific tactics or localized conflicts.

The radar chart effectively captured the comparative differences in terrorist tactics across regions, emphasizing the diversity of attack methods. However, the chart could benefit from **additional data points**, such as the number of casualties or the duration of the conflicts, to **provide a more comprehensive understanding**.

Following code snippet generated the above chart for the analysis:

```python
angles = np.linspace(0, 2 * np.pi, N, endpoint=False).tolist()

# The radar chart is a circle, so we need to "close the plot"
data = np.array(list(data_dict.values()))
data = np.concatenate((data, data[:,[0]]), axis=1)
angles += angles[:1]

fig, ax = plt.subplots(figsize=(8, 8), subplot_kw=dict(polar=True))

# Draw one axe per variable + add labels
ax.set_xticks(angles[:-1])
ax.set_xticklabels(categories)

# Draw ylabels
ax.set_yticklabels([])

# Plot data
for i, (region, values) in enumerate(data_dict.items()):
    ax.plot(angles, values + [values[0]], linewidth=1, linestyle='solid', label=region)
    ax.fill(angles, values + [values[0]], alpha=0.25)
```

V12: Targets of Terrorist Attacks

A **treemap** was used to provide a visual representation of the various targets of terrorist attacks, with each rectangle representing a different target type. The size of each rectangle corresponded to the number of incidents targeting that specific category from the dataset. This visualization aimed to **highlight the distribution and prominence of different targets** in terrorist strategies.

As seen in the chart below, **Private Citizens & Property** and **Military** were the most prominent targets, as **indicated by the largest rectangles**. This suggested that terrorist groups frequently targeted civilians and military entities, **likely due to their visibility and the impact such attacks have on public sentiment and national security**. The significant size of the **Government (General)** and **Police** categories also highlighted that **state and law enforcement institutions were common targets**, reflecting efforts by terrorist groups to undermine government authority and create a sense of insecurity.

Other notable categories included **Business**, **Transportation**, and **Religious Figures/Institutions**, which also had relatively large rectangles. Smaller rectangles, such as **Journalists & Media**, **Utilities**, and **Educational Institutions**, suggested these were less frequent targets. This could be potentially due to either their perceived lesser impact or the specific strategic objectives of the organizations involved

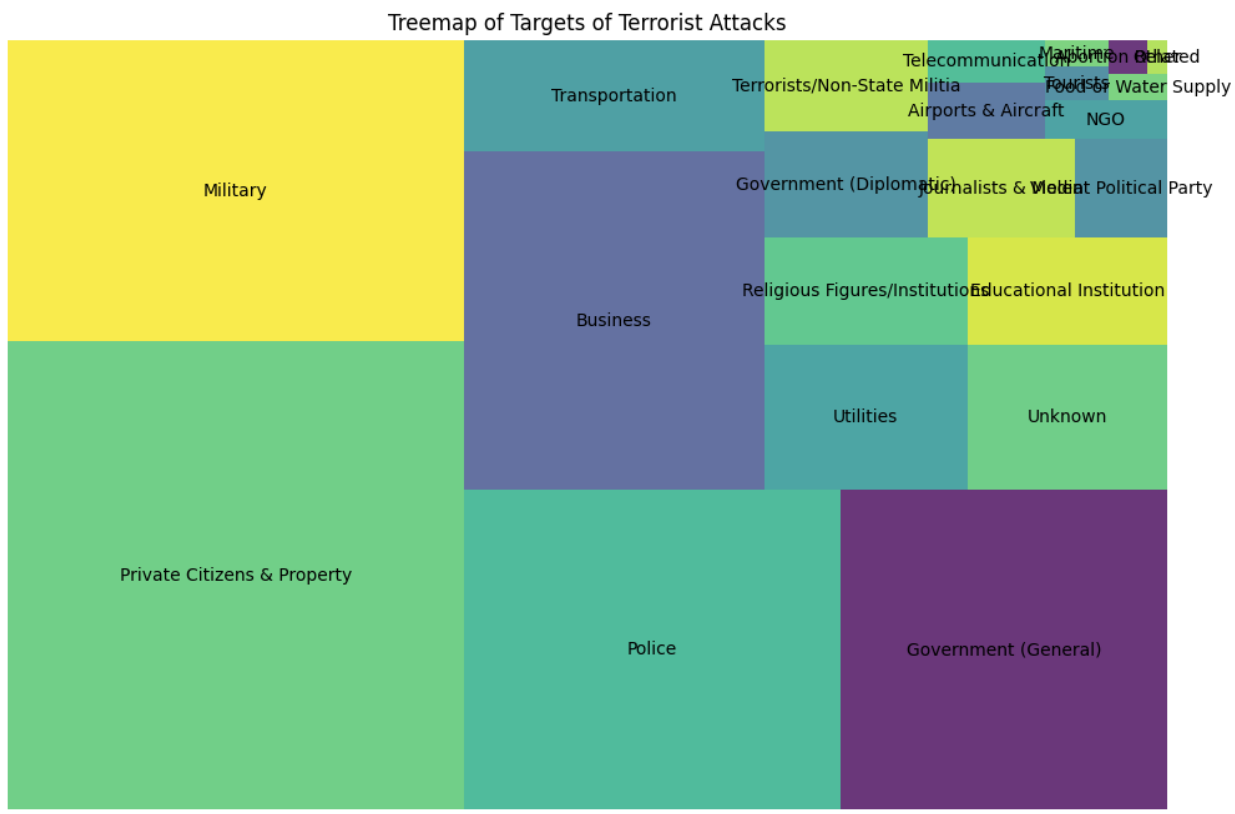


Fig 13 Visualization for Targets of Terrorist Attacks

The above chart was generated with the aid of the following code segment:

```
# Calculate the total number of incidents per target type
target_counts = data['targtype1_txt'].value_counts().reset_index()
target_counts.columns = ['Target Type', 'Number of Incidents']

# Plotting the Treemap
plt.figure(figsize=(12, 8))
squarify.plot(sizes=target_counts['Number of Incidents'], label=target_counts['Target Type'], alpha=0.8)
plt.axis('off')
plt.title('Treemap of Targets of Terrorist Attacks')
plt.show()
```

The chart could benefit from further refinement, such as **incorporating color gradients to indicate the relative severity or impact of attacks on different targets**. Additionally, including temporal data could provide insights into how target preferences have shifted over time, revealing trends in terrorist tactics.

