Global Terrorism Analysis

Introduction

This analysis delves into **global terrorism trends**, exploring how terrorist activities have evolved over time and identifying regions with significant deviations from global patterns. By examining attack success rates, prevalent tactics, and regional variations, we aim to uncover key insights into the nature of terrorist incidents worldwide. This exploration utilizes interactive plots and geographic visualizations to enhance understanding and engagement.

About the Dataset

The dataset, sourced from the **Global Terrorism Database (GTD)**, provides comprehensive data on over **180,000 terrorist attacks** from 1970 to 2017. Managed by the National Consortium for the Study of Terrorism and Responses to Terrorism (START), this open-source repository offers detailed information on both domestic and international incidents, enabling a thorough examination of global terrorism trends.

Key Features:

- **Time Span:** 1970 - 2017
- **Number of Incidents:** 180,000+
- **Scope:** Domestic and International
- **Data Limitations:** The dataset may have biases or gaps due to underreporting or varying definitions of terrorism

Project steps

`First`

- Download Dataset & load :- [https://drive.google.com/drive/folders/1KxVUhXePXaQBzBz9kMHEMl3lCjt3C-4P?usp=drive_link]

`second`

- import nessary libraries

`third`

- Let's dive deep into the data to find out

In [1]:

```
# Import Libraries
import pandas as pd
import numpy as np
import os
import chardet
import missingno as msno
```

```python
import time
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.animation as animation
import cartopy.crs as ccrs
import cartopy.feature
import io
import base64
from IPython.display import HTML
# plt.style.use('ggplot')

import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```python
# extra_addition
pd.set_option('display.max_columns', 1000)  # Show 1000 columns
pd.set_option('display.max_rows', 1000)     # Show 1000 rows
```

In [3]:

```python
path = os.path.join(os.getcwd(),"globalterrorismdb_0718dist.csv")
path
```

Out[3]:

```
'F:\\from_C\\proj_jupyter\\globalterrorismdb_0718dist.csv'
```

After download data on local machine check what is encoding using to encode data for this using (chardet) to define type data to avoid down UnicodeDecodeError.

In [4]:

```python
# If you don't have chardet installed, uncomment the line below to install it:
# !pip install chardet
```

In [5]:

```python
with open(path,"rb") as obj:
    res= chardet.detect(obj.read(10000))
print(res)
```

```
{'encoding': 'ISO-8859-1', 'confidence': 0.73, 'language': ''}
```

- Load and Explore the dataset

In [6]:

```python
start_time = time.time()

df = pd.read_csv("globalterrorismdb_0718dist.csv",encoding='ISO-8859-1',low_memory=False)

pandas_duration = time.time() - start_time

pandas_duration
```

Out[6]:

```
18.8396315574646
```

In [7]:

```python
df.head()
```

Out[7]:

	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	country_txt	region	region_txt

	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	country_txt	region	region_txt
0	197000000001	1970	7	2	NaN	0	NaN	58	Dominican Republic	2	Central America & Caribbean
1	197000000002	1970	0	0	NaN	0	NaN	130	Mexico	1	North America
2	197001000001	1970	1	0	NaN	0	NaN	160	Philippines	5	Southeast Asia
3	197001000002	1970	1	0	NaN	0	NaN	78	Greece	8	Western Europe
4	197001000003	1970	1	0	NaN	0	NaN	101	Japan	4	East Asia

In [8]:

```
df.shape
```

Out[8]:

```
(181691, 135)
```

In [9]:

```
df.columns.to_list()
```

Out[9]:

```
['eventid',
 'iyear',
 'imonth',
 'iday',
 'approxdate',
 'extended',
 'resolution',
 'country',
 'country_txt',
 'region',
 'region_txt',
 'provstate',
 'city',
 'latitude',
 'longitude',
 'specificity',
 'vicinity',
 'location',
 'summary',
 'crit1',
 'crit2',
 'crit3',
 'doubtterr',
 'alternative',
 'alternative_txt',
 'multiple',
 'success',
 'suicide',
 'attacktype1',
 'attacktype1_txt',
 'attacktype2',
 'attacktype2_txt',
 'attacktype3',
 'attacktype3_txt',
 'targtype1',
 'targtype1_txt',
 'targsubtype1',
```

```
'targsubtype1_txt',
'corp1',
'target1',
'natlty1',
'natlty1_txt',
'targtype2',
'targtype2_txt',
'targsubtype2',
'targsubtype2_txt',
'corp2',
'target2',
'natlty2',
'natlty2_txt',
'targtype3',
'targtype3_txt',
'targsubtype3',
'targsubtype3_txt',
'corp3',
'target3',
'natlty3',
'natlty3_txt',
'gname',
'gsubname',
'gname2',
'gsubname2',
'gname3',
'gsubname3',
'motive',
'guncertain1',
'guncertain2',
'guncertain3',
'individual',
'nperps',
'nperpcap',
'claimed',
'claimmode',
'claimmode_txt',
'claim2',
'claimmode2',
'claimmode2_txt',
'claim3',
'claimmode3',
'claimmode3_txt',
'compclaim',
'weaptype1',
'weaptype1_txt',
'weapsubtype1',
'weapsubtype1_txt',
'weaptype2',
'weaptype2_txt',
'weapsubtype2',
'weapsubtype2_txt',
'weaptype3',
'weaptype3_txt',
'weapsubtype3',
'weapsubtype3_txt',
'weaptype4',
'weaptype4_txt',
'weapsubtype4',
'weapsubtype4_txt',
'weapdetail',
'nkill',
'nkillus',
'nkillter',
'nwound',
'nwoundus',
```

```
 'nwoundte',
 'property',
 'propextent',
 'propextent_txt',
 'propvalue',
 'propcomment',
 'ishostkid',
 'nhostkid',
 'nhostkidus',
 'nhours',
 'ndays',
 'divert',
 'kidhijcountry',
 'ransom',
 'ransomamt',
 'ransomamtus',
 'ransompaid',
 'ransompaidus',
 'ransomnote',
 'hostkidoutcome',
 'hostkidoutcome_txt',
 'nreleased',
 'addnotes',
 'scite1',
 'scite2',
 'scite3',
 'dbsource',
 'INT_LOG',
 'INT_IDEO',
 'INT_MISC',
 'INT_ANY',
 'related']
```

In [10]:

```
# rename some columns which needed it to exploring.
df.rename(columns={'eventid':'id','iyear':'Year','imonth':'Month','iday':'Day','country_tx
                       'region_txt':'Region','attacktype1_txt':'AttackType','target1':'Ta
                       'nwound':'Wounded','summary':'Summary','gname':'Group','targtype1_t
                       'weaptype1_txt':'Weapon_type','motive':'Motive'},inplace=True )
```

Cheat Sheet showing the missing value & some other features!!

In [11]:

```
summary = pd.DataFrame({
    'Column Name': df.columns,
    'Data Type': df.dtypes,
    'Number of Nulls': df.isnull().sum(),
    'Percentage of Nulls': (df.isnull().sum() / len(df)) * 100,
    'Count of Non-Null Data': df.notnull().sum(),
    'number Unique Values': df.nunique()
})

summary['Percentage of Nulls'] = summary['Percentage of Nulls'].map('{:.4f}%'.format)

# # Ensure column names are unique
# summary.columns = [f'{col}_{i}' if summary.columns.tolist().count(col) > 1 else col
#                    for i, col in enumerate(summary.columns)]

# # Define a function to highlight rows where Number of Nulls > 0
# def highlight_nulls(row):
#     return ['background-color: yellow' if row['Number of Nulls'] > 0 else '' for _ in ro
```

```
# # Apply the styling to the DataFrame
# styled_summary = summary.style.apply(highlight_nulls, axis=1)

# # Display the styled DataFrame
# styled_summary

summary = summary.reset_index(drop=True)
summary.set_index('Column Name', inplace=True)
```

In [12]:

```
summary.T
```

Out[12]:

Column Name	id	Year	Month	Day	approxdate	extended	resolution	country	Country	region	Re
Data Type	int64	int64	int64	int64	object	int64	object	int64	object	int64	c
Number of Nulls	0	0	0	0	172452	0	179471	0	0	0	
Percentage of Nulls	0.0000%	0.0000%	0.0000%	0.0000%	94.9150%	0.0000%	98.7781%	0.0000%	0.0000%	0.0000%	0.0(
Count of Non-Null Data	181691	181691	181691	181691	9239	181691	2220	181691	181691	181691	18
number Unique Values	181691	47	13	32	2244	2	1859	205	205	12	

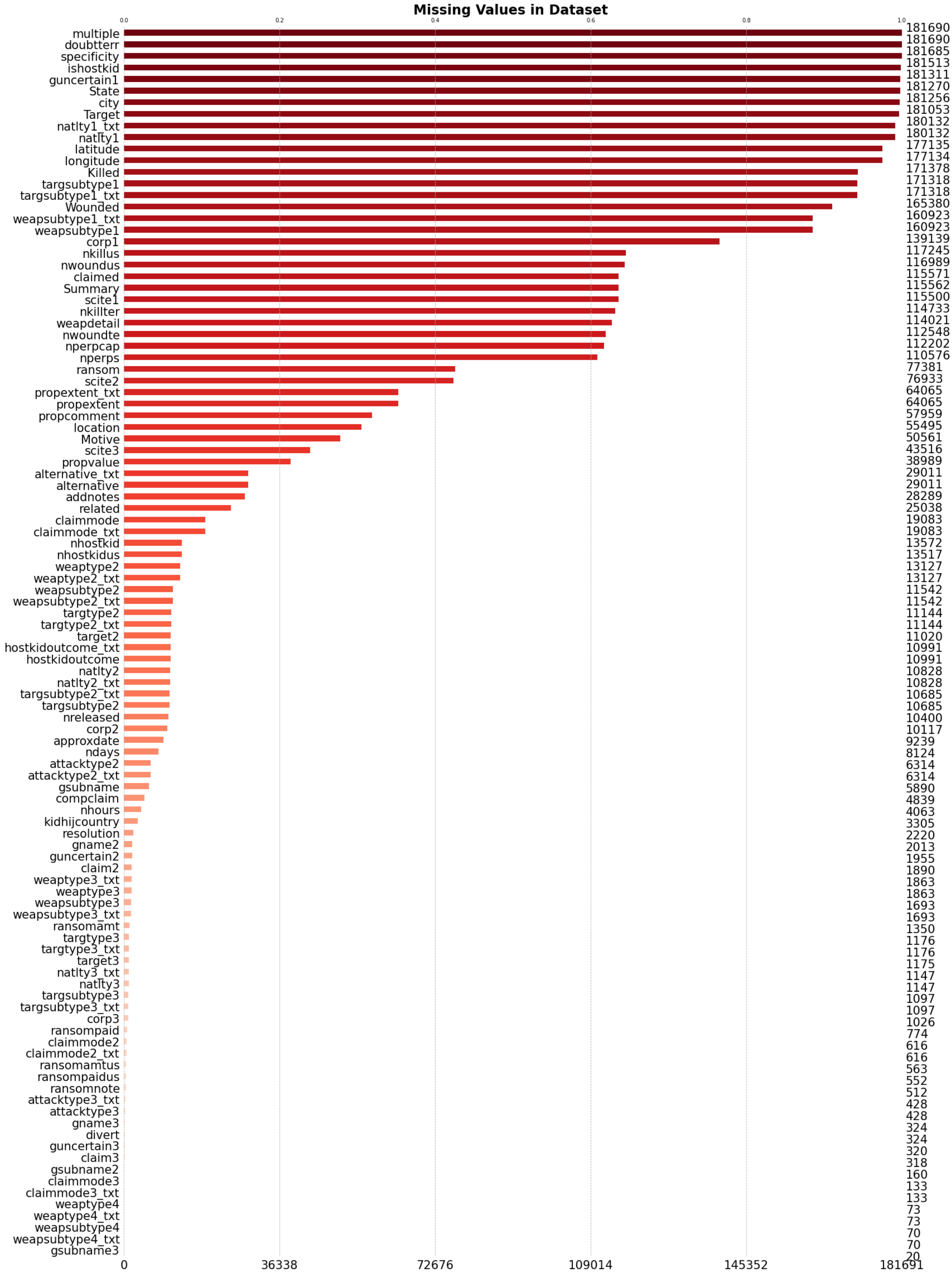
In [13]:

```
# ((df.isna().sum()/len(df))*100).sort_values()
```

In [ ]:

In [14]:

```
# A plot shows just how much the data is missing values
plt.style.use('default')
missing_values = df.isna().sum()
miss_value = missing_values[missing_values > 0]

# rest_miss = len(df)- miss_value
# rest_miss [miss_value.index]

msno.bar(df[miss_value.index],fontsize=21,sort="ascending",color=sns.color_palette("Reds",
plt.title('Missing Values in Dataset ', fontsize=24, weight='bold')
plt.grid(axis='x', linestyle='--', linewidth=0.9, alpha=0.9)
plt.show()
# msno.bar(df,fontsize=21,color="b",sort="descending")
```



In [15]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 181691 entries, 0 to 181690
```

```
Columns: 135 entries, id to related
dtypes: float64(55), int64(22), object(58)
memory usage: 187.1+ MB
```

In [16]:

```
df.dtypes.value_counts()
```

Out[16]:

```
object     58
float64    55
int64      22
Name: count, dtype: int64
```

In [17]:

```
df.iloc[:,1:].describe()
```

Out[17]:

	Year	Month	Day	extended	country	region	latitude
count	181691.000000	181691.000000	181691.000000	181691.000000	181691.000000	181691.000000	177135.000000
mean	2002.638997	6.467277	15.505644	0.045346	131.968501	7.160938	23.498343
std	13.259430	3.388303	8.814045	0.208063	112.414535	2.933408	18.569242
min	1970.000000	0.000000	0.000000	0.000000	4.000000	1.000000	-53.154613
25%	1991.000000	4.000000	8.000000	0.000000	78.000000	5.000000	11.510046
50%	2009.000000	6.000000	15.000000	0.000000	98.000000	6.000000	31.467463
75%	2014.000000	9.000000	23.000000	0.000000	160.000000	10.000000	34.685087
max	2017.000000	12.000000	31.000000	1.000000	1004.000000	12.000000	74.633553

#df= df.dropna(thresh=len(df)*.6,axis=1)

In [18]:

```
missing_values = df.isna().sum()
miss_value_prec = (missing_values[missing_values > 0]/len(df))*100
miss_value = missing_values[missing_values > 0]
miss_value
```

Out[18]:

```
approxdate            172452
resolution            179471
State                    421
city                     435
latitude                4556
longitude               4557
specificity                6
location              126196
Summary                66129
doubtterr                  1
alternative           152680
alternative_txt       152680
multiple                   1
attacktype2           175377
attacktype2_txt       175377
attacktype3           181263
attacktype3_txt       181263
targsubtype1           10373
targsubtype1_txt       10373
corp1                  42552
Target                   638
natlty1                 1559
natlty1_txt             1559
targtype2             170547
targtype2_txt         170547
targsubtype2          171006
targsubtype2_txt      171006
```

corp2	171574
target2	170671
natlty2	170863
natlty2_txt	170863
targtype3	180515
targtype3_txt	180515
targsubtype3	180594
targsubtype3_txt	180594
corp3	180665
target3	180516
natlty3	180544
natlty3_txt	180544
gsubname	175801
gname2	179678
gsubname2	181531
gname3	181367
gsubname3	181671
Motive	131130
guncertain1	380
guncertain2	179736
guncertain3	181371
nperps	71115
nperpcap	69489
claimed	66120
claimmode	162608
claimmode_txt	162608
claim2	179801
claimmode2	181075
claimmode2_txt	181075
claim3	181373
claimmode3	181558
claimmode3_txt	181558
compclaim	176852
weapsubtype1	20768
weapsubtype1_txt	20768
weaptype2	168564
weaptype2_txt	168564
weapsubtype2	170149
weapsubtype2_txt	170149
weaptype3	179828
weaptype3_txt	179828
weapsubtype3	179998
weapsubtype3_txt	179998
weaptype4	181618
weaptype4_txt	181618
weapsubtype4	181621
weapsubtype4_txt	181621
weapdetail	67670
Killed	10313
nkillus	64446
nkillter	66958
Wounded	16311
nwoundus	64702
nwoundte	69143
propextent	117626
propextent_txt	117626
propvalue	142702
propcomment	123732
ishostkid	178
nhostkid	168119
nhostkidus	168174
nhours	177628
ndays	173567
divert	181367
kidhijcountry	178386
ransom	104310

```
ransomamt              180341
ransomamtus            181128
ransompaid             180917
ransompaidus           181139
ransomnote             181179
hostkidoutcome         170700
hostkidoutcome_txt     170700
nreleased              171291
addnotes               153402
scite1                  66191
scite2                 104758
scite3                 138175
related                156653
dtype: int64
```

In [19]:

```python
# features contain nulls
indx_na = df.isna().sum()
indx_na[indx_na >0].index.to_list()
```

Out[19]:

```
['approxdate',
 'resolution',
 'State',
 'city',
 'latitude',
 'longitude',
 'specificity',
 'location',
 'Summary',
 'doubtterr',
 'alternative',
 'alternative_txt',
 'multiple',
 'attacktype2',
 'attacktype2_txt',
 'attacktype3',
 'attacktype3_txt',
 'targsubtype1',
 'targsubtype1_txt',
 'corp1',
 'Target',
 'natlty1',
 'natlty1_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',
 'guncertain1',
```

```
 'guncertain2',
 'guncertain3',
 'nperps',
 'nperpcap',
 'claimed',
 'claimmode',
 'claimmode_txt',
 'claim2',
 'claimmode2',
 'claimmode2_txt',
 'claim3',
 'claimmode3',
 'claimmode3_txt',
 'compclaim',
 'weapsubtype1',
 'weapsubtype1_txt',
 'weaptype2',
 'weaptype2_txt',
 'weapsubtype2',
 'weapsubtype2_txt',
 'weaptype3',
 'weaptype3_txt',
 'weapsubtype3',
 'weapsubtype3_txt',
 'weaptype4',
 'weaptype4_txt',
 'weapsubtype4',
 'weapsubtype4_txt',
 'weapdetail',
 'Killed',
 'nkillus',
 'nkillter',
 'Wounded',
 'nwoundus',
 'nwoundte',
 'propextent',
 'propextent_txt',
 'propvalue',
 'propcomment',
 'ishostkid',
 'nhostkid',
 'nhostkidus',
 'nhours',
 'ndays',
 'divert',
 'kidhijcountry',
 'ransom',
 'ransomamt',
 'ransomamtus',
 'ransompaid',
 'ransompaidus',
 'ransomnote',
 'hostkidoutcome',
 'hostkidoutcome_txt',
 'nreleased',
 'addnotes',
 'scite1',
 'scite2',
 'scite3',
 'related']
```

In [20]:

```
## features not contain nulls
indx_notna = df.notna().sum()
indx_notna[indx_notna == len(df)].index.to_list()
```

Out[20]:
```
['id',
 'Year',
 'Month',
 'Day',
 'extended',
 'country',
 'Country',
 'region',
 'Region',
 'vicinity',
 'crit1',
 'crit2',
 'crit3',
 'success',
 'suicide',
 'attacktype1',
 'AttackType',
 'targtype1',
 'Target_type',
 'Group',
 'individual',
 'weaptype1',
 'Weapon_type',
 'property',
 'dbsource',
 'INT_LOG',
 'INT_IDEO',
 'INT_MISC',
 'INT_ANY']
```

numeric_cols = df.select_dtypes(include=[np.number]) # Calculating mean, median, and standard deviation mean_values = numeric_cols.mean() median_values = numeric_cols.median() std_dev_values = numeric_cols.std()mean_values # dask import dask.dataframe as dd dtype={'approxdate': 'object', 'attacktype2_txt': 'object', 'attacktype3_txt': 'object', 'claimmode2_txt': 'object', 'claimmode3_txt': 'object', 'corp2': 'object', 'corp3': 'object', 'divert': 'object', 'gname2': 'object', 'gname3': 'object', 'gsubname': 'object', 'gsubname2': 'object', 'gsubname3': 'object', 'guncertain1': 'float64', 'hostkidoutcome_txt': 'object', 'ishostkid': 'float64', 'natlty1': 'float64', 'natlty2_txt': 'object', 'natlty3_txt': 'object', 'ransom': 'float64', 'ransomnote': 'object', 'related': 'object', 'resolution': 'object', 'specificity': 'float64', 'target2': 'object', 'target3': 'object', 'targsubtype1': 'float64', 'targsubtype2_txt': 'object', 'targsubtype3_txt': 'object', 'targtype2_txt': 'object', 'targtype3_txt': 'object', 'weapsubtype2_txt': 'object', 'weapsubtype3_txt': 'object', 'weapsubtype4_txt': 'object', 'weaptype2_txt': 'object', 'weaptype3_txt': 'object', 'weaptype4_txt': 'object'} start_time = time.time() # Read the CSV file into a Dask DataFrame df_dd = dd.read_csv('globalterrorismdb_0718dist.csv',dtype=dtype, encoding='ISO-8859-1',low_memory=False) dask_duration = time.time() - start_time dask_duration df_dd.head()

In [21]:
```
df['resolution'] = pd.to_datetime(df['resolution'], format='%m/%d/%Y')
```

In [ ]:
```
```

- Numerical Features

In [22]:
```
# only numeric columns.
numeric_cols = df.select_dtypes(include=[np.number])
numeric_cols
```

Out[22]:

	id	Year	Month	Day	extended	country	region	latitude	longitude	specificity	vicinity	c
0	197000000001	1970	7	2	0	58	2	18.456792	-69.951164	1.0	0	
1	197000000002	1970	0	0	0	130	1	19.371887	-99.086624	1.0	0	
2	197001000001	1970	1	0	0	160	5	15.478598	120.599741	4.0	0	
3	197001000002	1970	1	0	0	78	8	37.997490	23.762728	1.0	0	

	id	Year	Month	Day	extended	country	region	latitude	longitude	specificity	vicinity	c
4	197001000003	1970	1	0	0	101	4	33.580412	130.396361	1.0	0	
...	...	...	...	...	...	...	...	...	...	...	...	
181686	201712310022	2017	12	31	0	182	11	2.359673	45.385034	2.0	0	
181687	201712310029	2017	12	31	0	200	10	35.407278	35.942679	1.0	1	
181688	201712310030	2017	12	31	0	160	5	6.900742	124.437908	2.0	0	
181689	201712310031	2017	12	31	0	92	6	24.798346	93.940430	1.0	0	
181690	201712310032	2017	12	31	0	160	5	7.209594	124.241966	1.0	0	

181691 rows × 77 columns

In [23]:

```
# Calculate the mean, median, and standard deviation of relevant numeric columns.
numeric_cols = df.select_dtypes(include=[np.number])

numeric_summary = pd.DataFrame({
    'Mean': numeric_cols.mean(),
    'Median': numeric_cols.median(),
    'Standard Deviation': numeric_cols.std()
})

numeric_summary = numeric_summary.reset_index().rename(columns={'index': 'Column Name'})[1
numeric_summary
```

Out[23]:

Column Name	Year	Month	Day	extended	country	region	latitude	longitude	specificity	
Mean	2002.638997	6.467277	15.505644	0.045346	131.968501	7.160938	23.498343	-458.695653	1.451452	(
Median	2009.000000	6.000000	15.000000	0.000000	98.000000	6.000000	31.467463	43.246506	1.000000	(
Standard Deviation	13.259430	3.388303	8.814045	0.208063	112.414535	2.933408	18.569242	204778.988611	0.995430	(

- Categorical Features

In [24]:

```
#  most frequent values in categorical columns.
catego_cols = df.select_dtypes(include=['object'])
catego_cols
```

Out[24]:

	approxdate	Country	Region	State	city	location	Summary	alternative_txt	
0	NaN	Dominican Republic	Central America & Caribbean	NaN	Santo Domingo	NaN	NaN	NaN	
1	NaN	Mexico	North America	Federal	Mexico city	NaN	NaN	NaN	
2	NaN	Philippines	Southeast Asia	Tarlac	Unknown	NaN	NaN	NaN	
3	NaN	Greece	Western Europe	Attica	Athens	NaN	NaN	NaN	E

	approxdate	Country	Region	State	city	location	Summary	alternative_txt	
4	NaN	Japan	East Asia	Fukouka	Fukouka	NaN	NaN	NaN	Fa
...	...	...	...	...	...	...	...	...	
181686	NaN	Somalia	Sub-Saharan Africa	Middle Shebelle	Ceelka Geelow	The incident occurred near the town of Balcad.	12/31/2017: Assailants opened fire on a Somali...	Insurgency/Guerilla Action	
181687	NaN	Syria	Middle East & North Africa	Lattakia	Jableh	The incident occurred at the Humaymim Airport.	12/31/2017: Assailants launched mortars at the...	Insurgency/Guerilla Action	E
181688	NaN	Philippines	Southeast Asia	Maguindanao	Kubentog	The incident occurred in the Datu Hoffer distr...	12/31/2017: Assailants set fire to houses in K...	NaN	Fa
181689	NaN	India	South Asia	Manipur	Imphal	The incident occurred in the Mantripukhri neig...	12/31/2017: Assailants threw a grenade at a Fo...	NaN	E
181690	NaN	Philippines	Southeast Asia	Maguindanao	Cotabato City	NaN	12/31/2017: An explosive device was discovered...	NaN	E

181691 rows × 57 columns

In [25]:

```
# Find the most frequent value (mode) for each categorical column
most_frequent = catego_cols.mode()
most_frequent.index =["most_freq_"]
most_frequent
```

Out[25]:

	approxdate	Country	Region	State	city	location	Summary	alternative_txt	Attac
most_freq_	September 18-24, 2016	Iraq	Middle East & North Africa	Baghdad	Unknown	The attack took place in Baghdad, Baghdad, Iraq.	09/00/2016: Sometime between September 18, 201...	Insurgency/Guerilla Action	Bombing/Exp

Time to analysis!

Given the extensive number of columns in the dataset, we'll focus on selecting only the key columns for data preprocessing to ensure a more efficient and manageable analysis. By concentrating on the most relevant columns, we can streamline our efforts and derive meaningful insights from the dataset.

In [26]:

```
#As we have many columns, we take the columns that are necessary for analysis.
df_terr = df[['id','Year','Month','Day','Country','Region','State','city','latitude','long
             'Target','Summary','Group','Target_type','Weapon_type','Motive','success']]
df_terr.head()
```

Out[26]:

	id	Year	Month	Day	Country	Region	State	city	latitude	longitude	Attac
0	197000000001	1970	7	2	Dominican Republic	Central America & Caribbean	NaN	Santo Domingo	18.456792	-69.951164	Assassi
1	197000000002	1970	0	0	Mexico	North America	Federal	Mexico city	19.371887	-99.086624	Hostage (Kidna
2	197001000001	1970	1	0	Philippines	Southeast Asia	Tarlac	Unknown	15.478598	120.599741	Assassi
3	197001000002	1970	1	0	Greece	Western Europe	Attica	Athens	37.997490	23.762728	Bombing/Exp
4	197001000003	1970	1	0	Japan	East Asia	Fukouka	Fukouka	33.580412	130.396361	Facility/Infrastr

In [27]:

```
# to get on overall casualties killing and wounding.
df_terr['casualties'] = df_terr['Killed'].fillna(0) + df_terr['Wounded'].fillna(0)
```

In [28]:

```
df_terr.shape
```

Out[28]:

```
(181691, 21)
```

In [29]:

```
df_terr.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 181691 entries, 0 to 181690
Data columns (total 21 columns):
 #   Column       Non-Null Count   Dtype  
---  ------       --------------   -----  
 0   id           181691 non-null  int64  
 1   Year         181691 non-null  int64  
 2   Month        181691 non-null  int64  
 3   Day          181691 non-null  int64  
 4   Country      181691 non-null  object 
 5   Region       181691 non-null  object 
 6   State        181270 non-null  object 
 7   city         181256 non-null  object 
 8   latitude     177135 non-null  float64
 9   longitude    177134 non-null  float64
 10  AttackType   181691 non-null  object 
 11  Killed       171378 non-null  float64
 12  Wounded      165380 non-null  float64
 13  Target       181053 non-null  object 
 14  Summary      115562 non-null  object 
```

```
 15  Group         181691 non-null  object
 16  Target_type   181691 non-null  object
 17  Weapon_type   181691 non-null  object
 18  Motive        50561 non-null   object
 19  success       181691 non-null  int64
 20  casualties    181691 non-null  float64
dtypes: float64(5), int64(5), object(11)
memory usage: 29.1+ MB
```

In [30]:

```
# Calculate the mean, median, and standard deviation of relevant numeric columns -- on new
num_cols = df_terr.select_dtypes(include=[np.number])
```

In [31]:

```
num_summary = pd.DataFrame({
    'Mean': numeric_cols.mean(),
    'Median': numeric_cols.median(),
    'Standard Deviation': numeric_cols.std()
})

numeric_view = num_summary.reset_index().rename(columns={'index': 'Column Name'})[1:].set_
numeric_view
```

Out[31]:

Column Name	Year	Month	Day	extended	country	region	latitude	longitude	specificity
Mean	2002.638997	6.467277	15.505644	0.045346	131.968501	7.160938	23.498343	-458.695653	1.451452
Median	2009.000000	6.000000	15.000000	0.000000	98.000000	6.000000	31.467463	43.246506	1.000000
Standard Deviation	13.259430	3.388303	8.814045	0.208063	112.414535	2.933408	18.569242	204778.988611	0.995430

Identify the most frequent values in categorical columns. cate_cols = df_terr.select_dtypes(include=['object'])

In [32]:

```
# Find the most frequent value (mode) for each categorical column
most_freq = catego_cols.mode()
most_freq.index =["most_freq_"]
most_freq
```

Out[32]:

	approxdate	Country	Region	State	city	location	Summary	alternative_txt	Attac
most_freq_	September 18-24, 2016	Iraq	Middle East & North Africa	Baghdad	Unknown	The attack took place in Baghdad, Baghdad, Iraq.	09/00/2016: Sometime between September 18, 201...	Insurgency/Guerilla Action	Bombing/Ex

In [33]:

```
df_terr.iloc[:,1:].corr(numeric_only=True) # correlation between numerical features
```

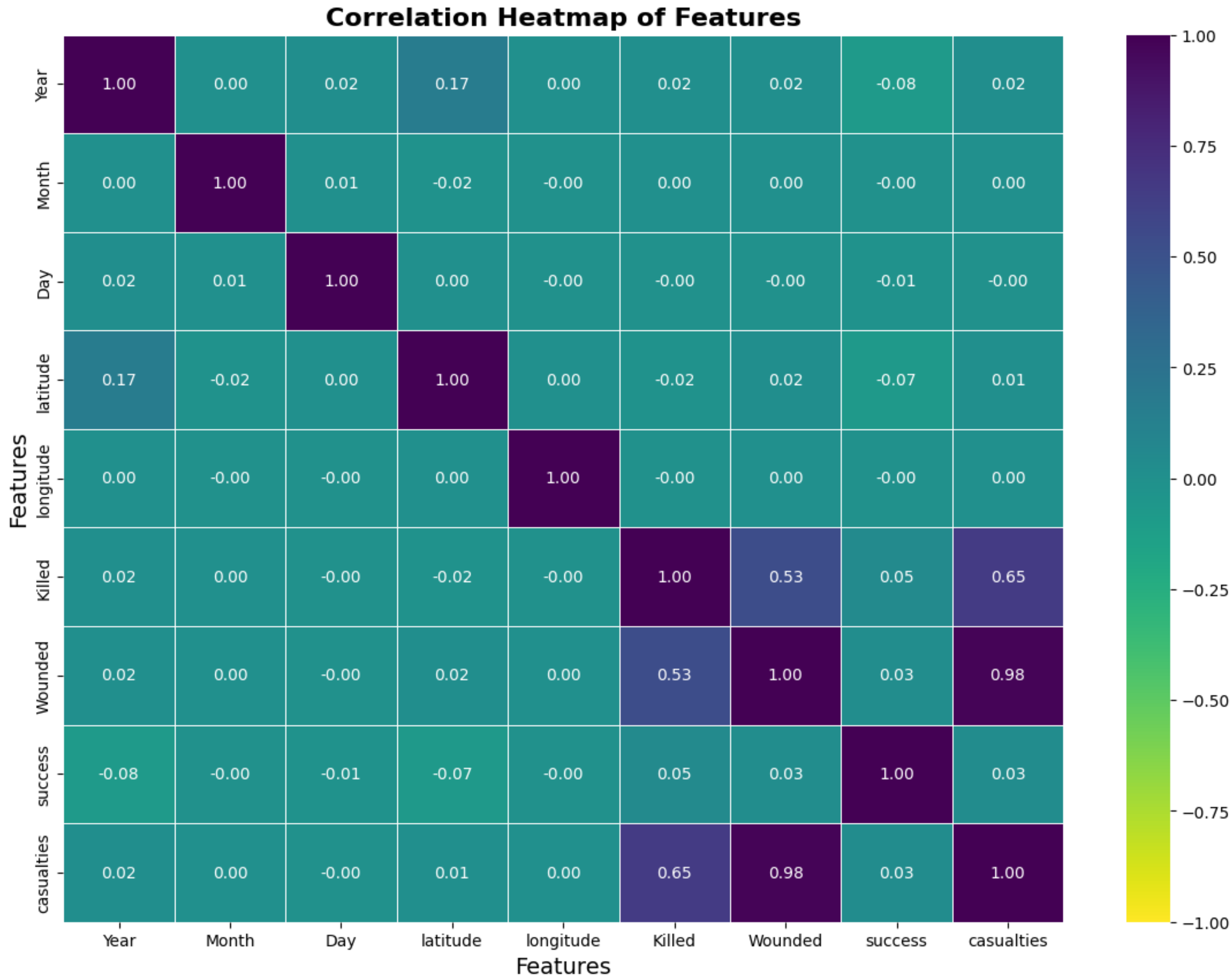
Out[33]:

	Year	Month	Day	latitude	longitude	Killed	Wounded	success	casualties
Year	1.000000	0.000139	0.018254	0.166933	0.003917	0.015341	0.015273	-0.082963	0.020675
Month	0.000139	1.000000	0.005497	-0.015978	-0.003880	0.003463	0.002938	-0.002845	0.003805
Day	0.018254	0.005497	1.000000	0.003423	-0.002285	-0.003693	-0.001268	-0.011802	-0.001808
latitude	0.166933	-0.015978	0.003423	1.000000	0.001463	-0.018124	0.015988	-0.073715	0.009899
longitude	0.003917	-0.003880	-0.002285	0.001463	1.000000	-0.000562	0.000223	-0.000858	0.000013

	Year	Month	Day	latitude	longitude	Killed	Wounded	success	casualties
Killed	0.015341	0.003463	-0.003693	-0.018124	-0.000562	1.000000	0.534375	0.053115	0.651615
Wounded	0.015273	0.002938	-0.001268	0.015988	0.000223	0.534375	1.000000	0.025804	0.980386
success	-0.082963	-0.002845	-0.011802	-0.073715	-0.000858	0.053115	0.025804	1.000000	0.033487
casualties	0.020675	0.003805	-0.001808	0.009899	0.000013	0.651615	0.980386	0.033487	1.000000

In [34]:

```
#Heatmap to visualize the correlation between numerical features
import pandas as pd
import seaborn as sns
import matplotlib.pyplot

correlation_matrix =  df_terr.iloc[:,1:].corr(numeric_only=True)
# correlation_matrix=num_cols.corr()
plt.style.use('default')

# Create the heatmap
plt.figure(figsize=(14, 10))
sns.heatmap(correlation_matrix, annot=True, cmap=sns.mpl_palette("viridis_r", as_cmap=True

# Add labels and title
plt.title('Correlation Heatmap of Features', fontsize=16, fontweight='bold')
plt.xlabel('Features', fontsize=14)
plt.ylabel('Features', fontsize=14)

# Show the plot
plt.show()
```



> From a glance at the diagram, it becomes clear that the extent to which the numerical features are related to each other suggests a weak correlation.

In [35]:

```
# number terrsiom attacks per year
attacks_per_year = df_terr.groupby('Year').size().reset_index(name='number of attacks per
attacks_per_year.T
```

Out[35]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Year	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986
number of attacks per year	651	471	568	473	581	740	923	1319	1526	2662	2662	2586	2544	2870	3495	2915	2860

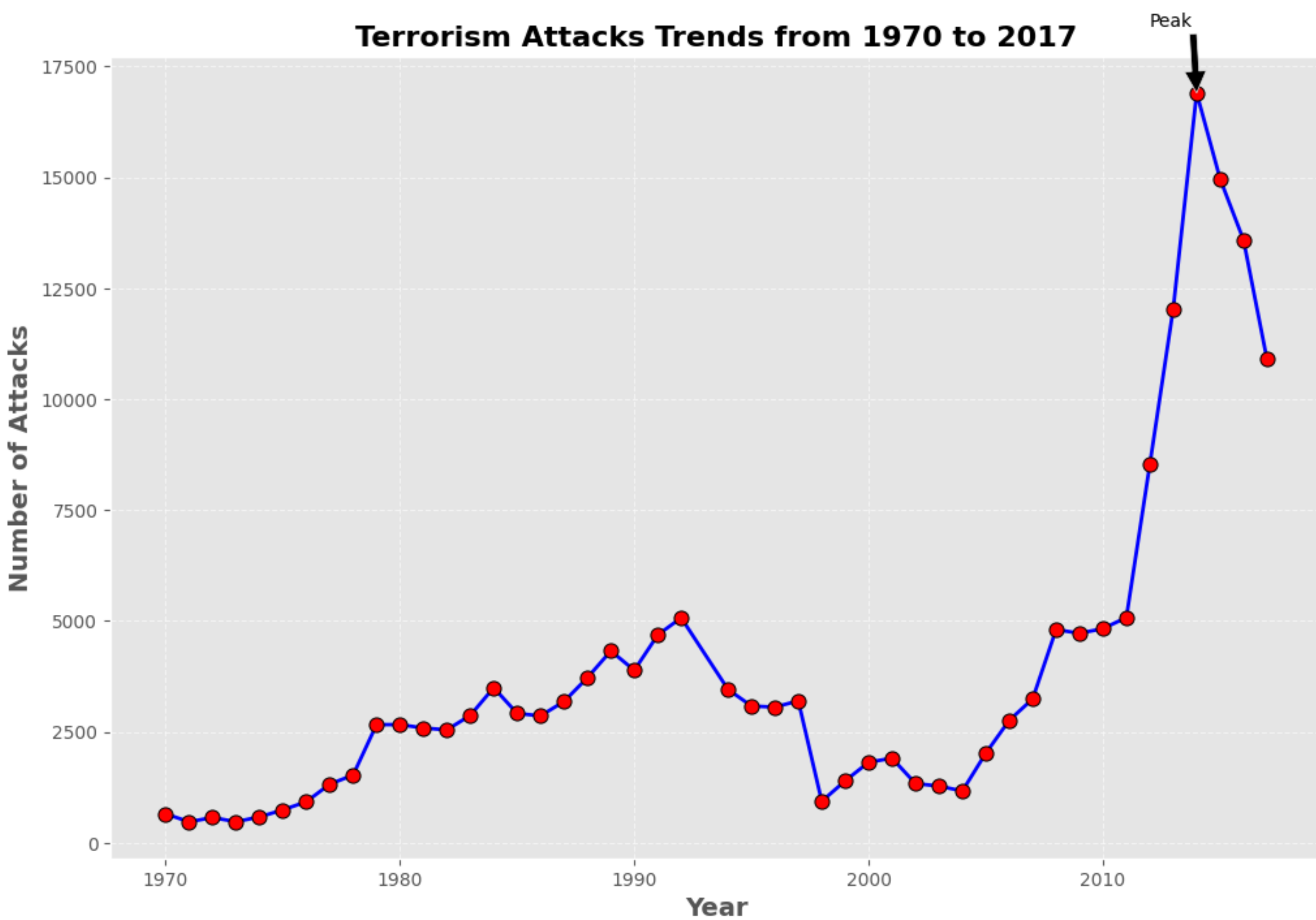
In [36]:

```
# attacks_per_year.loc[attacks_per_year['number of attacks per year'].idxmax(), 'Year']
```

In [37]:

```
# sns.lineplot(x='Year', y='number of attacks per year', data=attacks_per_year, color='b',
```

```
plt.style.use('ggplot')
plt.figure(figsize=(12, 8))

# Create the line plot
sns.lineplot(
    x='Year',y='number of attacks per year',data=attacks_per_year,color='b',marker='o',
#     linestyle=':',  # Line style
    linewidth=2, markersize=8,
    markerfacecolor='red',   # Marker fill color
    markeredgecolor='black'   # Marker edge color
)

# Add gridlines for better readability
plt.grid(True, linestyle='--', alpha=0.6)

# Add labels and title
plt.xlabel('Year', fontsize=14, fontweight='bold')
plt.ylabel('Number of Attacks', fontsize=14, fontweight='bold')
plt.title("Terrorism Attacks Trends from 1970 to 2017", fontsize=16, fontweight='bold')

# Add annotations or highlights (optional)
plt.annotate('Peak', xy=(attacks_per_year.loc[attacks_per_year['number of attacks per year
             xytext=(2012, attacks_per_year['number of attacks per year'].max()+1500),
             arrowprops=dict(facecolor='black'))

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



> We conclude from here that most terrorist attacks are concentrated in `2014`.

- Here are the five largest attacks over the years.

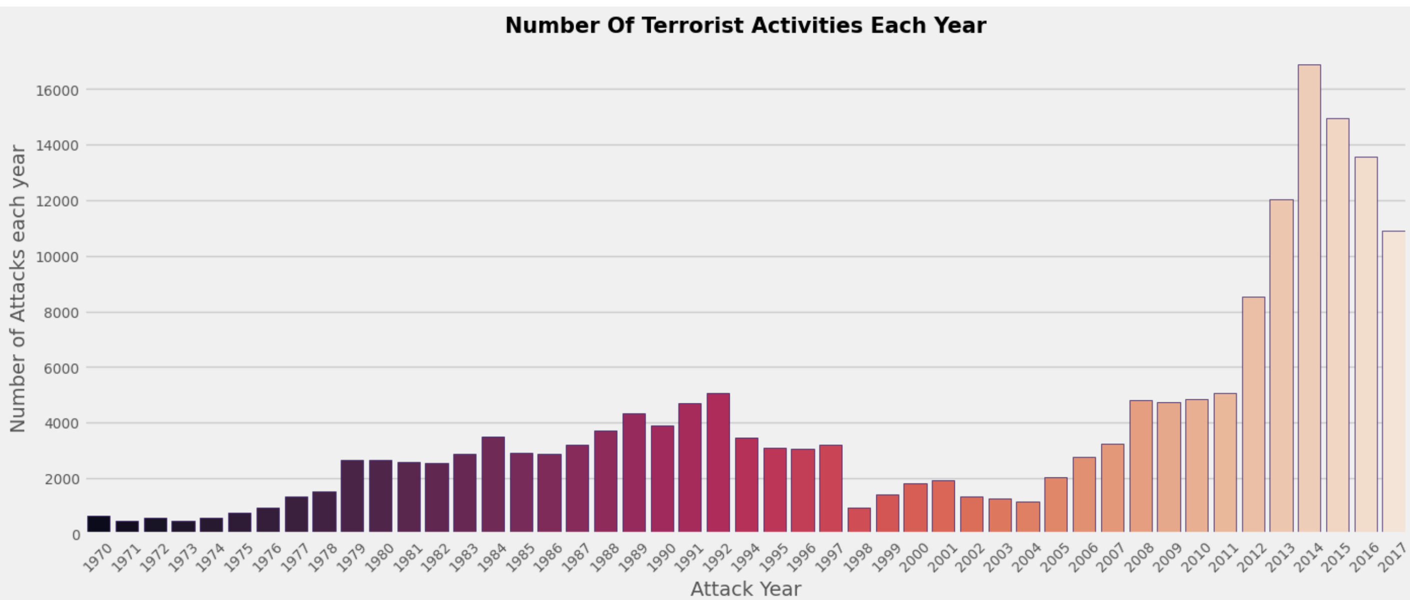
In [38]:
```
attacks_per_year.nlargest(5,'number of attacks per year')
```

Out[38]:

	Year	number of attacks per year
43	2014	16903
44	2015	14965
45	2016	13587
42	2013	12036
46	2017	10900

Plot showing Terrorist Activities Each Year

In [39]:
```
# Number Of Terrorist Activities Each Year
plt.style.use('fivethirtyeight')
plt.subplots(figsize=(15,6))
sns.countplot(x='Year',data=df_terr,palette='rocket',edgecolor=sns.color_palette('magma',5


plt.xlabel('Attack Year', fontsize=14)
plt.ylabel('Number of Attacks each year', fontsize=14)
plt.xticks(rotation=45,fontsize=10)
plt.yticks(fontsize=10)

plt.title('Number Of Terrorist Activities Each Year',fontsize=15, fontweight='bold')
# plt.tight_layout()

plt.show()
```



Total active terrorist attacks on region.

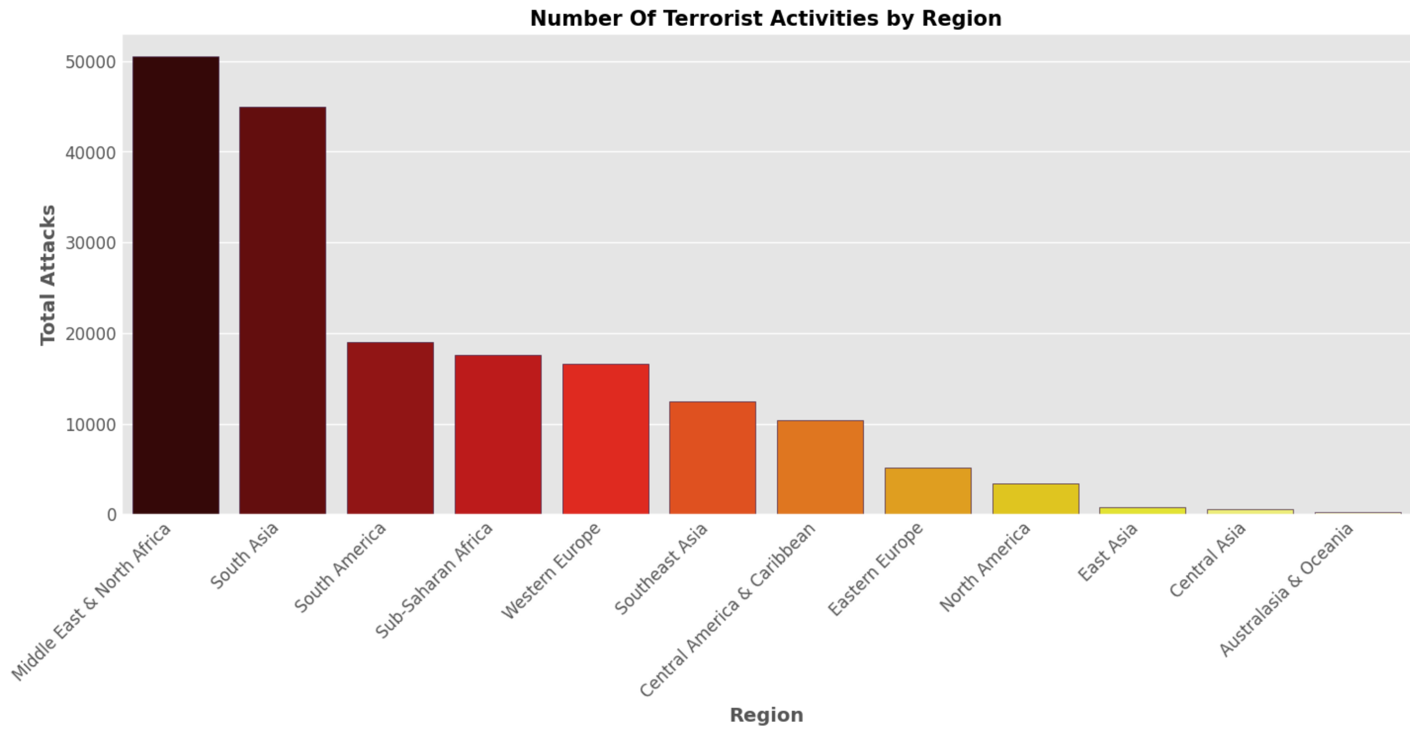
In [40]:
```
# numbers terrrosim attacks on region
attacks_per_region = df_terr.groupby('Region').size().reset_index(name='Total_attacks_regi
attacks_per_region.sort_values(by='Total_attacks_region',ascending=False)
```

Out[40]:

	Region	Total_attacks_region
5	Middle East & North Africa	50474
8	South Asia	44974
7	South America	18978
10	Sub-Saharan Africa	17550
11	Western Europe	16639
9	Southeast Asia	12485
1	Central America & Caribbean	10344
4	Eastern Europe	5144
6	North America	3456
3	East Asia	802
2	Central Asia	563
0	Australasia & Oceania	282

In [41]:

```
plt.style.use('ggplot')
plt.subplots(figsize=(15, 6))
sns.barplot(x='Region',y='Total_attacks_region' ,
            data=attacks_per_region,palette='hot',
            order=df_terr['Region'].value_counts().index,edgecolor=sns.color_palette('rock

# Customize y-axis
plt.yticks(fontsize=12)

# Add labels and title
plt.xlabel('Region', fontsize=14,fontweight='bold')
plt.ylabel('Total Attacks', fontsize=14,fontweight='bold')
plt.title('Number Of Terrorist Activities by Region', fontsize=15, fontweight='bold')

plt.xticks(rotation=45,fontsize=12,ha='right')
plt.yticks(fontsize=12)


# Adjust layout to prevent overlapping
# plt.tight_layout()

# Show the plot
plt.show()
```



- `Middle East and North Africa` are the most terrorism prone regions followed by `South Asia`. The `Australian` Region have experienced very few terrorist events. Collectively we can say that The African and Asian Continent experience the highest terrorist attacks.

Sheet showing the number of terrorism attacks per `region` each `year`.