V13: Alliance among Top 20 Terrorist Groups

The **network graph** illustrated the alliances among the top 20 terrorist groups, highlighting their connections based on shared targets or operational overlaps. **Each node represented a terrorist group**, while the **edges (lines) between them indicated associations or commonalities in their activities.** The size of the nodes reflected the prominence of each group in terms of the number of incidents or influence, **with larger nodes indicating groups with a higher number of terrorist incidents**.

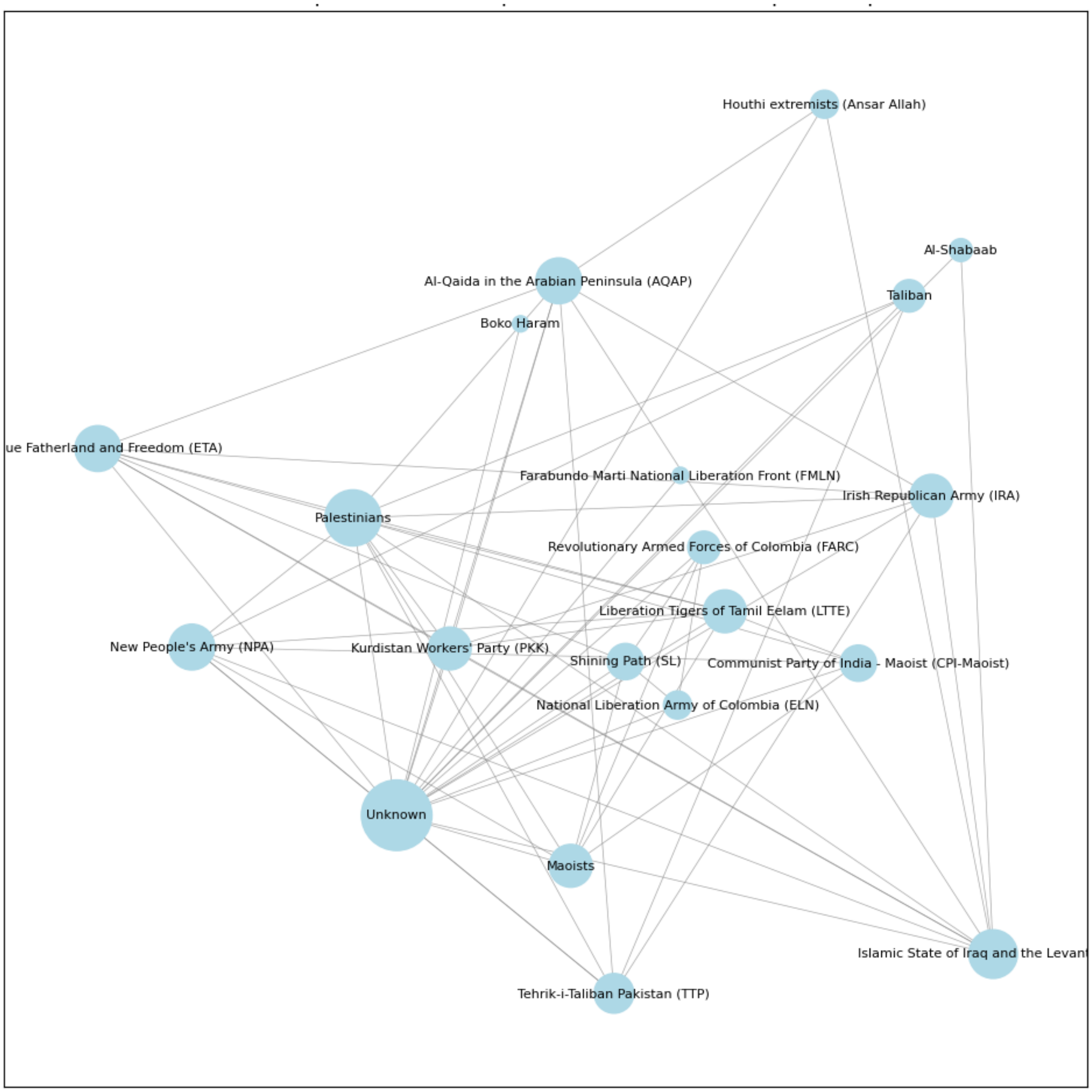


Fig 14 Visualization for Alliance among Top 20 Terrorist Group

In the chart above, '**Islamic State of Iraq and the Levant' (ISIL)** and '**Taliban' appeared as central nodes**, connected to multiple other groups, suggesting a **significant influence or operational overlap with other terrorist organizations**. '**Al-Qaida in the Arabian Peninsula (AQAP)'** and '**Boko Haram'** were also notably connected, further **revealing their**

active involvement in global terrorist networks and their complex relationships with other groups.

The presence of several peripheral nodes, such as '**Irish Republican Army' (IRA)** and '**Houthi extremists (Ansar Allah)'**, with fewer connections, suggested that **some groups operated more independently or had fewer overlapping activities with other top groups**. The '**Unknown'** node's centrality also highlighted the **challenges in categorizing and understanding all terrorist activities, indicating a signif**icant amount of incidents attributed to unidentified groups or alliances.

The visualization effectively captured the complexity of terrorist networks, showing how different groups were interconnected through alliances, shared tactics, or overlapping operational territories. Such a visualization was achieved by creating a concurrency matrix of top 20 organizations with the activities:

```
# Create a co-occurrence matrix for groups
co_occurrence = pd.crosstab(data_top_groups['gname'], data_top_groups['country_name'])
co_occurrence = co_occurrence.dot(co_occurrence.T)
np.fill_diagonal(co_occurrence.values, 0)
```

Further, the plot construct was created using Spring layout for better spacing and readability.

```
# Draw the graph using a spring layout for better spacing
plt.figure(figsize=(12, 12))
pos = nx.spring_layout(G, k=0.15, seed=42)  # k controls the distance between nodes, seed for reproducibility
nx.draw_networkx_nodes(G, pos, node_size=node_size, node_color='lightblue')
nx.draw_networkx_edges(G, pos, width=0.5, edge_color='grey', alpha=0.7)
nx.draw_networkx_labels(G, pos, font_size=8, font_color='black')
```

However, the graph could be **enhanced by including additional attributes**, such as the nature of the connections (whether cooperative or adversarial) to provide deeper insights into the dynamics between these groups. Furthermore, **using a weighted network approach**, where the thickness of the edges represents the strength or frequency of the connections, could offer more granular insights into the relationships.

V14: Top 20 Terrorist Groups and their Target Types

A **bipartite network graph** illustrated the relationships between the top 20 terrorist groups and their preferred target types, **presenting a clear visualization of how different terrorist organizations select their targets**. The visualization applied here is a one-step detailed version of the previous visualization.

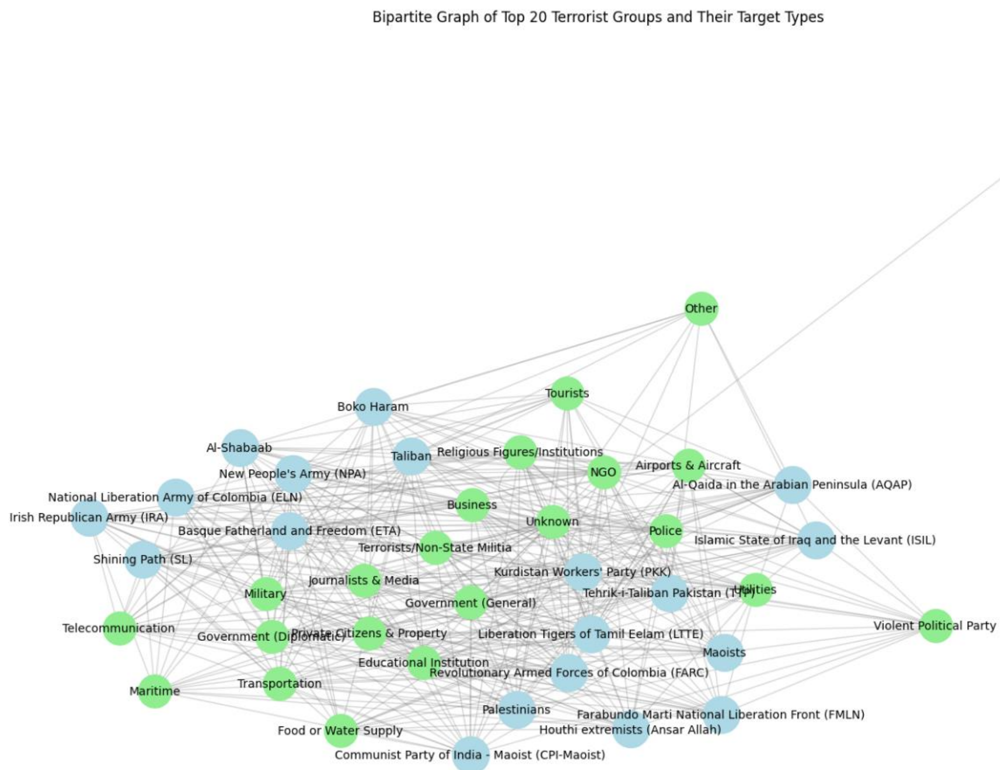


Fig 15 Visualization for Top 20 Terrorist Groups and Their Target Types

From the chart above, '**Islamic State of Iraq and the Levant' (ISIL)** and '**Taliban'** were prominently connected to a wide range of target types, including **Police**, **Business**, and **Private Citizens & Property**. This reflected their diverse operational strategies and broad targeting scope. The pattern indicated that these **groups were not only highly active but also adaptable, employing various tactics against multiple types of targets**. Similarly, groups like '**Al-Qaida in the Arabian Peninsula' (AQAP)** and '**Boko Haram'** showed multiple connections, emphasizing their strategic diversity and the breadth of their operational activities.

The graph also highlighted more specialized targeting patterns among some groups. For example, **Communist Party of India - Maoist (CPI-Maoist)** showed specific connections, reflecting targeted **actions aligned with their ideological goals**. In contrast, nodes like **Abortion Related** were connected to fewer groups, indicating niche targets that were less commonly attacked or targeted only by specific terrorist entities.

The **Unknown** node's multiple connections underscored the complexity and gaps in data, suggesting a **significant number of incidents where the responsible groups or specific motives were not clearly identified**.

The visualization effectively **captured the diverse targeting strategies employed by terrorist groups**, emphasizing the need for tailored counter-terrorism approaches that account for these differences.

This visualization was obtained through a two-step process, first simple bi-partite connections were created for each organization and its activity.

```
# Add nodes with the bipartite attribute
B.add_nodes_from(data_top_groups['gname'].unique(), bipartite=0)  # terrorist groups
B.add_nodes_from(data_top_groups['targtype1_txt'].unique(), bipartite=1)  # target types

# Add edges between terrorist groups and their target types
edges = list(zip(data_top_groups['gname'], data_top_groups['targtype1_txt']))
B.add_edges_from(edges)

# Separate the nodes by their bipartite class
group_nodes = {n for n, d in B.nodes(data=True) if d.get('bipartite') == 0}
target_nodes = set(B) - group_nodes

# Position nodes using a bipartite layout
pos = nx.spring_layout(B, k=0.1, iterations=100, seed=42)  # More iterations for better positioning
```

Then multiple nodes were connected in the form of a network to obtain the desired graph.

```
# Draw nodes for terrorist groups and target types with distinct colors
nx.draw_networkx_nodes(B, pos, nodelist=group_nodes, node_color='lightblue', node_size=1000, label='Terrorist Groups')
nx.draw_networkx_nodes(B, pos, nodelist=target_nodes, node_color='lightgreen', node_size=800, label='Target Types')

# Draw edges with reduced alpha for less clutter
nx.draw_networkx_edges(B, pos, alpha=0.3, edge_color='grey')

# Draw labels with reduced font size
nx.draw_networkx_labels(B, pos, font_size=10, font_color='black')
```

The graph could be further **enhanced by incorporating additional attributes**, such as the geographical regions of operations or the frequency of attacks, to provide a more detailed understanding of the relationships. I**ncluding weighted edges to reflect the strength** or frequency of each group-target connection would also add depth to the analysis.