In [42]:

```
# sheet showing number terrosim attacks on region per year .

attc_region_per_year=pd.crosstab(df_terr.Year,df_terr.Region)

attc_region_per_year['[sum_attacks on region per year]'] = attc_region_per_year.sum(axis=
attc_region_per_year.loc['[Total_attac_region]']= attc_region_per_year.sum()
attc_region_per_year.T
```

Out[42]:

Year	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	198
Region																
Australasia & Oceania	1	1	8	1	1	0	0	0	2	2	7	3	2	0	11	
Central America & Caribbean	7	5	3	6	11	9	45	24	199	609	1070	1148	996	858	681	78
Central Asia	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
East Asia	2	1	0	2	4	12	2	4	35	16	1	4	3	13	15	1
Eastern Europe	12	5	1	1	2	0	0	2	2	1	1	4	3	2	4	
Middle East & North Africa	28	55	53	19	42	44	55	211	128	455	437	312	290	334	268	13

Year	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	198
Region																
North America	472	247	73	64	111	159	125	149	117	79	75	77	86	47	67	4
South America	65	24	33	83	81	55	91	119	222	236	319	383	639	950	1492	104
South Asia	1	0	1	1	2	4	4	2	2	34	12	23	20	63	244	16
Southeast Asia	10	6	16	2	3	7	12	8	44	86	87	50	43	22	46	12
Sub-Saharan Africa	3	2	4	4	7	12	11	29	46	124	58	98	60	106	126	14
Western Europe	50	125	376	290	317	438	578	771	729	1020	595	484	402	475	541	46
[sum_attacks on region per year]	651	471	568	473	581	740	923	1319	1526	2662	2662	2586	2544	2870	3495	291

Total terrorist strikes per `country`

In [43]:

```python
# total terssiom attacks on country each year.
attacks_country = df_terr.Country.value_counts().to_frame().reset_index()
attacks_country.columns=['Country','total_attacks per country']
attacks_country.T
```

Out[43]:

	0	1	2	3	4	5	6	7	8	9	10	
Country	Iraq	Pakistan	Afghanistan	India	Colombia	Philippines	Peru	El Salvador	United Kingdom	Turkey	Somalia	N
total_attacks per country	24636	14368	12731	11960	8306	6908	6096	5320	5235	4292	4142	

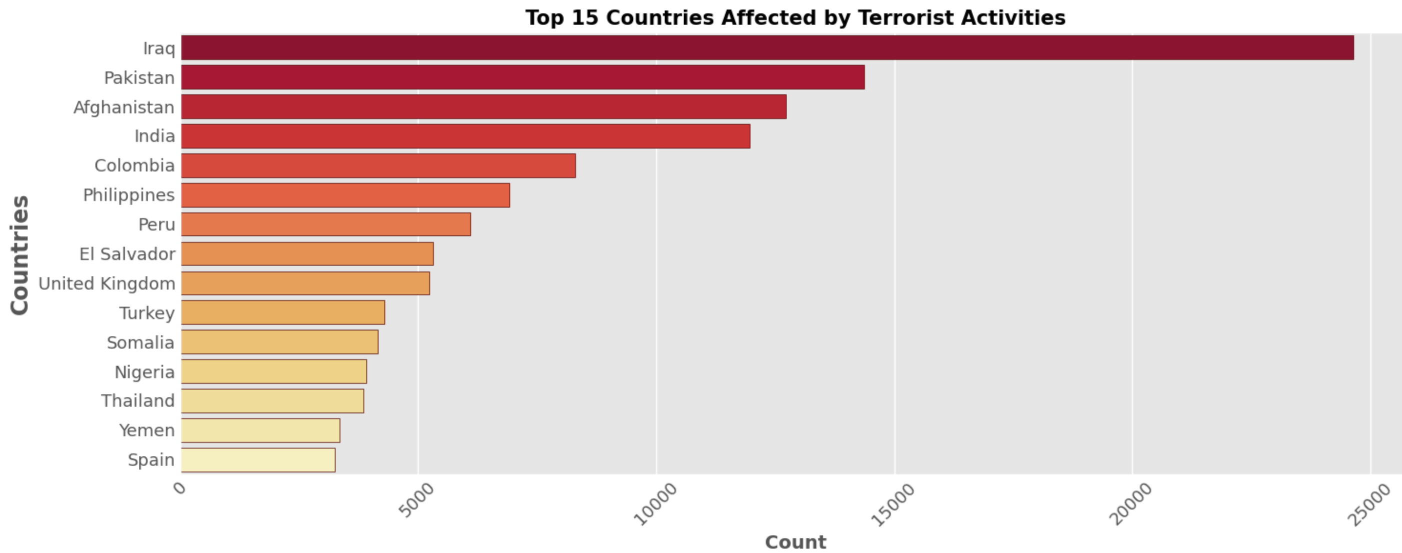
In [44]:

```python
# df_terr.groupby('Country').size()
# attacks_country = df_terr.groupby('Country').size().reset_index()
# attacks_country
```

In [45]:

```python
# plt.style.use('fivethirtyeight')
plt.style.use('ggplot')

plt.figure(figsize=(15, 6))

# Assuming `df_terr` is your DataFrame with a 'Country' column
top_countries = df_terr['Country'].value_counts().nlargest(15)
sns.barplot(y=top_countries.index, x=top_countries.values, palette='YlOrRd_r',orient="h",e

plt.title('Top 15 Countries Affected by Terrorist Activities', fontsize=15, fontweight='bo
plt.ylabel('Countries', fontsize=17,fontweight='semibold')
plt.xlabel('Count', fontsize=14,fontweight='semibold')
```

```
plt.xticks(rotation=45, fontsize=13)
plt.yticks(fontsize=13)

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



From the graph we can see **The Most `5 Targered` Affected Country with Terrorism Attacks** are:

1. **Iraq**
2. **Pakistan**
3. **Afghanistan**
4. **India**
5. **Colombia**

- **`iraq`** has witnessed a very large number of terrorist activities followed by `Pakistan` , `Afghanistan` ..etc . One thing to note is the countries with highest attacks, are mostly densely populated countries, thus it will eventually claim many lives.

---

sheet showing the number of terrorism attacks per `country` each `year`.

In [46]:

```
attac_country_per_year = pd.crosstab(df_terr.Country,df_terr.Year)
attac_country_per_year["[sum_per_year]"] = attac_country_per_year.sum(axis=1)
attac_country_per_year.loc["[sum_attacks on country per year]"] =attac_country_per_year.su
attac_country_per_year.T
```

Out[46]:

Country	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria
Year										
1970	0	0	0	0	0	0	21	0	1	0
1971	0	0	0	0	0	0	7	0	1	0

Country	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria
Year										
1972	0	0	1	0	0	0	20	0	8	7
1973	1	0	0	0	0	0	60	0	0	2
1974	0	0	0	1	0	0	71	0	1	1
1975	0	0	0	0	0	0	38	0	0	2
1976	0	0	1	0	0	0	54	0	0	4
1977	0	0	0	0	1	0	17	0	0	5
1978	0	0	1	0	2	0	25	0	2	1
1979	3	0	1	0	3	0	16	0	2	5
1980	0	0	0	0	0	0	6	0	6	2
1981	0	1	0	0	1	0	2	0	1	3
1982	0	0	0	0	2	0	9	0	2	11
1983	0	0	0	0	7	0	18	0	0	0
1984	0	0	0	0	11	0	46	0	0	4
1985	0	0	0	0	6	0	43	0	0	4
1986	0	0	0	0	5	0	33	0	2	4
1987	1	0	0	0	3	0	80	0	0	4
1988	11	0	0	0	12	0	33	0	3	1
1989	10	0	0	0	12	0	32	0	2	2
1990	2	1	2	0	205	0	31	0	0	1
1991	30	1	30	0	16	0	27	1	4	3
1992	36	3	215	0	50	1	41	2	4	8
1994	9	2	227	0	9	0	14	4	9	5
1995	6	0	185	0	10	0	16	1	5	12
1996	4	6	129	0	4	0	19	1	5	4
1997	1	41	344	0	7	0	11	0	4	1
1998	1	7	151	0	20	0	0	2	6	1
1999	9	3	106	0	34	0	1	2	0	0
2000	14	2	138	0	22	0	0	2	1	0
2001	14	1	113	0	40	0	2	2	2	0
2002	38	0	132	0	6	0	0	0	0	0
2003	100	1	75	0	0	0	1	0	0	0
2004	88	0	67	0	0	0	0	0	0	0
2005	155	0	104	0	0	0	3	0	0	0

Country	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria
Year										
2006	282	0	152	0	0	0	0	0	2	1
2007	341	0	124	0	0	0	0	1	0	1
2008	414	0	107	0	0	0	0	0	3	7
2009	503	1	108	0	1	0	1	1	1	3
2010	542	0	100	0	2	0	5	0	1	0
2011	421	0	15	0	0	0	1	0	0	1
2012	1469	0	41	0	0	0	2	0	0	0
2013	1443	1	22	0	0	0	2	1	1	1
2014	1824	2	13	0	0	0	1	0	8	0
2015	1928	4	16	0	0	0	1	2	14	0
2016	1617	2	9	0	2	0	2	2	9	3
2017	1414	1	14	0	6	0	3	0	4	1
[sum_per_year]	12731	80	2743	1	499	1	815	24	114	115

Total Cacasualties & Killed & Wounded each Country under Region

In [47]:

```
# sheet view total cacasualties & killed & wounded each country under region
results_terr =df_terr[['Region','Country','Killed','Wounded','casualties']]
results_terr = results_terr.groupby(['Region','Country']).sum().sort_values(by='casualties
results_terr.T
```

Out[47]:

	0	1	2	3	4	5	6	7	8	9	10
Region	Middle East & North Africa	South Asia	South Asia	South Asia	Sub-Saharan Africa	South Asia	Middle East & North Africa	South America	North America	Southeast Asia	Middle East & North Africa
Country	Iraq	Afghanistan	Pakistan	India	Nigeria	Sri Lanka	Syria	Colombia	United States	Philippines	Algeria
Killed	78589.0	39384.0	23822.0	19341.0	22682.0	15530.0	15229.0	14698.0	3771.0	9559.0	11066.0
Wounded	134690.0	44277.0	42038.0	28980.0	10239.0	15561.0	14109.0	10328.0	20702.0	13367.0	9150.0
casualties	213279.0	83661.0	65860.0	48321.0	32921.0	31091.0	29338.0	25026.0	24473.0	22926.0	20216.0

In [48]:

```
# killed per year
k = df_terr[["Year", "Killed"]].groupby("Year").sum()
```

```
# wounded per year
w = df_terr[["Year", "Wounded"]].groupby("Year").sum()
print(w.head())
print(k.head())
```

```
      Wounded
Year
1970    212.0
1971     82.0
1972    409.0
1973    495.0
1974    865.0
      Killed
Year
1970   174.0
1971   173.0
1972   566.0
1973   370.0
1974   539.0
```

In [49]:

```
merged_k_w = pd.merge(w, k, on="Year")
merged_k_w.reset_index(inplace=True)
merged_k_w.head(10)
```

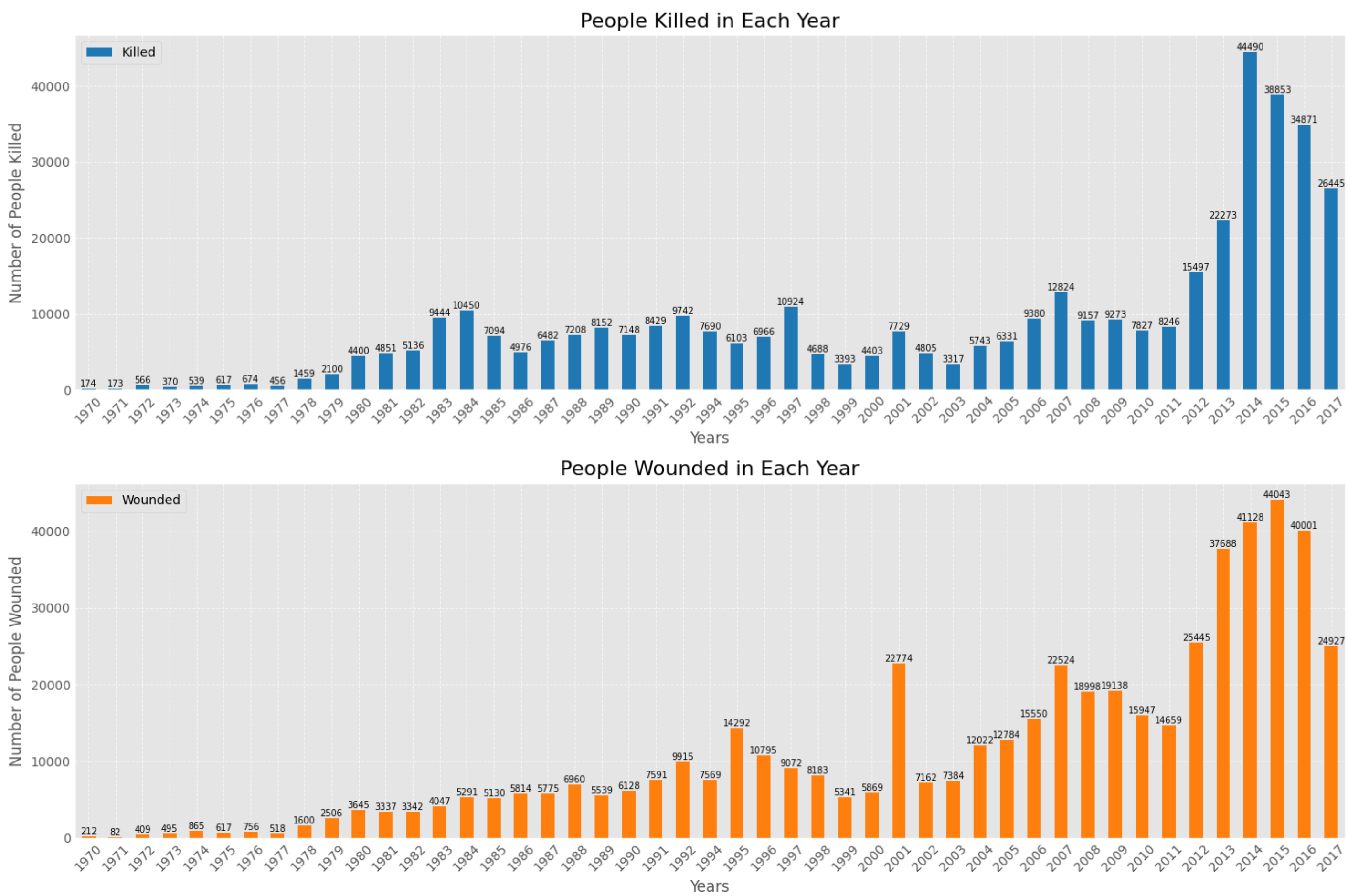
Out[49]:

	Year	Wounded	Killed
0	1970	212.0	174.0
1	1971	82.0	173.0
2	1972	409.0	566.0
3	1973	495.0	370.0
4	1974	865.0	539.0
5	1975	617.0	617.0
6	1976	756.0	674.0
7	1977	518.0	456.0
8	1978	1600.0	1459.0
9	1979	2506.0	2100.0

In [50]:

```
# Apply style
plt.style.use('ggplot')

# Create a figure and two subplots
fig, (ax0, ax1) = plt.subplots(2, 1, figsize=(15, 10))

# Plot 'Killed' data
k.plot(kind="bar", color='#1f77b4', ax=ax0, legend=True)
ax0.set_title("People Killed in Each Year", fontsize=16)
ax0.set_xlabel("Years", fontsize=12)
ax0.set_ylabel("Number of People Killed", fontsize=12)

ax0.set_xticklabels(k.index, rotation=45, fontsize=10)
ax0.grid(True, which='both', linestyle='--', linewidth=0.7, alpha=0.7)
ax0.tick_params(axis='both', which='major', labelsize=10)
ax0.legend(["Killed"], loc='upper left', bbox_to_anchor=(0, 1))

# Add data labels
for i in ax0.containers:
```

```python
    ax0.bar_label(i, label_type='edge', fontsize=7)

# Plot 'Wounded' data
w.plot(kind="bar", color='#ff7f0e', ax=ax1, legend=True)
ax1.set_title("People Wounded in Each Year", fontsize=16)
ax1.set_xlabel("Years", fontsize=12)
ax1.set_ylabel("Number of People Wounded", fontsize=12)

ax1.set_xticklabels(w.index, rotation=45, fontsize=10)
ax1.grid(True, which='both', linestyle='--', linewidth=0.7, alpha=0.7)
ax1.tick_params(axis='both', which='major', labelsize=10)
ax1.legend(["Wounded"], loc='upper left', bbox_to_anchor=(0, 1))

# Add data labels
for i in ax1.containers:
    ax1.bar_label(i, label_type='edge', fontsize=7)

# Adjust layout for better visualization
plt.tight_layout()

# Show the plot
plt.show()
```



- > Looking at these annual charts & sheet , we can say that the largest year was killing in `2014` , and then it began a gradual decline to `2017`

In [51]:

```python
# largest 10 years people killed
merged_k_w.nlargest(10,['Killed']).drop('Wounded',axis=1)
```

Out[51]:

Year	Killed

	Year	Killed
43	2014	44490.0
44	2015	38853.0
45	2016	34871.0
46	2017	26445.0
42	2013	22273.0
41	2012	15497.0
36	2007	12824.0
26	1997	10924.0
14	1984	10450.0
22	1992	9742.0

- Regarding the wounded we can say that the largest year was killing in `2015` and then it began a gradual decline to `2017`

In [52]:
```
# largest 10 years people Wounded
merged_k_w.nlargest(10,['Wounded']).drop('Killed',axis=1)
```

Out[52]:

	Year	Wounded
44	2015	44043.0
43	2014	41128.0
45	2016	40001.0
42	2013	37688.0
41	2012	25445.0
46	2017	24927.0
30	2001	22774.0
36	2007	22524.0
38	2009	19138.0
37	2008	18998.0

- success - Success of a terrorist strike

> - 0 : Represents failure to perform the operation.
> - 1: Represents success to perform the operation.

In [53]:
```
# sheet demonstrate number of terrorist attscks each countery & extent of the success and
# Aggregate the data
coun_stats = df_terr.groupby(['Region','Country', 'success']).agg({
    'Killed': 'sum',
    'Wounded': 'sum',
    'Country': 'count'  # Counts the number of occurrences for each (Country, success) pa
```

```
}).rename(columns={'Country': 'Num_Attacks'}).reset_index()

# Melt the DataFrame to combine 'Killed' and 'Wounded' into a single 'Types' column
coun_stats_melted = pd.melt(coun_stats, id_vars=['Region','Country', 'Num_Attacks', 'succe