V15: Geographics Trends of Attacks over Time : Interactive Plot

On the final attempt of this visual analysis, a geographical plot of terrorist attack was created to enable viewers to go to any year and see the overall pattern. Below are the screenshot results of the trend that occurred every 5 years.

Use of this chart provided easy and straight-forward analysis of the temporal trends over time. An animated scatter geo plot was then created using the filtered data. The scatter_geo function from Plotly Express.

```python
# Create an animated scatter geo plot
fig = px.scatter_geo(
    data_filtered,
    lat='latitude',
    lon='longitude',
    color='region_name',
    hover_name='country_name',
    size='killed',
    animation_frame='year',
    title='Animated Temporal Analysis of Global Terrorist Incidents (1970-2017)',
    projection='natural earth',
    labels={'killed': 'Fatalities'},
    template='plotly_dark'
)
```

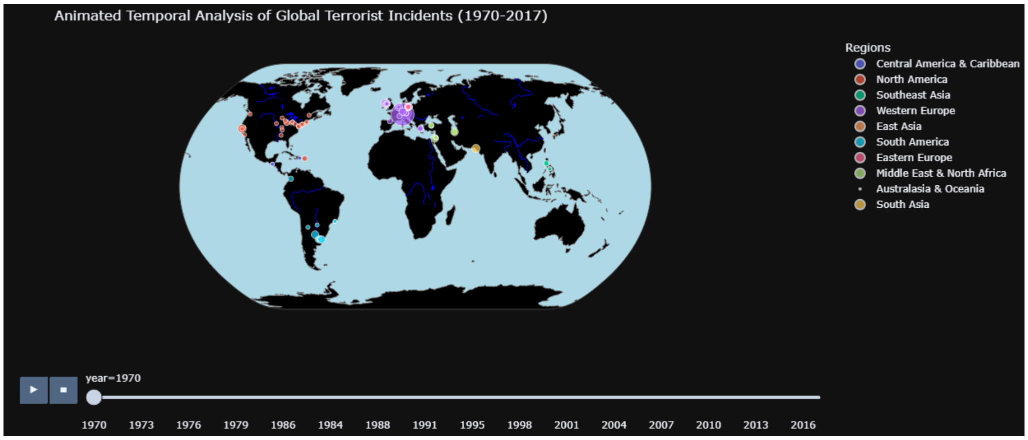
Subsequently, geographical settings of the plot were updated to enhance visual clarity. Coastlines, land, oceans, lakes, and rivers were made visible, and their respective colors were set for better differentiation:

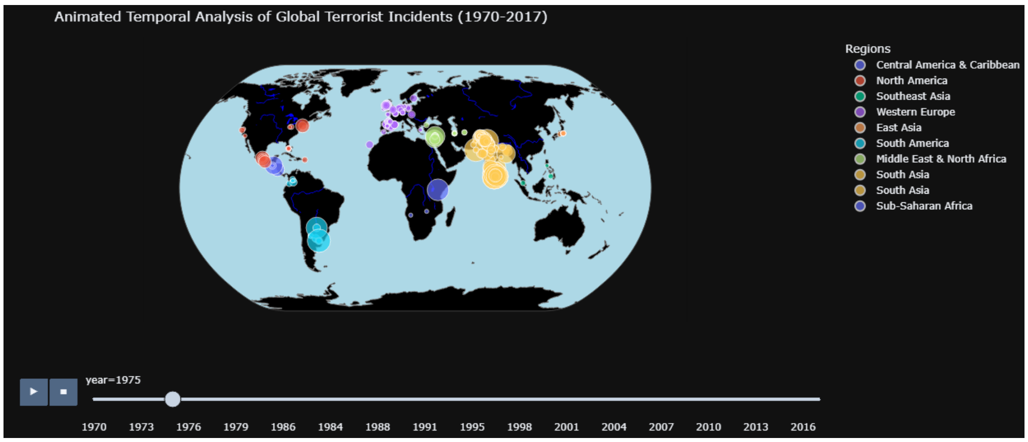
```python
# Update geographical settings
fig.update_geos(
    showcoastlines=True, coastlinecolor="Gray",
    showland=True, landcolor="Black",
    showocean=True, oceancolor="LightBlue",
    showlakes=True, lakecolor="Blue",
    showrivers=True, rivercolor="Blue"
)
```

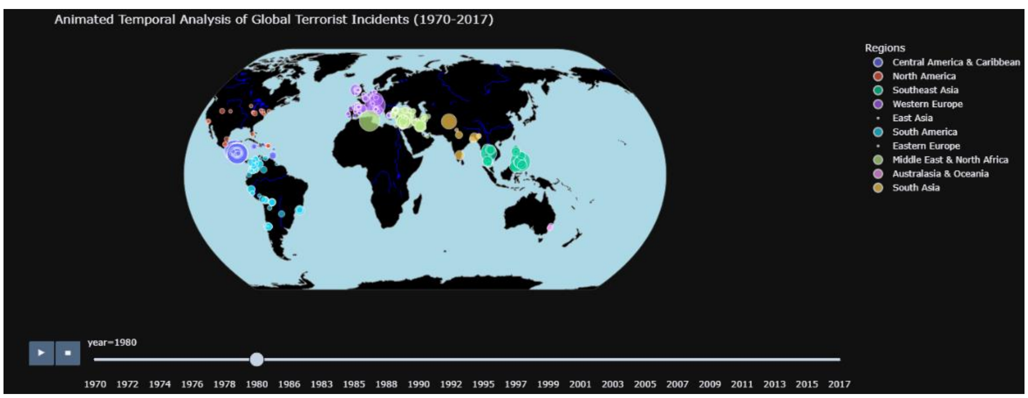
To refine the plot's appearance, the layout was adjusted. Margins were set to zero for all sides except the top, which was set to 40. A title for the legend and a color bar title for the regions were also added:

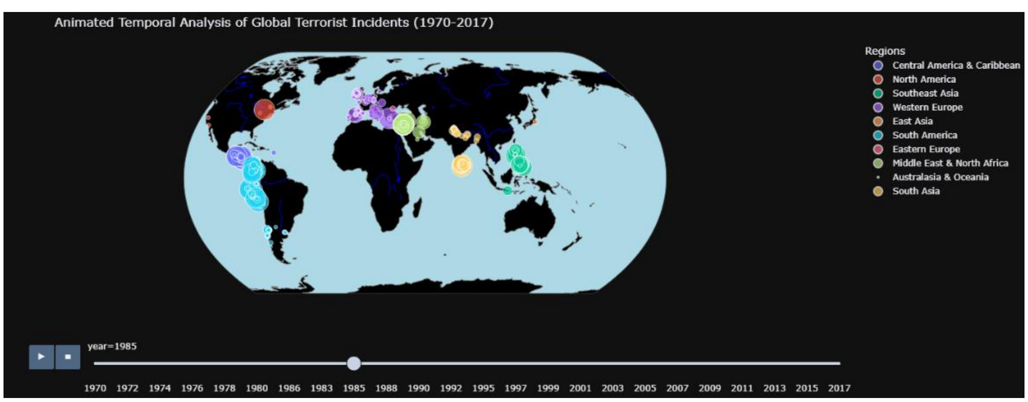
```python
# Update layout for a cleaner look
fig.update_layout(
    margin={"r":0,"t":40,"l":0,"b":0},
    legend_title="Regions",
    coloraxis_colorbar=dict(title="Region"),
)
```

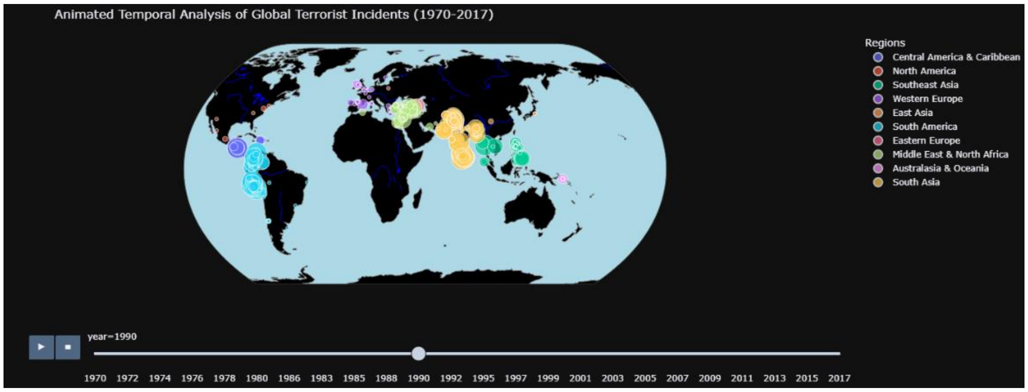
The realized visualization contained 50+ visuals, all interactive. For sample sake, visualization of data every 5 years, starting 1970 and final visualization of 2017 data are attached in the following section of the report.

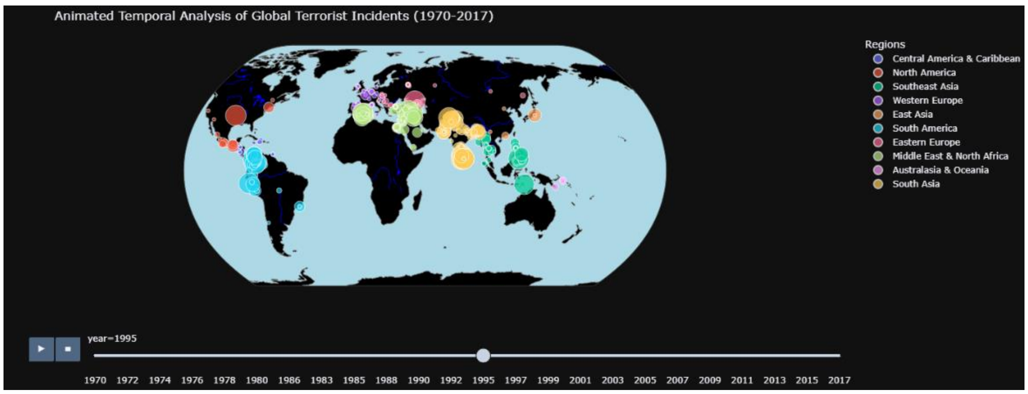

Animated Temporal Analysis of Global Terrorist Incidents (1970-2017)
Regions
Central America & Caribbean
North America
Southeast Asia
Western Europe
East Asia
South America
Eastern Europe
Middle East & North Africa
Australasia & Oceania
South Asia
year=1970
1970 1973 1976 1979 1986 1984 1988 1991 1995 1998 2001 2004 2007 2010 2013 2016

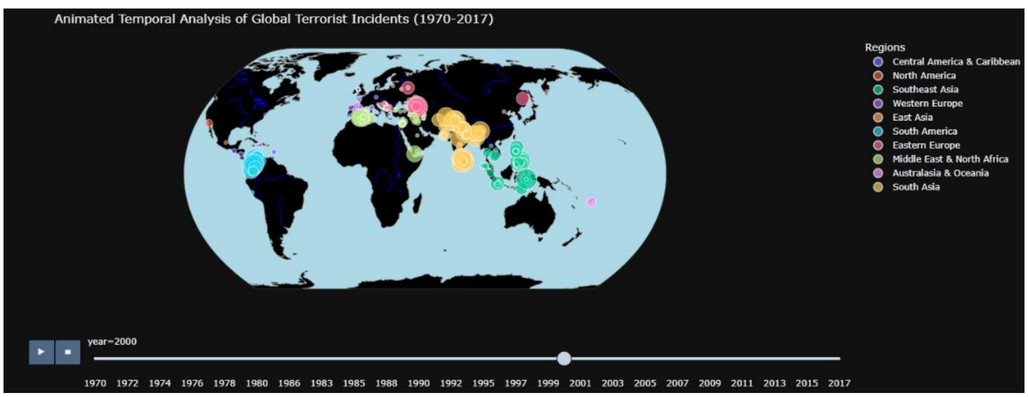

Animated Temporal Analysis of Global Terrorist Incidents (1970-2017)
Regions
Central America & Caribbean
North America
Southeast Asia
Western Europe
East Asia
South America
Middle East & North Africa
South Asia
South Asia
Sub-Saharan Africa
year=1975
1970 1973 1976 1979 1986 1984 1988 1991 1995 1998 2001 2004 2007 2010 2013 2016