# Display the melted DataFrame
coun_stats_melted.sort_values(by=['Region', 'Country', 'Types']).reset_index(drop=True).sc
```

Out[53]:

	Region	Country	Num_Attacks	success	Types	Counts_Types
0	Middle East & North Africa	Iraq	21861	1	Wounded	132572.0
1	Middle East & North Africa	Iraq	21861	1	Killed	73036.0
2	South Asia	Afghanistan	11141	1	Wounded	41643.0
3	South Asia	Pakistan	12600	1	Wounded	41132.0
4	South Asia	Afghanistan	11141	1	Killed	36552.0
5	South Asia	India	10280	1	Wounded	28373.0
6	South Asia	Pakistan	12600	1	Killed	23294.0
7	Sub-Saharan Africa	Nigeria	3593	1	Killed	22228.0
8	North America	United States	2340	1	Wounded	20634.0
9	South Asia	India	10280	1	Killed	19119.0
10	South Asia	Sri Lanka	2849	1	Killed	15377.0
11	South Asia	Sri Lanka	2849	1	Wounded	15193.0
12	Middle East & North Africa	Syria	2119	1	Killed	15119.0
13	South America	Colombia	7712	1	Killed	14381.0
14	Middle East & North Africa	Syria	2119	1	Wounded	14047.0
15	Southeast Asia	Philippines	5975	1	Wounded	12855.0
16	South America	Peru	5755	1	Killed	12631.0
17	Central America & Caribbean	El Salvador	5227	1	Killed	12004.0
18	Middle East & North Africa	Algeria	2561	1	Killed	11008.0
19	Middle East & North Africa	Lebanon	2182	1	Wounded	10653.0
20	Central America & Caribbean	Nicaragua	1939	1	Killed	10569.0
21	Sub-Saharan Africa	Nigeria	3593	1	Wounded	10106.0
22	South America	Colombia	7712	1	Wounded	10017.0
23	Sub-Saharan Africa	Somalia	3804	1	Killed	9824.0
24	Middle East & North Africa	Turkey	3909	1	Wounded	9702.0
25	Southeast Asia	Philippines	5975	1	Killed	9289.0
26	Middle East & North Africa	Yemen	2837	1	Wounded	8987.0
27	Middle East & North Africa	Algeria	2561	1	Wounded	8970.0
28	Sub-Saharan Africa	Somalia	3804	1	Wounded	8511.0
29	Middle East & North Africa	Yemen	2837	1	Killed	8399.0
30	South Asia	Bangladesh	1519	1	Wounded	7992.0
31	Middle East & North Africa	Israel	1683	1	Wounded	7805.0

	Region	Country	Num_Attacks	success	Types	Counts_Types
32	Southeast Asia	Thailand	3626	1	Wounded	7654.0
33	Eastern Europe	Russia	1810	1	Wounded	7289.0
34	East Asia	Japan	341	1	Wounded	6990.0
35	Middle East & North Africa	Turkey	3909	1	Killed	6705.0
36	Sub-Saharan Africa	Kenya	608	1	Wounded	6247.0
37	Western Europe	United Kingdom	4206	1	Wounded	5863.0
38	Middle East & North Africa	Iraq	2775	0	Killed	5553.0
39	Central America & Caribbean	Guatemala	1936	1	Killed	5133.0
40	Central America & Caribbean	El Salvador	5227	1	Wounded	5013.0
41	Western Europe	Spain	2818	1	Wounded	4674.0
42	Middle East & North Africa	Egypt	2011	1	Wounded	4639.0
43	Sub-Saharan Africa	South Africa	1877	1	Wounded	4458.0
44	Eastern Europe	Russia	1810	1	Killed	4192.0
45	Sub-Saharan Africa	Burundi	590	1	Killed	4181.0
46	Middle East & North Africa	Lebanon	2182	1	Killed	3996.0
47	Sub-Saharan Africa	Democratic Republic of the Congo	717	1	Killed	3991.0
48	Middle East & North Africa	Iran	594	1	Wounded	3976.0
49	South America	Peru	5755	1	Wounded	3928.0
50	Sub-Saharan Africa	Sudan	933	1	Killed	3801.0
51	North America	United States	2340	1	Killed	3758.0
52	Middle East & North Africa	Egypt	2011	1	Killed	3588.0
53	Western Europe	United Kingdom	4206	1	Killed	3300.0
54	Sub-Saharan Africa	Rwanda	154	1	Killed	3235.0
55	Middle East & North Africa	Libya	1986	1	Wounded	3162.0
56	Sub-Saharan Africa	Uganda	363	1	Killed	3057.0
57	Sub-Saharan Africa	Angola	486	1	Killed	3005.0
58	South Asia	Afghanistan	1590	0	Killed	2832.0
59	Middle East & North Africa	West Bank and Gaza Strip	1766	1	Wounded	2808.0
60	Eastern Europe	Ukraine	1529	1	Wounded	2769.0
61	Southeast Asia	Thailand	3626	1	Killed	2719.0
62	Sub-Saharan Africa	Mozambique	346	1	Killed	2689.0
63	Sub-Saharan Africa	South Africa	1877	1	Killed	2647.0
64	South Asia	Afghanistan	1590	0	Wounded	2634.0
65	Sub-Saharan Africa	South Sudan	193	1	Killed	2515.0
66	Middle East & North Africa	Libya	1986	1	Killed	2507.0
67	Western Europe	France	2481	1	Wounded	2459.0

	Region	Country	Num_Attacks	success	Types	Counts_Types
68	Sub-Saharan Africa	Angola	486	1	Wounded	2432.0
69	Sub-Saharan Africa	Burundi	590	1	Wounded	2432.0
70	Southeast Asia	Indonesia	666	1	Wounded	2428.0
71	Sub-Saharan Africa	Cameroon	309	1	Killed	2298.0
72	Eastern Europe	Ukraine	1529	1	Killed	2255.0
73	Sub-Saharan Africa	Sudan	933	1	Wounded	2147.0
74	Middle East & North Africa	Iraq	2775	0	Wounded	2118.0
75	South Asia	Nepal	956	1	Wounded	2098.0
76	Sub-Saharan Africa	Central African Republic	263	1	Killed	1983.0
77	South Asia	Nepal	956	1	Killed	1965.0
78	Sub-Saharan Africa	Kenya	608	1	Killed	1921.0
79	East Asia	China	220	1	Wounded	1826.0
80	Sub-Saharan Africa	Ethiopia	181	1	Killed	1748.0
81	Central America & Caribbean	Nicaragua	1939	1	Wounded	1706.0
82	Sub-Saharan Africa	Chad	87	1	Wounded	1681.0
83	Middle East & North Africa	Israel	1683	1	Killed	1664.0
84	Middle East & North Africa	Iran	594	1	Killed	1647.0
85	Middle East & North Africa	Saudi Arabia	319	1	Wounded	1631.0
86	Southeast Asia	Myanmar	509	1	Wounded	1624.0
87	Sub-Saharan Africa	Mozambique	346	1	Wounded	1514.0
88	Sub-Saharan Africa	Mali	523	1	Killed	1412.0
89	Sub-Saharan Africa	Democratic Republic of the Congo	717	1	Wounded	1365.0
90	Sub-Saharan Africa	Niger	146	1	Killed	1356.0
91	Sub-Saharan Africa	Mali	523	1	Wounded	1351.0
92	Middle East & North Africa	West Bank and Gaza Strip	1766	1	Killed	1322.0
93	Sub-Saharan Africa	South Sudan	193	1	Wounded	1311.0
94	Southeast Asia	Myanmar	509	1	Killed	1260.0
95	Western Europe	Spain	2818	1	Killed	1254.0
96	Western Europe	Italy	1392	1	Wounded	1235.0
97	South Asia	Bangladesh	1519	1	Killed	1227.0
98	Southeast Asia	Indonesia	666	1	Killed	1227.0
99	Sub-Saharan Africa	Ethiopia	181	1	Wounded	1212.0
100	Central America & Caribbean	Guatemala	1936	1	Wounded	1178.0
101	Sub-Saharan Africa	Uganda	363	1	Wounded	1111.0
102	Central Asia	Tajikistan	180	1	Wounded	1103.0
103	Sub-Saharan Africa	Chad	87	1	Killed	1102.0

	Region	Country	Num_Attacks	success	Types	Counts_Types
104	Sub-Saharan Africa	Cameroon	309	1	Wounded	1063.0
105	East Asia	China	220	1	Killed	1002.0
106	Sub-Saharan Africa	Central African Republic	263	1	Wounded	931.0
107	Sub-Saharan Africa	Rwanda	154	1	Wounded	921.0
108	South Asia	Pakistan	1768	0	Wounded	906.0
109	Western Europe	West Germany (FRG)	465	1	Wounded	847.0
110	Sub-Saharan Africa	Sierra Leone	98	1	Killed	833.0
111	Southeast Asia	Cambodia	243	1	Wounded	782.0
112	North America	Mexico	479	1	Killed	759.0
113	South America	Argentina	714	1	Wounded	717.0
114	Western Europe	Greece	1126	1	Wounded	707.0
115	South America	Chile	2221	1	Wounded	697.0
116	Western Europe	Germany	659	1	Wounded	665.0
117	North America	Mexico	479	1	Wounded	660.0
118	Middle East & North Africa	Saudi Arabia	319	1	Killed	631.0
119	South Asia	India	1680	0	Wounded	607.0
120	Southeast Asia	Cambodia	243	1	Killed	542.0
121	South Asia	Pakistan	1768	0	Killed	528.0
122	Western Europe	France	2481	1	Killed	513.0
123	Southeast Asia	Philippines	933	0	Wounded	512.0
124	Western Europe	Belgium	125	1	Wounded	512.0
125	South America	Argentina	714	1	Killed	484.0
126	Sub-Saharan Africa	Niger	146	1	Wounded	467.0
127	Sub-Saharan Africa	Nigeria	314	0	Killed	454.0
128	Sub-Saharan Africa	Somalia	338	0	Killed	449.0
129	Middle East & North Africa	Tunisia	97	1	Wounded	431.0
130	Western Europe	Italy	1392	1	Killed	407.0
131	Sub-Saharan Africa	Namibia	147	1	Wounded	405.0
132	Middle East & North Africa	Yemen	510	0	Killed	377.0
133	Central Asia	Georgia	192	1	Wounded	372.0
134	South Asia	Sri Lanka	173	0	Wounded	368.0
135	North America	Canada	75	1	Killed	365.0
136	Sub-Saharan Africa	Somalia	338	0	Wounded	364.0
137	Eastern Europe	Kosovo	172	1	Wounded	357.0
138	Middle East & North Africa	Yemen	510	0	Wounded	341.0
139	Middle East & North Africa	Tunisia	97	1	Killed	337.0

	Region	Country	Num_Attacks	success	Types	Counts_Types
140	Central America & Caribbean	Haiti	185	1	Killed	335.0
141	Sub-Saharan Africa	Senegal	116	1	Killed	325.0
142	Sub-Saharan Africa	Zaire	45	1	Killed	324.0
143	Sub-Saharan Africa	Senegal	116	1	Wounded	322.0
144	Western Europe	Greece	1126	1	Killed	321.0
145	South America	Colombia	594	0	Killed	317.0
146	South America	Colombia	594	0	Wounded	311.0
147	Central Asia	Tajikistan	180	1	Killed	305.0
148	Central America & Caribbean	Honduras	286	1	Killed	304.0
149	Middle East & North Africa	Morocco	33	1	Killed	292.0
150	Middle East & North Africa	Kuwait	63	1	Wounded	291.0
151	Central America & Caribbean	Haiti	185	1	Wounded	282.0
152	Middle East & North Africa	Egypt	468	0	Killed	281.0
153	Eastern Europe	Yugoslavia	179	1	Wounded	274.0
154	Sub-Saharan Africa	Djibouti	22	1	Killed	274.0
155	Central Asia	Georgia	192	1	Killed	272.0
156	Southeast Asia	Philippines	933	0	Killed	270.0
157	Western Europe	Spain	431	0	Wounded	261.0
158	Central Asia	Azerbaijan	44	1	Killed	257.0
159	Sub-Saharan Africa	Ivory Coast	67	1	Killed	253.0
160	Middle East & North Africa	Jordan	84	1	Wounded	251.0
161	Middle East & North Africa	Lebanon	296	0	Wounded	251.0
162	Eastern Europe	Croatia	55	1	Killed	248.0
163	Western Europe	United Kingdom	1029	0	Wounded	243.0
164	South Asia	Bangladesh	129	0	Wounded	233.0
165	South America	Venezuela	246	1	Wounded	232.0
166	Sub-Saharan Africa	Tanzania	50	1	Wounded	232.0
167	South Asia	India	1680	0	Killed	222.0
168	South America	Venezuela	246	1	Killed	222.0
169	Central America & Caribbean	Honduras	286	1	Wounded	221.0
170	Sub-Saharan Africa	Zimbabwe	96	1	Wounded	220.0
171	Sub-Saharan Africa	Namibia	147	1	Killed	220.0
172	Sub-Saharan Africa	Burkina Faso	45	1	Wounded	218.0
173	Sub-Saharan Africa	Rhodesia	80	1	Killed	217.0
174	Sub-Saharan Africa	Guinea	21	1	Killed	213.0
175	Eastern Europe	Belarus	13	1	Wounded	212.0

	Region	Country	Num_Attacks	success	Types	Counts_Types
176	Sub-Saharan Africa	Zaire	45	1	Wounded	211.0
177	South America	Chile	2221	1	Killed	207.0
178	Middle East & North Africa	West Bank and Gaza Strip	461	0	Wounded	206.0
179	South America	Brazil	237	1	Killed	201.0
180	Central Asia	Uzbekistan	19	1	Wounded	200.0
181	Middle East & North Africa	Morocco	33	1	Wounded	199.0
182	Middle East & North Africa	Turkey	383	0	Wounded	197.0
183	Middle East & North Africa	Bahrain	179	1	Wounded	188.0
184	Middle East & North Africa	Egypt	468	0	Wounded	183.0
185	Middle East & North Africa	Turkey	383	0	Killed	183.0
186	Sub-Saharan Africa	Republic of the Congo	35	1	Killed	182.0
187	Middle East & North Africa	Algeria	182	0	Wounded	180.0
188	Central Asia	Azerbaijan	44	1	Wounded	180.0
189	Middle East & North Africa	West Bank and Gaza Strip	461	0	Killed	178.0
190	Sub-Saharan Africa	Liberia	32	1	Killed	177.0
191	Sub-Saharan Africa	Ivory Coast	67	1	Wounded	171.0
192	Sub-Saharan Africa	Madagascar	23	1	Wounded	169.0
193	Southeast Asia	Thailand	223	0	Wounded	164.0
194	Sub-Saharan Africa	Djibouti	22	1	Wounded	162.0
195	South America	Bolivia	273	1	Wounded	160.0
196	Sub-Saharan Africa	Rhodesia	80	1	Wounded	158.0
197	South Asia	Sri Lanka	173	0	Killed	153.0
198	Sub-Saharan Africa	Zimbabwe	96	1	Killed	152.0
199	Southeast Asia	Malaysia	86	1	Killed	152.0
200	Eastern Europe	Russia	384	0	Wounded	152.0
201	South America	Peru	341	0	Wounded	150.0
202	Eastern Europe	Soviet Union	67	1	Wounded	149.0
203	South America	Brazil	237	1	Wounded	148.0
204	Eastern Europe	Bosnia-Herzegovina	151	1	Wounded	148.0
205	Middle East & North Africa	Libya	263	0	Wounded	148.0
206	Middle East & North Africa	Israel	500	0	Wounded	141.0
207	South America	Peru	341	0	Killed	140.0
208	North America	Canada	75	1	Wounded	139.0
209	Sub-Saharan Africa	Burkina Faso	45	1	Killed	133.0
210	Sub-Saharan Africa	Nigeria	314	0	Wounded	133.0
211	East Asia	South Korea	34	1	Wounded	133.0

	Region	Country	Num_Attacks	success	Types	Counts_Types
212	Middle East & North Africa	Jordan	84	1	Killed	131.0
213	Middle East & North Africa	United Arab Emirates	17	1	Killed	123.0
214	Central America & Caribbean	Dominican Republic	85	1	Wounded	123.0
215	Western Europe	Austria	88	1	Wounded	122.0
216	Sub-Saharan Africa	Sierra Leone	98	1	Wounded	122.0
217	Sub-Saharan Africa	South Sudan	32	0	Killed	119.0
218	Sub-Saharan Africa	Niger	8	0	Killed	118.0
219	South Asia	Maldives	18	1	Wounded	118.0
220	Eastern Europe	Russia	384	0	Killed	116.0
221	Western Europe	Ireland	139	1	Killed	115.0
222	Eastern Europe	Albania	64	1	Wounded	115.0
223	Eastern Europe	Yugoslavia	179	1	Killed	114.0
224	Australasia & Oceania	Australia	97	1	Wounded	112.0
225	Middle East & North Africa	Syria	82	0	Killed	110.0
226	Western Europe	United Kingdom	1029	0	Killed	110.0
227	Southeast Asia	Malaysia	86	1	Wounded	100.0
228	Western Europe	West Germany (FRG)	465	1	Killed	93.0
229	Eastern Europe	Soviet Union	67	1	Killed	93.0
230	Middle East & North Africa	Libya	263	0	Killed	91.0
231	Eastern Europe	Moldova	18	1	Wounded	88.0
232	Western Europe	Norway	16	1	Wounded	87.0
233	Australasia & Oceania	Papua New Guinea	77	1	Wounded	87.0
234	Sub-Saharan Africa	South Africa	139	0	Wounded	87.0
235	Western Europe	Portugal	129	1	Wounded	86.0
236	Western Europe	Germany	659	1	Killed	84.0
237	Eastern Europe	Kosovo	172	1	Killed	83.0
238	Central America & Caribbean	Panama	110	1	Wounded	82.0
239	Sub-Saharan Africa	Sudan	34	0	Killed	82.0
240	Southeast Asia	South Vietnam	1	1	Killed	81.0
241	Western Europe	Switzerland	90	1	Wounded	81.0
242	Eastern Europe	Bosnia-Herzegovina	151	1	Killed	79.0
243	Western Europe	Norway	16	1	Killed	79.0
244	Western Europe	Belgium	125	1	Killed	79.0
245	East Asia	Taiwan	37	1	Wounded	78.0
246	Sub-Saharan Africa	Democratic Republic of the Congo	58	0	Killed	78.0
247	Western Europe	Sweden	113	1	Wounded	77.0

	Region	Country	Num_Attacks	success	Types	Counts_Types
248	Central America & Caribbean	Barbados	3	1	Killed	76.0
249	East Asia	Hong Kong	20	1	Wounded	75.0
250	South America	Ecuador	203	1	Wounded	74.0
251	Australasia & Oceania	Papua New Guinea	77	1	Killed	74.0
252	Eastern Europe	Croatia	55	1	Wounded	73.0
253	Sub-Saharan Africa	Tanzania	50	1	Killed	73.0
254	Southeast Asia	Laos	24	1	Wounded	73.0
255	Western Europe	Switzerland	90	1	Killed	73.0
256	Eastern Europe	Ukraine	180	0	Wounded	72.0
257	Sub-Saharan Africa	Zambia	58	1	Killed	70.0
258	Central Asia	Armenia	20	1	Wounded	70.0
259	North America	United States	496	0	Wounded	68.0
260	Central Asia	Uzbekistan	19	1	Killed	67.0
261	East Asia	Japan	341	1	Killed	66.0
262	Middle East & North Africa	Lebanon	296	0	Killed	65.0
263	South America	Paraguay	101	1	Wounded	63.0
264	Middle East & North Africa	Syria	82	0	Wounded	62.0
265	Sub-Saharan Africa	Eritrea	10	1	Wounded	62.0
266	Middle East & North Africa	Kuwait	63	1	Killed	61.0
267	Sub-Saharan Africa	Zambia	58	1	Wounded	61.0
268	East Asia	Taiwan	37	1	Killed	60.0
269	Sub-Saharan Africa	Togo	42	1	Killed	60.0
270	South America	Paraguay	101	1	Killed	59.0
271	Middle East & North Africa	Algeria	182	0	Killed	58.0
272	South America	Chile	144	0	Wounded	58.0
273	Western Europe	France	212	0	Wounded	57.0
274	Western Europe	Italy	173	0	Wounded	56.0
275	Sub-Saharan Africa	Guinea	21	1	Wounded	56.0
276	Sub-Saharan Africa	Republic of the Congo	35	1	Wounded	54.0
277	Eastern Europe	Macedonia	107	1	Wounded	53.0
278	South America	Ecuador	203	1	Killed	53.0
279	Central America & Caribbean	Guatemala	114	0	Wounded	53.0
280	South Asia	Nepal	259	0	Wounded	53.0
281	Middle East & North Africa	Iran	90	0	Wounded	53.0
282	Sub-Saharan Africa	Cameroon	23	0	Killed	49.0
283	Central America & Caribbean	El Salvador	93	0	Killed	49.0

	Region	Country	Num_Attacks	success	Types	Counts_Types
284	Central America & Caribbean	El Salvador	93	0	Wounded	49.0
285	Eastern Europe	Macedonia	107	1	Killed	48.0
286	Sub-Saharan Africa	Eritrea	10	1	Killed	46.0
287	Sub-Saharan Africa	Lesotho	25	1	Killed	45.0
288	Middle East & North Africa	Bahrain	179	1	Killed	44.0
289	East Asia	Macau	27	1	Wounded	44.0
290	Western Europe	Netherlands	107	1	Wounded	44.0
291	Central America & Caribbean	Guadeloupe	47	1	Wounded	43.0
292	Western Europe	Cyprus	112	1	Killed	42.0
293	Eastern Europe	Albania	64	1	Killed	42.0
294	Sub-Saharan Africa	Mauritania	14	1	Killed	42.0
295	Central America & Caribbean	Jamaica	32	1	Killed	41.0
296	Middle East & North Africa	Saudi Arabia	52	0	Killed	41.0
297	South America	Guyana	19	1	Killed	40.0
298	South America	Bolivia	273	1	Killed	40.0
299	Middle East & North Africa	Israel	500	0	Killed	39.0
300	South America	Argentina	101	0	Wounded	38.0
301	Sub-Saharan Africa	Angola	13	0	Killed	38.0
302	Central Asia	Kazakhstan	22	1	Killed	37.0
303	Central America & Caribbean	Panama	110	1	Killed	37.0
304	Central Asia	Armenia	20	1	Killed	37.0
305	Sub-Saharan Africa	Liberia	32	1	Wounded	36.0
306	Eastern Europe	Latvia	13	1	Wounded	36.0
307	Eastern Europe	East Germany (GDR)	35	1	Wounded	36.0
308	Western Europe	Cyprus	112	1	Wounded	36.0
309	Eastern Europe	Bulgaria	46	1	Wounded	36.0
310	Australasia & Oceania	New Caledonia	28	1	Killed	35.0
311	Middle East & North Africa	Saudi Arabia	52	0	Wounded	35.0
312	Central America & Caribbean	Costa Rica	56	1	Wounded	34.0
313	Western Europe	Spain	431	0	Killed	34.0
314	Central America & Caribbean	Dominican Republic	85	1	Killed	34.0
315	Central America & Caribbean	Trinidad and Tobago	20	1	Wounded	34.0
316	Central America & Caribbean	Guatemala	114	0	Killed	34.0
317	Sub-Saharan Africa	Uganda	31	0	Wounded	34.0
318	Sub-Saharan Africa	Malawi	5	1	Killed	33.0
319	Central America & Caribbean	Jamaica	32	1	Wounded	33.0

	Region	Country	Num_Attacks	success	Types	Counts_Types
320	Sub-Saharan Africa	Togo	42	1	Wounded	32.0
321	Western Europe	Portugal	129	1	Killed	31.0
322	Eastern Europe	Poland	34	1	Wounded	30.0
323	Eastern Europe	Czech Republic	23	1	Wounded	29.0
324	Sub-Saharan Africa	Lesotho	25	1	Wounded	29.0
325	Western Europe	Ireland	139	1	Wounded	29.0
326	Central America & Caribbean	Nicaragua	31	0	Killed	29.0
327	Western Europe	Netherlands	107	1	Killed	29.0
328	Western Europe	Denmark	35	1	Wounded	28.0
329	Sub-Saharan Africa	Mauritania	14	1	Wounded	28.0
330	Western Europe	Austria	88	1	Killed	28.0
331	East Asia	Hong Kong	6	0	Wounded	27.0
332	Western Europe	Finland	19	1	Wounded	27.0
333	Southeast Asia	Vietnam	7	1	Wounded	27.0
334	Southeast Asia	Laos	24	1	Killed	27.0
335	Sub-Saharan Africa	Kenya	75	0	Killed	27.0
336	Sub-Saharan Africa	South Africa	139	0	Killed	27.0
337	Sub-Saharan Africa	Guinea-Bissau	8	1	Wounded	27.0
338	Eastern Europe	Czechoslovakia	7	1	Killed	27.0
339	Western Europe	Greece	149	0	Wounded	26.0
340	South America	Suriname	61	1	Killed	26.0
341	Middle East & North Africa	Iran	90	0	Killed	26.0
342	Eastern Europe	Bulgaria	46	1	Killed	26.0
343	Central Asia	Kyrgyzstan	27	1	Wounded	25.0
344	Middle East & North Africa	United Arab Emirates	17	1	Wounded	25.0
345	Central America & Caribbean	Nicaragua	31	0	Wounded	25.0
346	South America	Suriname	61	1	Wounded	24.0
347	Sub-Saharan Africa	Burundi	23	0	Killed	24.0
348	Sub-Saharan Africa	Madagascar	23	1	Killed	23.0
349	North America	Mexico	45	0	Wounded	23.0
350	Sub-Saharan Africa	Angola	13	0	Wounded	23.0
351	Southeast Asia	Thailand	223	0	Killed	23.0
352	Sub-Saharan Africa	Mozambique	17	0	Killed	22.0
353	Australasia & Oceania	Australia	97	1	Killed	22.0
354	Sub-Saharan Africa	Burundi	23	0	Wounded	22.0
355	South America	Guyana	19	1	Wounded	22.0

	Region	Country	Num_Attacks	success	Types	Counts_Types
356	Sub-Saharan Africa	Sudan	34	0	Wounded	22.0
357	Eastern Europe	Czechoslovakia	7	1	Wounded	21.0
358	North America	Mexico	45	0	Killed	21.0
359	South America	Chile	144	0	Killed	21.0
360	Australasia & Oceania	New Caledonia	28	1	Wounded	21.0
361	Western Europe	Sweden	113	1	Killed	21.0
362	Western Europe	France	212	0	Killed	21.0
363	South Asia	Maldives	18	1	Killed	20.0
364	Central America & Caribbean	Grenada	3	1	Wounded	20.0