Animated Temporal Analysis of Global Terrorist Incidents (1970-2017)
Regions
Central America & Caribbean
North America
Southeast Asia
Western Europe
East Asia
South America
Eastern Europe
Middle East & North Africa
Australasia & Oceania
South Asia
year=1980
1970 1972 1974 1976 1978 1980 1986 1983 1985 1988 1990 1992 1995 1997 1999 2001 2003 2005 2007 2009 2011 2013 2015 2017


Animated Temporal Analysis of Global Terrorist Incidents (1970-2017)
Regions
Central America & Caribbean
North America
Southeast Asia
Western Europe
East Asia
South America
Eastern Europe
Middle East & North Africa
Australasia & Oceania
South Asia
year=1985
1970 1972 1974 1976 1978 1980 1986 1983 1985 1988 1990 1992 1995 1997 1999 2001 2003 2005 2007 2009 2011 2013 2015 2017


Animated Temporal Analysis of Global Terrorist Incidents (1970-2017)
Regions
Central America & Caribbean
North America
Southeast Asia
Western Europe
East Asia
South America
Eastern Europe
Middle East & North Africa
Australasia & Oceania
South Asia
year=1990
1970 1972 1974 1976 1978 1980 1986 1983 1985 1988 1990 1992 1995 1997 1999 2001 2003 2005 2007 2009 2011 2013 2015 2017


Animated Temporal Analysis of Global Terrorist Incidents (1970-2017)
Regions
Central America & Caribbean
North America
Southeast Asia
Western Europe
East Asia
South America
Eastern Europe
Middle East & North Africa
Australasia & Oceania
South Asia
year=1995
1970 1972 1974 1976 1978 1980 1986 1983 1985 1988 1990 1992 1995 1997 1999 2001 2003 2005 2007 2009 2011 2013 2015 2017


Animated Temporal Analysis of Global Terrorist Incidents (1970-2017)
Regions
Central America & Caribbean
North America
Southeast Asia
Western Europe
East Asia
South America
Eastern Europe
Middle East & North Africa
Australasia & Oceania
South Asia
year=2000
1970 1972 1974 1976 1978 1980 1986 1983 1985 1988 1990 1992 1995 1997 1999 2001 2003 2005 2007 2009 2011 2013 2015 2017

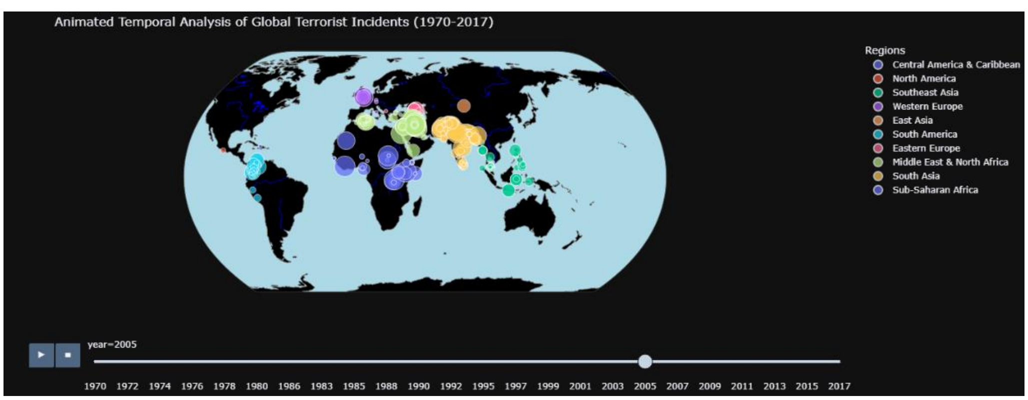

Animated Temporal Analysis of Global Terrorist Incidents (1970-2017)
Regions
Central America & Caribbean
North America
Southeast Asia
Western Europe
East Asia
South America
Eastern Europe
Middle East & North Africa
South Asia
Sub-Saharan Africa
year=2005
1970 1972 1974 1976 1978 1980 1986 1983 1985 1988 1990 1992 1995 1997 1999 2001 2003 2005 2007 2009 2011 2013 2015 2017

Fig 16 Visualization for Geographics Trends of Attacks over Time : Interactive Plot

Big Data Technologies

Big data technologies have changed the way of processing, analyzing, and storing massive datasets for organizations and researchers that were previously unmanageable(Warren & Marz, 2015). This capability allows organizations to uncover valuable insights, that facilitate real-time data processing, and advanced analytics, train a predictive model, and drive informed decision-making across various industries. Apache Hadoop and Apache Spark are widely used for large-scale data processing and distributed computing.

As discussed earlier, GTD is a comprehensive dataset that contains detailed information on terrorist events around the world, allowing it to provide valuable insights into global terrorism patterns, trends, and behavior. Here's how the big data technologies can be applied to the GTD for managing, analyzing, and illustrating their benefits.

- **Enhanced Data Processing and Analysis.**

With over 150,000 incidents recorded since 1970, it is difficult to analyze using traditional data processing tools. Apache Spark with its in-memory computing capabilities can play a vital role in processing this dataset efficiently when running complex queries(Damji et al., 2020), such as clustering terrorist groups based on their activities.

- **Scalability and Flexibility.**

As it holds a great history, it will continue growing with new incidents. Thus, a system like NoSQL databases like Cassandra, MongoDB, and HBase are required and well-suited for this purpose. It allows horizontal scaling and is flexible enough to handle the varied and unstructured nature of the data(Damji et al., 2020) within the GTD like free-text descriptions of incidents and varying formats of different regions and dates.

- **Data integration and Data lakes**

Big data technologies also excel in integrating diverse data sources and creating data lakes(Warren & Marz, 2015). Integrating the same with the GTD can create a more comprehensive analytical framework, as the data lakes built in Spark might have relevant datasets like political stability indexes, economic indicators, or social media data. This integration allows for multi-dimensional analysis, enabling researchers to explore the relationships between terrorism and other relevant datasets.

- **Advanced Analytics and Machine Learning**

Another sector in which Big Data excels is in advanced machine learning and analytics applications. Tools like MLlib (Spark's ML library) and Apache Mahout provides scalable ML Algorithms, enabling organizations to build predictive models that analyze patterns a forecast future terrorist activities.

- **Real-Time Data Processing**

In addition to the features of big data technologies, it also includes real-time data processing for immediate insights. Technologies like Apache Storm, Apache Flink, and Apache Kafka have real-time streaming capabilities that allow organizations to batch process and analyze data as it is generated. This is particularly useful in scenarios like fraud detection, and maintaining industrial equipment, where immediate actions is required to reduce the failures.

- **Data Privacy and Security**

When it comes to protecting sensitive information, Big Data Technologies also offers robust security and privacy features(Warren & Marz, 2015). Tools like Apache Ranger and Apache Knox offer comprehensive security frameworks including authentication, authorization, and data encryption.

Application of Big-Data Technologies

Among all the big data analysis tools, PySpark was used on our dataset for big data analysis. PySpark, the Python API of the well-known big data analysis tool Apache Spark, enabled real-time, large-size data processing in a distributed environment using Python.

The use of PySpark for managing and analyzing large datasets was considered a valuable experience(Damji et al., 2020). Challenges of traditional tools, where working on large datasets was tedious and time-consuming, were overcome. PySpark was recognized as a powerful tool for big data analysis due to its efficiency and scalability, which made it possible to handle enormous datasets quickly through distributed computing(Damji et al., 2020). Tasks such as data cleansing, transformation, and aggregation were significantly accelerated by Spark's ability to process and analyze large-scale data in-memory. It was seen as a great choice for communities working on big data analysis due to its distributed computing capabilities.

PySpark's intuitive API, which was familiar to users of pandas, facilitated an easier transition to big data tools and integrated seamlessly with the Python ecosystem, allowing for advanced analytics and machine learning. The efficiency of PySpark in distributed computing further enhanced data processing by parallelizing tasks(Databricks, 2017), which substantially reduced the time required for complex operations like aggregations and text processing.

While working with the tools, a few challenges were encountered:

- **Complexity in Cluster Management:** A good understanding of distributed computing principles and cluster management techniques was required to use such tools effectively. Without these, effective resource management (memory, CPU allocation) and optimization of data partitioning could not be performed, directly affecting performance. Detailed knowledge was necessary due to the complexity compared to single-node data processing tools.

- **Debugging and Troubleshooting:** Debugging errors in a distributed system could be more challenging if knowledge about system handling was lacking.

Some of the data analysis done on our dataset using PySpark were:

Firstly we install PySpark using:

```
pip install pyspark
```

Initial Data Inspection:

```python
# Show the first few rows
spark_df.show(5)

# Display schema
spark_df.printSchema()
```

```
+------------+-----+------+----+----------+--------+----------+-------+------------------+------+------------------+---------+---------------+-
|     eventid|iyear|imonth|iday|approxdate|extended|resolution|country|       country_txt|region|         region_txt|provstate|           city|
+------------+-----+------+----+----------+--------+----------+-------+------------------+------+------------------+---------+---------------+-
|197000000001| 1970|     7|   2|      NULL|       0|      NULL|     58|Dominican Republic|     2|Central America &...|     NULL|Santo Domingo|1
|197000000002| 1970|     0|   0|      NULL|       0|      NULL|    130|            Mexico|     1|     North America|  Federal|  Mexico city|1
|197001000001| 1970|     1|   0|      NULL|       0|      NULL|    160|       Philippines|     5|    Southeast Asia|   Tarlac|      Unknown|1
|197001000002| 1970|     1|   0|      NULL|       0|      NULL|     78|            Greece|     8|    Western Europe|  Attica|       Athens|
|197001000003| 1970|     1|   0|      NULL|       0|      NULL|    101|             Japan|     4|         East Asia|  Fukouka|      Fukouka|3
+------------+-----+------+----+----------+--------+----------+-------+------------------+------+------------------+---------+---------------+-
only showing top 5 rows

root
 |-- eventid: long (nullable = true)
 |-- iyear: integer (nullable = true)
 |-- imonth: integer (nullable = true)
 |-- iday: integer (nullable = true)
 |-- approxdate: string (nullable = true)
 |-- extended: integer (nullable = true)
 |-- resolution: string (nullable = true)
 |-- country: integer (nullable = true)
 |-- country_txt: string (nullable = true)
 |-- region: integer (nullable = true)
 |-- region_txt: string (nullable = true)
```

Showing the total number of rows

```python
# Count the number of rows
row_count = spark_df.count()
print("Number of rows in the dataset:", row_count)
```

```
Number of rows in the dataset: 181691
```

Grouping and Counting Events by Country:

```python
# Count the number of events by country
spark_df.groupBy("country_txt").count().orderBy("count", ascending=False).show()
```

```
+--------------+-----+
|   country_txt|count|
+--------------+-----+
|          Iraq|24636|
|      Pakistan|14368|
|   Afghanistan|12731|
|         India|11960|
|      Colombia| 8306|
|   Philippines| 6908|
|          Peru| 6096|
|   El Salvador| 5320|
|United Kingdom| 5235|
|        Turkey| 4292|
|       Somalia| 4142|
|       Nigeria| 3907|
|      Thailand| 3849|
|         Yemen| 3347|
|         Spain| 3249|
|     Sri Lanka| 3022|
| United States| 2836|
|       Algeria| 2743|
|        France| 2693|
|         Egypt| 2479|
+--------------+-----+
only showing top 20 rows
```

Checking for missing values in each column with null or NaN

```python
from pyspark.sql.functions import col, isnan, when, count

missing_data = spark_df.select([count(when(isnan(c) | col(c).isNull(), c)).alias(c) for c in spark_df.columns])
missing_data.show()
```

```
+-------+-----+------+----+----------+--------+----------+-------+-----------+------+----------+--------+----+--------+------
|eventid|iyear|imonth|iday|approxdate|extended|resolution|country|country_txt|region|region_txt|provstate|city|latitude|longitu
+-------+-----+------+----+----------+--------+----------+-------+-----------+------+----------+--------+----+--------+------
|      0|    0|     0|   0|    172452|       0|    179471|      0|          0|     0|         0|     423| 434|    4556|    45
+-------+-----+------+----+----------+--------+----------+-------+-----------+------+----------+--------+----+--------+------
```

Grouped by the year, counting the number of occurrences for each year and sorting them chronologically.

```
# Groups the data by year, counts the number of occurrences for each year
spark_df.groupBy("iyear").count().orderBy("iyear").show()
```

```
+-----+-----+
|iyear|count|
+-----+-----+
| 1970|  651|
| 1971|  471|
| 1972|  568|
| 1973|  473|
| 1974|  581|
| 1975|  740|
| 1976|  923|
| 1977| 1319|
| 1978| 1526|
| 1979| 2662|
| 1980| 2662|
| 1981| 2586|
| 1982| 2544|
| 1983| 2870|
| 1984| 3495|
| 1985| 2915|
| 1986| 2860|
| 1987| 3183|
| 1988| 3721|
| 1989| 4324|
+-----+-----+
only showing top 20 rows
```

Count By Region

```
spark_df.groupBy("region_txt").count().orderBy("count", ascending=False).show(10)
```

```
+--------------------+-----+
|          region_txt|count|
+--------------------+-----+
|Middle East & Nor...|50474|
|          South Asia|44974|
|       South America|18978|
|  Sub-Saharan Africa|17550|
|      Western Europe|16639|
|      Southeast Asia|12485|
|Central America &...|10344|
|      Eastern Europe| 5144|
|       North America| 3456|
|           East Asia|  802|
+--------------------+-----+
only showing top 10 rows
```

Correlation between number of kills and number of wounds:

```
correlation = spark_df.corr("nkill", "nwound")
print(f"Correlation between number of kills and number of wounds: {correlation}")
```

```
Correlation between number of kills and number of wounds: 0.4437453082033473
```

Total Count of attack Type

```
spark_df.groupBy("attacktype1_txt").count().orderBy("count", ascending=False).show()
```

```
+--------------------+-----+
|     attacktype1_txt|count|
+--------------------+-----+
|   Bombing/Explosion|88101|
|       Armed Assault|42566|
|       Assassination|19285|
|Hostage Taking (K...|11115|
|Facility/Infrastr...|10222|
|             Unknown| 7267|
|     Unarmed Assault| 1003|
|Hostage Taking (B...|  989|
|           Hijacking|  657|
|                   0|  167|
|                   1|  100|
|                   3|   56|
|                   7|   52|
|                   2|   40|
|                NULL|   35|
|                   6|   16|
|                   8|    5|
|                   9|    4|
|                   5|    2|
|  the Red Hand Def...|    1|
+--------------------+-----+
only showing top 20 rows
```

Conclusion

The Global Terrorism Database (GTD) stands as a crucial tool for scholars, policymakers, and analysts seeking to understand and mitigate the threat of terrorism. By offering an in-depth view of terrorism's global landscape, the GTD helps inform evidence-based policies and strategies that are crucial for countering terrorism. The database's breadth and depth allow for nuanced analysis, enabling stakeholders to develop targeted interventions and responses that address the underlying drivers of terrorism. Furthermore, its transparency and accessibility promote rigorous scholarly work, fostering a deeper understanding of the complex phenomena of terrorism and its far-reaching impacts on global peace and security.

GitHub Link

https://github.com/karki-dennis/Global-Terrorism-Datas-Visualization
https://github.com/Acharya-jyu/Global-Terrorism-Datas-Visualization
https://github.com/leeraaz/Global-Terrorism-Datas-Visualization

Division of Work

Work-Report Segment	Aashish Acharya	Dennish Karki	Liraj Maharjan
Global Terrorism : A Global Problem	✓		
Our Approach to the Problem	✓	✓	✓
Introduction to the (Global Terrorism) Dataset	✓		
Origin of the Dataset	✓		
Scope of the Dataset	✓		
Methodology and Data Collection		✓	
Data Types			✓
Significance of the Dataset	✓		
Dataset Composition	✓		
Dataset Exploration			✓
Dataset Cleaning and Preprocessing	✓	✓	✓
V1: Casualty Severity by Country		✓	
V2: Attack Types Vs Regions	✓		
V3: Most Common Targets by Region	✓		
V4: Trends of Global Terrorist Attacks over time		✓	
V5: Trends of Attacks on Various Regions			✓
V6: Factors Affecting the Success of Attacks			✓
V7: Trend of Success Rate for Assassination			✓
V8: Trend for Suicide Attacks over time	✓		
V9: Regional Preference for Specific Attack Types			✓
V10: Attack Incidents per County		✓	
V11: Comparison of Attacks Across Top 5 Regions			✓
V12: Targets of Terrorist Attacks		✓	
V13: Alliance among Top 20 Terrorist Groups	✓		
V14: Top 20 Terrorist Groups and their Target Types		✓	
V15: Geographic Trends of Attacks Over Time: Interactive Plot	✓		
Big Data Technologies		✓	✓
Application of Big Data Technologies	✓	✓	
Conclusion	✓		

References

Bardwell, H., & Iqbal, M. (2021). The Economic Impact of Terrorism from 2000 to 2018. In *Peace Economics, Peace Science and Public Policy* (Vol. 27, Issue 2). https://doi.org/10.1515/peps-2020-0031

Damji, J. S., Wenig, B., Das, T., & Lee, D. (2020). Learning Spark: Lightning-Fast Data Analytics. In *Learning Spark: Lightning-Fast Data Analytics*.

Databricks. (2017). A Gentle Introduction To Apache Spark. *Communication*.

Institute for Economics & Peace. (2024). *2024 GTI*. www.economicsandpeace.org

START, U. of M. (2024). *GTD*.

UN Security Council. (2020). *Action taken by Member States and United Nations entities to address the issue of linkages between terrorism and organized crime Report of the Secretary-General*.

Warren, J., & Marz, N. (2015). *Big Data: Principles and best practices of scalable realtime data systems*. Manning.

Yin, D. P., Sankary, H. N., Ma, L. L., Shen, J., Qin, Y., Blinder, L., Williams, J. W., & Chong, A. S. (2001). Python for Data Analysis Data Wrangling with Pandas, NumPy, and IPython. In *Transplantation* (Vol. 71, Issue 10).