365	Sub-Saharan Africa	Mali	43	0	Killed	20.0
366	Southeast Asia	Myanmar	37	0	Killed	20.0
367	Central Asia	Georgia	25	0	Wounded	20.0
368	Central Asia	Kazakhstan	22	1	Wounded	20.0
369	Central America & Caribbean	Honduras	37	0	Wounded	19.0
370	Sub-Saharan Africa	Ghana	15	1	Killed	19.0
371	Australasia & Oceania	Fiji	16	1	Wounded	18.0
372	Central America & Caribbean	Haiti	28	0	Wounded	18.0
373	Western Europe	Germany	76	0	Wounded	18.0
374	Southeast Asia	Indonesia	95	0	Wounded	17.0
375	South Asia	Bangladesh	129	0	Killed	17.0
376	Sub-Saharan Africa	Guinea-Bissau	8	1	Killed	17.0
377	Sub-Saharan Africa	Chad	4	0	Killed	17.0
378	Sub-Saharan Africa	Ethiopia	9	0	Killed	17.0
379	East Asia	China	32	0	Wounded	16.0
380	Sub-Saharan Africa	Madagascar	4	0	Wounded	16.0
381	Sub-Saharan Africa	Kenya	75	0	Wounded	16.0
382	Eastern Europe	Hungary	40	1	Wounded	16.0
383	Sub-Saharan Africa	Togo	6	0	Killed	16.0
384	South America	Venezuela	47	0	Wounded	15.0
385	Sub-Saharan Africa	Ivory Coast	7	0	Killed	15.0
386	Western Europe	West Germany (FRG)	76	0	Wounded	15.0
387	Sub-Saharan Africa	People's Republic of the Congo	3	1	Killed	15.0
388	Eastern Europe	Belarus	13	1	Killed	14.0
389	Western Europe	Netherlands	23	0	Wounded	14.0
390	Middle East & North Africa	Tunisia	12	0	Killed	14.0
391	Middle East & North Africa	Qatar	6	1	Wounded	13.0

	Region	Country	Num_Attacks	success	Types	Counts_Types
392	Australasia & Oceania	French Polynesia	3	1	Wounded	13.0
393	Sub-Saharan Africa	Gambia	2	0	Killed	13.0
394	Eastern Europe	Moldova	18	1	Killed	13.0
395	Western Europe	Switzerland	21	0	Wounded	13.0
396	North America	United States	496	0	Killed	13.0
397	Western Europe	Italy	173	0	Killed	13.0
398	South America	French Guiana	6	1	Wounded	13.0
399	Central America & Caribbean	Costa Rica	56	1	Killed	12.0
400	South America	Brazil	36	0	Wounded	12.0
401	Western Europe	Malta	22	1	Wounded	12.0
402	Middle East & North Africa	International	1	1	Wounded	12.0
403	Central America & Caribbean	St. Lucia	1	1	Wounded	12.0
404	Sub-Saharan Africa	South Sudan	32	0	Wounded	12.0
405	Sub-Saharan Africa	Botswana	10	1	Wounded	11.0
406	Sub-Saharan Africa	Botswana	10	1	Killed	11.0
407	Southeast Asia	Indonesia	95	0	Killed	11.0
408	Eastern Europe	Albania	16	0	Wounded	11.0
409	Western Europe	Finland	19	1	Killed	11.0
410	Sub-Saharan Africa	Central African Republic	20	0	Wounded	11.0
411	Eastern Europe	Slovak Republic	15	1	Wounded	11.0
412	Eastern Europe	Estonia	16	1	Wounded	11.0
413	Central Asia	Kyrgyzstan	27	1	Killed	10.0
414	Central America & Caribbean	Dominica	2	1	Wounded	10.0
415	East Asia	South Korea	34	1	Killed	10.0
416	Middle East & North Africa	Jordan	29	0	Wounded	9.0
417	Middle East & North Africa	Kuwait	13	0	Wounded	9.0
418	East Asia	Taiwan	13	0	Wounded	9.0
419	South Asia	Bhutan	6	1	Killed	9.0
420	Central America & Caribbean	St. Kitts and Nevis	2	1	Wounded	9.0
421	Eastern Europe	Serbia	11	1	Wounded	9.0
422	Sub-Saharan Africa	Ghana	15	1	Wounded	9.0
423	Eastern Europe	Poland	34	1	Killed	9.0
424	Western Europe	Portugal	11	0	Wounded	9.0
425	Central America & Caribbean	Guadeloupe	47	1	Killed	8.0
426	Australasia & Oceania	Fiji	16	1	Killed	8.0
427	Western Europe	Netherlands	23	0	Killed	8.0

	Region	Country	Num_Attacks	success	Types	Counts_Types
428	Sub-Saharan Africa	Madagascar	4	0	Killed	8.0
429	Sub-Saharan Africa	Uganda	31	0	Killed	8.0
430	Central America & Caribbean	Grenada	3	1	Killed	8.0
431	Sub-Saharan Africa	Benin	8	1	Wounded	8.0
432	Eastern Europe	Kosovo	24	0	Wounded	8.0
433	East Asia	Japan	61	0	Wounded	8.0
434	Middle East & North Africa	Qatar	6	1	Killed	7.0
435	Southeast Asia	East Timor	7	1	Killed	7.0
436	North America	Canada	21	0	Wounded	7.0
437	Sub-Saharan Africa	Central African Republic	20	0	Killed	7.0
438	Eastern Europe	Yugoslavia	24	0	Wounded	7.0
439	Southeast Asia	Myanmar	37	0	Wounded	7.0
440	Central Asia	Azerbaijan	5	0	Wounded	7.0
441	Central America & Caribbean	Cuba	21	1	Wounded	7.0
442	Sub-Saharan Africa	Sierra Leone	3	0	Killed	7.0
443	Eastern Europe	Macedonia	11	0	Wounded	7.0
444	South America	Bolivia	41	0	Wounded	6.0
445	Sub-Saharan Africa	Swaziland	15	1	Killed	6.0
446	Central Asia	Tajikistan	8	0	Wounded	6.0
447	South America	Argentina	101	0	Killed	6.0
448	South America	Uruguay	71	1	Wounded	6.0
449	Eastern Europe	Ukraine	180	0	Killed	6.0
450	South America	Uruguay	71	1	Killed	6.0
451	South America	Paraguay	13	0	Wounded	6.0
452	Central Asia	Georgia	25	0	Killed	6.0
453	Sub-Saharan Africa	Gabon	6	1	Killed	6.0
454	Western Europe	Luxembourg	14	1	Wounded	6.0
455	Eastern Europe	Czech Republic	23	1	Killed	6.0
456	Eastern Europe	Slovak Republic	15	1	Killed	6.0
457	Southeast Asia	East Timor	7	1	Wounded	6.0
458	Eastern Europe	Hungary	40	1	Killed	6.0
459	Eastern Europe	Romania	4	1	Wounded	6.0
460	Central America & Caribbean	Trinidad and Tobago	20	1	Killed	6.0
461	East Asia	China	32	0	Killed	6.0
462	Sub-Saharan Africa	Niger	8	0	Wounded	5.0
463	Sub-Saharan Africa	Mali	43	0	Wounded	5.0

	Region	Country	Num_Attacks	success	Types	Counts_Types
464	South America	Ecuador	17	0	Wounded	5.0
465	Australasia & Oceania	Papua New Guinea	12	0	Killed	5.0
466	South America	Venezuela	47	0	Killed	5.0
467	Eastern Europe	Yugoslavia	24	0	Killed	5.0
468	Southeast Asia	Singapore	7	1	Killed	5.0
469	Eastern Europe	Serbia-Montenegro	10	1	Wounded	5.0
470	Western Europe	Denmark	35	1	Killed	5.0
471	Central America & Caribbean	Costa Rica	11	0	Killed	5.0
472	Western Europe	Malta	22	1	Killed	5.0
473	South Asia	Bhutan	6	1	Wounded	5.0
474	Western Europe	Cyprus	20	0	Wounded	5.0
475	Sub-Saharan Africa	Lesotho	4	0	Wounded	5.0
476	Western Europe	West Germany (FRG)	76	0	Killed	4.0
477	South Asia	Maldives	4	0	Wounded	4.0
478	Western Europe	Greece	149	0	Killed	4.0
479	Western Europe	Belgium	29	0	Wounded	4.0
480	Southeast Asia	Cambodia	16	0	Wounded	4.0
481	South Asia	Nepal	259	0	Killed	4.0
482	Western Europe	Austria	27	0	Wounded	4.0
483	Sub-Saharan Africa	Ghana	4	0	Wounded	4.0
484	Australasia & Oceania	Papua New Guinea	12	0	Wounded	4.0
485	Central America & Caribbean	Costa Rica	11	0	Wounded	4.0
486	East Asia	North Korea	1	1	Wounded	4.0
487	Middle East & North Africa	Western Sahara	5	1	Wounded	4.0
488	Australasia & Oceania	Solomon Islands	4	1	Killed	4.0
489	Central America & Caribbean	Cuba	9	0	Killed	4.0
490	Central America & Caribbean	Cuba	21	1	Killed	4.0
491	Sub-Saharan Africa	Mozambique	17	0	Wounded	4.0
492	Sub-Saharan Africa	Democratic Republic of the Congo	58	0	Wounded	4.0
493	Sub-Saharan Africa	Equatorial Guinea	2	1	Wounded	3.0
494	Western Europe	Vatican City	1	0	Wounded	3.0
495	Sub-Saharan Africa	Gabon	2	0	Wounded	3.0
496	Central Asia	Turkmenistan	1	1	Killed	3.0
497	Eastern Europe	Estonia	16	1	Killed	3.0
498	Eastern Europe	Serbia-Montenegro	10	1	Killed	3.0
499	Southeast Asia	Singapore	7	1	Wounded	3.0

	Region	Country	Num_Attacks	success	Types	Counts_Types
500	Western Europe	Cyprus	20	0	Killed	3.0
501	Sub-Saharan Africa	Swaziland	15	1	Wounded	3.0
502	Central America & Caribbean	Cuba	9	0	Wounded	3.0
503	Central America & Caribbean	Dominica	2	1	Killed	3.0
504	Central America & Caribbean	Barbados	3	1	Wounded	3.0
505	Western Europe	Sweden	19	0	Wounded	3.0
506	Eastern Europe	Romania	2	0	Wounded	3.0
507	South America	Suriname	5	0	Killed	3.0
508	Central America & Caribbean	Jamaica	4	0	Wounded	3.0
509	Eastern Europe	Romania	4	1	Killed	3.0
510	Eastern Europe	Serbia	11	1	Killed	3.0
511	East Asia	North Korea	1	1	Killed	3.0
512	Central America & Caribbean	Belize	7	1	Killed	3.0
513	Central America & Caribbean	Honduras	37	0	Killed	3.0
514	Eastern Europe	Soviet Union	11	0	Killed	3.0
515	East Asia	Macau	6	0	Wounded	2.0
516	Sub-Saharan Africa	Togo	6	0	Wounded	2.0
517	Sub-Saharan Africa	Tanzania	9	0	Wounded	2.0
518	South America	Guyana	7	0	Wounded	2.0
519	Sub-Saharan Africa	Gambia	1	1	Wounded	2.0
520	Western Europe	Ireland	168	0	Wounded	2.0
521	Eastern Europe	Slovenia	6	1	Wounded	2.0
522	South America	Brazil	36	0	Killed	2.0
523	South America	Bolivia	41	0	Killed	2.0
524	Central America & Caribbean	St. Lucia	1	1	Killed	2.0
525	Central Asia	Kazakhstan	5	0	Killed	2.0
526	Middle East & North Africa	United Arab Emirates	5	0	Wounded	2.0
527	Central America & Caribbean	Guadeloupe	9	0	Wounded	2.0
528	Middle East & North Africa	Tunisia	12	0	Wounded	2.0
529	Middle East & North Africa	Jordan	29	0	Killed	2.0
530	Middle East & North Africa	Kuwait	13	0	Killed	2.0
531	Middle East & North Africa	South Yemen	2	1	Wounded	2.0
532	Sub-Saharan Africa	Zimbabwe	5	0	Killed	2.0
533	Central Asia	Tajikistan	8	0	Killed	2.0
534	Eastern Europe	East Germany (GDR)	35	1	Killed	2.0
535	Central Asia	Turkmenistan	1	1	Wounded	2.0

	Region	Country	Num_Attacks	success	Types	Counts_Types
536	East Asia	Hong Kong	20	1	Killed	2.0
537	Australasia & Oceania	New Zealand	11	1	Wounded	2.0
538	East Asia	Hong Kong	6	0	Killed	2.0
539	Western Europe	Ireland	168	0	Killed	2.0
540	Eastern Europe	Bosnia-Herzegovina	8	0	Wounded	2.0
541	Australasia & Oceania	New Caledonia	3	0	Wounded	2.0
542	Eastern Europe	Bulgaria	6	0	Killed	2.0
543	Southeast Asia	East Timor	3	0	Killed	2.0
544	Sub-Saharan Africa	Comoros	3	1	Wounded	2.0
545	Eastern Europe	Czechoslovakia	3	0	Wounded	2.0
546	Middle East & North Africa	North Yemen	4	1	Killed	2.0
547	Sub-Saharan Africa	Equatorial Guinea	2	1	Killed	2.0
548	Western Europe	Austria	27	0	Killed	2.0
549	Sub-Saharan Africa	Ethiopia	9	0	Wounded	2.0
550	Eastern Europe	Latvia	13	1	Killed	2.0
551	Sub-Saharan Africa	Zimbabwe	5	0	Wounded	2.0
552	Western Europe	Portugal	11	0	Killed	1.0
553	Sub-Saharan Africa	Rwanda	5	0	Wounded	1.0
554	Sub-Saharan Africa	Rwanda	5	0	Killed	1.0
555	Western Europe	Norway	3	0	Wounded	1.0
556	Western Europe	Sweden	19	0	Killed	1.0
557	Western Europe	Denmark	6	0	Wounded	1.0
558	Sub-Saharan Africa	Namibia	4	0	Wounded	1.0
559	Western Europe	Switzerland	21	0	Killed	1.0
560	Sub-Saharan Africa	Senegal	2	0	Wounded	1.0
561	Sub-Saharan Africa	Zambia	4	0	Wounded	1.0
562	Sub-Saharan Africa	Mauritania	4	0	Killed	1.0
563	Australasia & Oceania	Australia	17	0	Killed	1.0
564	South America	Ecuador	17	0	Killed	1.0
565	Eastern Europe	Slovenia	6	1	Killed	1.0
566	Eastern Europe	Romania	2	0	Killed	1.0
567	Southeast Asia	Cambodia	16	0	Killed	1.0
568	Southeast Asia	Brunei	6	0	Wounded	1.0
569	Central America & Caribbean	Dominican Republic	5	0	Wounded	1.0
570	Central America & Caribbean	Grenada	2	0	Killed	1.0
571	South Asia	Mauritius	2	0	Wounded	1.0

	Region	Country	Num_Attacks	success	Types	Counts_Types
572	Central America & Caribbean	Grenada	2	0	Wounded	1.0
573	Central Asia	Uzbekistan	2	0	Killed	1.0
574	Central America & Caribbean	Haiti	28	0	Killed	1.0
575	Eastern Europe	Slovak Republic	3	0	Killed	1.0
576	Central America & Caribbean	Jamaica	4	0	Killed	1.0
577	Central Asia	Turkmenistan	1	0	Wounded	1.0
578	South America	Guyana	7	0	Killed	1.0
579	Central Asia	Kyrgyzstan	8	0	Wounded	1.0
580	Southeast Asia	East Timor	3	0	Wounded	1.0
581	Eastern Europe	Soviet Union	11	0	Wounded	1.0
582	South America	French Guiana	6	1	Killed	1.0
583	Central America & Caribbean	Martinique	12	1	Wounded	1.0
584	Central America & Caribbean	Panama	17	0	Killed	1.0
585	Central America & Caribbean	Panama	17	0	Wounded	1.0
586	Central Asia	Kazakhstan	5	0	Wounded	1.0
587	Middle East & North Africa	Western Sahara	5	1	Killed	1.0
588	Central America & Caribbean	Trinidad and Tobago	2	0	Wounded	1.0
589	Middle East & North Africa	Bahrain	28	0	Wounded	1.0
590	Central Asia	Armenia	4	0	Wounded	1.0
591	Middle East & North Africa	International	1	1	Killed	1.0
592	Middle East & North Africa	North Yemen	4	1	Wounded	1.0
593	Central Asia	Azerbaijan	5	0	Killed	1.0
594	Middle East & North Africa	Morocco	3	0	Wounded	1.0
595	Eastern Europe	Poland	5	0	Wounded	1.0
596	Middle East & North Africa	North Yemen	2	0	Killed	1.0
597	East Asia	South Korea	4	0	Wounded	1.0
598	Australasia & Oceania	New Zealand	11	1	Killed	1.0
599	Eastern Europe	Lithuania	7	1	Killed	1.0
600	Eastern Europe	Lithuania	1	0	Wounded	1.0
601	Sub-Saharan Africa	Comoros	2	0	Killed	1.0
602	East Asia	Macau	27	1	Killed	1.0
603	Eastern Europe	Lithuania	7	1	Wounded	1.0
604	Eastern Europe	Macedonia	11	0	Killed	1.0
605	Sub-Saharan Africa	Burkina Faso	7	0	Wounded	1.0
606	Sub-Saharan Africa	Burkina Faso	7	0	Killed	1.0
607	Eastern Europe	Hungary	6	0	Wounded	1.0

	Region	Country	Num_Attacks	success	Types	Counts_Types
608	Southeast Asia	Vietnam	7	1	Killed	1.0
609	Sub-Saharan Africa	Guinea	4	0	Wounded	1.0
610	Central America & Caribbean	Bahamas	4	1	Killed	1.0
611	Eastern Europe	East Germany (GDR)	3	0	Wounded	1.0
612	Southeast Asia	Malaysia	13	0	Wounded	1.0
613	Australasia & Oceania	Australia	17	0	Wounded	1.0
614	Sub-Saharan Africa	Lesotho	4	0	Killed	1.0
615	Eastern Europe	Montenegro	5	1	Killed	1.0
616	Eastern Europe	Bulgaria	6	0	Wounded	0.0
617	Eastern Europe	Croatia	2	0	Wounded	0.0
618	Eastern Europe	Czech Republic	9	0	Killed	0.0
619	Eastern Europe	Croatia	2	0	Killed	0.0
620	Central Asia	Kyrgyzstan	8	0	Killed	0.0
621	East Asia	Macau	6	0	Killed	0.0
622	Western Europe	Denmark	6	0	Killed	0.0
623	Eastern Europe	Bosnia-Herzegovina	8	0	Killed	0.0
624	Eastern Europe	Albania	16	0	Killed	0.0
625	East Asia	Taiwan	13	0	Killed	0.0
626	Western Europe	Finland	1	0	Killed	0.0
627	Western Europe	Finland	1	0	Wounded	0.0
628	Western Europe	Germany	76	0	Killed	0.0
629	Central Asia	Uzbekistan	2	0	Wounded	0.0
630	East Asia	Japan	61	0	Killed	0.0
631	East Asia	South Korea	4	0	Killed	0.0
632	Central Asia	Turkmenistan	1	0	Killed	0.0
633	Western Europe	Luxembourg	2	0	Killed	0.0
634	Western Europe	Iceland	4	1	Killed	0.0
635	Australasia & Oceania	New Zealand	9	0	Wounded	0.0
636	Central America & Caribbean	Antigua and Barbuda	1	1	Wounded	0.0
637	Central America & Caribbean	Antigua and Barbuda	1	1	Killed	0.0
638	Australasia & Oceania	Wallis and Futuna	1	1	Wounded	0.0
639	Australasia & Oceania	Wallis and Futuna	1	1	Killed	0.0
640	Australasia & Oceania	Vanuatu	2	1	Wounded	0.0
641	Australasia & Oceania	Vanuatu	2	1	Killed	0.0
642	Australasia & Oceania	Solomon Islands	4	1	Wounded	0.0
643	Australasia & Oceania	New Zealand	9	0	Killed	0.0

	Region	Country	Num_Attacks	success	Types	Counts_Types
644	Central America & Caribbean	Bahamas	1	0	Wounded	0.0
645	Australasia & Oceania	New Hebrides	1	1	Wounded	0.0
646	Australasia & Oceania	New Hebrides	1	1	Killed	0.0
647	Australasia & Oceania	New Caledonia	3	0	Killed	0.0
648	Western Europe	Vatican City	1	0	Killed	0.0
649	Australasia & Oceania	French Polynesia	3	1	Killed	0.0
650	Australasia & Oceania	Fiji	1	0	Wounded	0.0
651	Australasia & Oceania	Fiji	1	0	Killed	0.0
652	Central America & Caribbean	Bahamas	1	0	Killed	0.0
653	Central America & Caribbean	Bahamas	4	1	Wounded	0.0
654	Western Europe	Iceland	4	1	Wounded	0.0
655	Western Europe	Malta	1	0	Killed	0.0
656	Central Asia	Armenia	4	0	Killed	0.0
657	Central America & Caribbean	Trinidad and Tobago	2	0	Killed	0.0
658	Central America & Caribbean	St. Kitts and Nevis	2	1	Killed	0.0
659	Eastern Europe	Czechoslovakia	3	0	Killed	0.0
660	Western Europe	Luxembourg	14	1	Killed	0.0
661	Western Europe	Luxembourg	2	0	Wounded	0.0
662	Central America & Caribbean	Martinique	12	1	Killed	0.0
663	Western Europe	Malta	1	0	Wounded	0.0
664	Central America & Caribbean	Belize	1	0	Killed	0.0
665	Central America & Caribbean	Guadeloupe	9	0	Killed	0.0
666	Western Europe	Norway	3	0	Killed	0.0
667	Central America & Caribbean	Dominican Republic	5	0	Killed	0.0
668	Central America & Caribbean	Dominica	1	0	Wounded	0.0
669	Central America & Caribbean	Dominica	1	0	Killed	0.0
670	Central America & Caribbean	Belize	7	1	Wounded	0.0
671	Central America & Caribbean	Belize	1	0	Wounded	0.0
672	Eastern Europe	Czech Republic	9	0	Wounded	0.0
673	Sub-Saharan Africa	Liberia	2	0	Wounded	0.0
674	Eastern Europe	East Germany (GDR)	3	0	Killed	0.0
675	Sub-Saharan Africa	Republic of the Congo	1	0	Wounded	0.0
676	Southeast Asia	Malaysia	13	0	Killed	0.0
677	Southeast Asia	Laos	3	0	Wounded	0.0
678	Southeast Asia	Laos	3	0	Killed	0.0
679	Sub-Saharan Africa	People's Republic of the Congo	1	0	Killed	0.0

	Region	Country	Num_Attacks	success	Types	Counts_Types
680	Sub-Saharan Africa	People's Republic of the Congo	1	0	Wounded	0.0
681	Sub-Saharan Africa	People's Republic of the Congo	3	1	Wounded	0.0
682	Sub-Saharan Africa	Republic of the Congo	1	0	Killed	0.0
683	Sub-Saharan Africa	Rhodesia	3	0	Killed	0.0
684	Southeast Asia	Vietnam	5	0	Killed	0.0
685	Southeast Asia	Brunei	6	0	Killed	0.0
686	Sub-Saharan Africa	Rhodesia	3	0	Wounded	0.0
687	South Asia	Mauritius	2	0	Killed	0.0
688	South Asia	Maldives	4	0	Killed	0.0
689	South America	Uruguay	11	0	Wounded	0.0
690	South America	Uruguay	11	0	Killed	0.0
691	Sub-Saharan Africa	Senegal	2	0	Killed	0.0
692	Southeast Asia	South Vietnam	1	1	Wounded	0.0
693	Southeast Asia	Vietnam	5	0	Wounded	0.0
694	Western Europe	Belgium	29	0	Killed	0.0
695	Sub-Saharan Africa	Gambia	1	1	Killed	0.0
696	Sub-Saharan Africa	Ivory Coast	7	0	Wounded	0.0
697	Sub-Saharan Africa	Guinea-Bissau	1	0	Wounded	0.0
698	Sub-Saharan Africa	Guinea-Bissau	1	0	Killed	0.0
699	Sub-Saharan Africa	Malawi	5	1	Wounded	0.0
700	Sub-Saharan Africa	Guinea	4	0	Killed	0.0
701	Sub-Saharan Africa	Ghana	4	0	Killed	0.0
702	Sub-Saharan Africa	Gambia	2	0	Wounded	0.0
703	Sub-Saharan Africa	Gabon	6	1	Wounded	0.0
704	Sub-Saharan Africa	Benin	8	1	Killed	0.0
705	Sub-Saharan Africa	Gabon	2	0	Killed	0.0
706	Sub-Saharan Africa	Mauritania	4	0	Wounded	0.0
707	Sub-Saharan Africa	Comoros	2	0	Wounded	0.0
708	Sub-Saharan Africa	Comoros	3	1	Killed	0.0
709	Sub-Saharan Africa	Chad	4	0	Wounded	0.0
710	Sub-Saharan Africa	Cameroon	23	0	Wounded	0.0
711	Sub-Saharan Africa	Namibia	4	0	Killed	0.0
712	South America	Suriname	5	0	Wounded	0.0
713	South America	Paraguay	13	0	Killed	0.0
714	Sub-Saharan Africa	Seychelles	2	1	Killed	0.0
715	Eastern Europe	Moldova	3	0	Killed	0.0

	Region	Country	Num_Attacks	success	Types	Counts_Types
716	Eastern Europe	Serbia	1	0	Wounded	0.0
717	Eastern Europe	Serbia	1	0	Killed	0.0
718	Eastern Europe	Poland	5	0	Killed	0.0
719	Eastern Europe	Montenegro	5	1	Wounded	0.0
720	Sub-Saharan Africa	Zaire	5	0	Killed	0.0
721	Eastern Europe	Moldova	3	0	Wounded	0.0
722	Sub-Saharan Africa	Zaire	5	0	Wounded	0.0
723	Sub-Saharan Africa	Liberia	2	0	Killed	0.0
724	Sub-Saharan Africa	Seychelles	2	1	Wounded	0.0
725	Eastern Europe	Lithuania	1	0	Killed	0.0
726	Eastern Europe	Latvia	4	0	Wounded	0.0
727	Eastern Europe	Latvia	4	0	Killed	0.0
728	Western Europe	Andorra	1	1	Killed	0.0
729	Western Europe	Andorra	1	1	Wounded	0.0
730	Eastern Europe	Kosovo	24	0	Killed	0.0
731	Eastern Europe	Hungary	6	0	Killed	0.0
732	Eastern Europe	Serbia-Montenegro	1	0	Killed	0.0
733	Eastern Europe	Serbia-Montenegro	1	0	Wounded	0.0
734	Eastern Europe	Slovak Republic	3	0	Wounded	0.0
735	Sub-Saharan Africa	Tanzania	9	0	Killed	0.0
736	South America	French Guiana	1	0	Wounded	0.0
737	Sub-Saharan Africa	Sierra Leone	3	0	Wounded	0.0
738	South America	French Guiana	1	0	Killed	0.0
739	South America	Falkland Islands	1	1	Wounded	0.0
740	South America	Falkland Islands	1	1	Killed	0.0
741	North America	Canada	21	0	Killed	0.0
742	Middle East & North Africa	United Arab Emirates	5	0	Killed	0.0
743	Middle East & North Africa	South Yemen	2	1	Killed	0.0
744	Middle East & North Africa	Qatar	1	0	Wounded	0.0
745	Middle East & North Africa	Qatar	1	0	Killed	0.0
746	Middle East & North Africa	North Yemen	2	0	Wounded	0.0
747	Middle East & North Africa	Morocco	3	0	Killed	0.0
748	Middle East & North Africa	Bahrain	28	0	Killed	0.0
749	Sub-Saharan Africa	Swaziland	1	0	Killed	0.0
750	Sub-Saharan Africa	Swaziland	1	0	Wounded	0.0
751	Sub-Saharan Africa	Zambia	4	0	Killed	0.0

In [ ]:

In [54]:

```
# Group by Country and Year and sum the number of killings
country_killings = df_terr.groupby(['Country','success'])[['Killed','Wounded']].sum().rese

# Sort the data by the number of killings in descending order
# top_countries=country_killings.nlargest(10,'Killed')
# top_countries
country_killings
```

Out[54]:

	Country	success	Killed	Wounded
0	Afghanistan	0	2832.0	2634.0
1	Afghanistan	1	36552.0	41643.0
2	Albania	0	0.0	11.0
3	Albania	1	42.0	115.0
4	Algeria	0	58.0	180.0
5	Algeria	1	11008.0	8970.0
6	Andorra	1	0.0	0.0
7	Angola	0	38.0	23.0
8	Angola	1	3005.0	2432.0
9	Antigua and Barbuda	1	0.0	0.0
10	Argentina	0	6.0	38.0
11	Argentina	1	484.0	717.0
12	Armenia	0	0.0	1.0
13	Armenia	1	37.0	70.0
14	Australia	0	1.0	1.0
15	Australia	1	22.0	112.0
16	Austria	0	2.0	4.0
17	Austria	1	28.0	122.0
18	Azerbaijan	0	1.0	7.0
19	Azerbaijan	1	257.0	180.0
20	Bahamas	0	0.0	0.0
21	Bahamas	1	1.0	0.0
22	Bahrain	0	0.0	1.0
23	Bahrain	1	44.0	188.0
24	Bangladesh	0	17.0	233.0
25	Bangladesh	1	1227.0	7992.0
26	Barbados	1	76.0	3.0
27	Belarus	1	14.0	212.0
28	Belgium	0	0.0	4.0

	Country	success	Killed	Wounded
29	Belgium	1	79.0	512.0
30	Belize	0	0.0	0.0
31	Belize	1	3.0	0.0
32	Benin	1	0.0	8.0
33	Bhutan	1	9.0	5.0
34	Bolivia	0	2.0	6.0
35	Bolivia	1	40.0	160.0
36	Bosnia-Herzegovina	0	0.0	2.0
37	Bosnia-Herzegovina	1	79.0	148.0
38	Botswana	1	11.0	11.0
39	Brazil	0	2.0	12.0
40	Brazil	1	201.0	148.0
41	Brunei	0	0.0	1.0
42	Bulgaria	0	2.0	0.0
43	Bulgaria	1	26.0	36.0
44	Burkina Faso	0	1.0	1.0
45	Burkina Faso	1	133.0	218.0
46	Burundi	0	24.0	22.0
47	Burundi	1	4181.0	2432.0
48	Cambodia	0	1.0	4.0
49	Cambodia	1	542.0	782.0
50	Cameroon	0	49.0	0.0
51	Cameroon	1	2298.0	1063.0
52	Canada	0	0.0	7.0
53	Canada	1	365.0	139.0
54	Central African Republic	0	7.0	11.0
55	Central African Republic	1	1983.0	931.0
56	Chad	0	17.0	0.0
57	Chad	1	1102.0	1681.0
58	Chile	0	21.0	58.0
59	Chile	1	207.0	697.0
60	China	0	6.0	16.0
61	China	1	1002.0	1826.0
62	Colombia	0	317.0	311.0
63	Colombia	1	14381.0	10017.0
64	Comoros	0	1.0	0.0

	Country	success	Killed	Wounded
65	Comoros	1	0.0	2.0
66	Costa Rica	0	5.0	4.0
67	Costa Rica	1	12.0	34.0
68	Croatia	0	0.0	0.0
69	Croatia	1	248.0	73.0
70	Cuba	0	4.0	3.0
71	Cuba	1	4.0	7.0
72	Cyprus	0	3.0	5.0
73	Cyprus	1	42.0	36.0
74	Czech Republic	0	0.0	0.0
75	Czech Republic	1	6.0	29.0
76	Czechoslovakia	0	0.0	2.0
77	Czechoslovakia	1	27.0	21.0
78	Democratic Republic of the Congo	0	78.0	4.0
79	Democratic Republic of the Congo	1	3991.0	1365.0
80	Denmark	0	0.0	1.0
81	Denmark	1	5.0	28.0
82	Djibouti	1	274.0	162.0
83	Dominica	0	0.0	0.0
84	Dominica	1	3.0	10.0
85	Dominican Republic	0	0.0	1.0
86	Dominican Republic	1	34.0	123.0
87	East Germany (GDR)	0	0.0	1.0
88	East Germany (GDR)	1	2.0	36.0
89	East Timor	0	2.0	1.0
90	East Timor	1	7.0	6.0
91	Ecuador	0	1.0	5.0
92	Ecuador	1	53.0	74.0
93	Egypt	0	281.0	183.0
94	Egypt	1	3588.0	4639.0
95	El Salvador	0	49.0	49.0
96	El Salvador	1	12004.0	5013.0
97	Equatorial Guinea	1	2.0	3.0
98	Eritrea	1	46.0	62.0
99	Estonia	1	3.0	11.0
100	Ethiopia	0	17.0	2.0

	Country	success	Killed	Wounded
101	Ethiopia	1	1748.0	1212.0
102	Falkland Islands	1	0.0	0.0
103	Fiji	0	0.0	0.0
104	Fiji	1	8.0	18.0
105	Finland	0	0.0	0.0
106	Finland	1	11.0	27.0
107	France	0	21.0	57.0
108	France	1	513.0	2459.0
109	French Guiana	0	0.0	0.0
110	French Guiana	1	1.0	13.0
111	French Polynesia	1	0.0	13.0
112	Gabon	0	0.0	3.0
113	Gabon	1	6.0	0.0
114	Gambia	0	13.0	0.0
115	Gambia	1	0.0	2.0
116	Georgia	0	6.0	20.0
117	Georgia	1	272.0	372.0
118	Germany	0	0.0	18.0
119	Germany	1	84.0	665.0
120	Ghana	0	0.0	4.0
121	Ghana	1	19.0	9.0
122	Greece	0	4.0	26.0
123	Greece	1	321.0	707.0
124	Grenada	0	1.0	1.0
125	Grenada	1	8.0	20.0
126	Guadeloupe	0	0.0	2.0
127	Guadeloupe	1	8.0	43.0
128	Guatemala	0	34.0	53.0
129	Guatemala	1	5133.0	1178.0
130	Guinea	0	0.0	1.0
131	Guinea	1	213.0	56.0
132	Guinea-Bissau	0	0.0	0.0
133	Guinea-Bissau	1	17.0	27.0
134	Guyana	0	1.0	2.0
135	Guyana	1	40.0	22.0
136	Haiti	0	1.0	18.0

	Country	success	Killed	Wounded
137	Haiti	1	335.0	282.0
138	Honduras	0	3.0	19.0
139	Honduras	1	304.0	221.0
140	Hong Kong	0	2.0	27.0
141	Hong Kong	1	2.0	75.0
142	Hungary	0	0.0	1.0
143	Hungary	1	6.0	16.0
144	Iceland	1	0.0	0.0
145	India	0	222.0	607.0
146	India	1	19119.0	28373.0
147	Indonesia	0	11.0	17.0
148	Indonesia	1	1227.0	2428.0
149	International	1	1.0	12.0
150	Iran	0	26.0	53.0
151	Iran	1	1647.0	3976.0
152	Iraq	0	5553.0	2118.0
153	Iraq	1	73036.0	132572.0
154	Ireland	0	2.0	2.0
155	Ireland	1	115.0	29.0
156	Israel	0	39.0	141.0
157	Israel	1	1664.0	7805.0
158	Italy	0	13.0	56.0
159	Italy	1	407.0	1235.0
160	Ivory Coast	0	15.0	0.0
161	Ivory Coast	1	253.0	171.0
162	Jamaica	0	1.0	3.0
163	Jamaica	1	41.0	33.0
164	Japan	0	0.0	8.0
165	Japan	1	66.0	6990.0
166	Jordan	0	2.0	9.0
167	Jordan	1	131.0	251.0
168	Kazakhstan	0	2.0	1.0
169	Kazakhstan	1	37.0	20.0
170	Kenya	0	27.0	16.0
171	Kenya	1	1921.0	6247.0
172	Kosovo	0	0.0	8.0

	Country	success	Killed	Wounded
173	Kosovo	1	83.0	357.0
174	Kuwait	0	2.0	9.0
175	Kuwait	1	61.0	291.0
176	Kyrgyzstan	0	0.0	1.0
177	Kyrgyzstan	1	10.0	25.0
178	Laos	0	0.0	0.0
179	Laos	1	27.0	73.0
180	Latvia	0	0.0	0.0
181	Latvia	1	2.0	36.0
182	Lebanon	0	65.0	251.0
183	Lebanon	1	3996.0	10653.0
184	Lesotho	0	1.0	5.0
185	Lesotho	1	45.0	29.0
186	Liberia	0	0.0	0.0
187	Liberia	1	177.0	36.0
188	Libya	0	91.0	148.0
189	Libya	1	2507.0	3162.0
190	Lithuania	0	0.0	1.0
191	Lithuania	1	1.0	1.0
192	Luxembourg	0	0.0	0.0
193	Luxembourg	1	0.0	6.0
194	Macau	0	0.0	2.0
195	Macau	1	1.0	44.0
196	Macedonia	0	1.0	7.0
197	Macedonia	1	48.0	53.0
198	Madagascar	0	8.0	16.0
199	Madagascar	1	23.0	169.0
200	Malawi	1	33.0	0.0
201	Malaysia	0	0.0	1.0
202	Malaysia	1	152.0	100.0
203	Maldives	0	0.0	4.0
204	Maldives	1	20.0	118.0
205	Mali	0	20.0	5.0
206	Mali	1	1412.0	1351.0
207	Malta	0	0.0	0.0
208	Malta	1	5.0	12.0

	Country	success	Killed	Wounded
209	Martinique	1	0.0	1.0
210	Mauritania	0	1.0	0.0
211	Mauritania	1	42.0	28.0
212	Mauritius	0	0.0	1.0
213	Mexico	0	21.0	23.0
214	Mexico	1	759.0	660.0
215	Moldova	0	0.0	0.0
216	Moldova	1	13.0	88.0
217	Montenegro	1	1.0	0.0
218	Morocco	0	0.0	1.0
219	Morocco	1	292.0	199.0
220	Mozambique	0	22.0	4.0
221	Mozambique	1	2689.0	1514.0
222	Myanmar	0	20.0	7.0
223	Myanmar	1	1260.0	1624.0
224	Namibia	0	0.0	1.0
225	Namibia	1	220.0	405.0
226	Nepal	0	4.0	53.0
227	Nepal	1	1965.0	2098.0
228	Netherlands	0	8.0	14.0
229	Netherlands	1	29.0	44.0
230	New Caledonia	0	0.0	2.0
231	New Caledonia	1	35.0	21.0
232	New Hebrides	1	0.0	0.0
233	New Zealand	0	0.0	0.0
234	New Zealand	1	1.0	2.0
235	Nicaragua	0	29.0	25.0
236	Nicaragua	1	10569.0	1706.0
237	Niger	0	118.0	5.0
238	Niger	1	1356.0	467.0
239	Nigeria	0	454.0	133.0
240	Nigeria	1	22228.0	10106.0
241	North Korea	1	3.0	4.0
242	North Yemen	0	1.0	0.0
243	North Yemen	1	2.0	1.0
244	Norway	0	0.0	1.0

	Country	success	Killed	Wounded
245	Norway	1	79.0	87.0
246	Pakistan	0	528.0	906.0
247	Pakistan	1	23294.0	41132.0
248	Panama	0	1.0	1.0
249	Panama	1	37.0	82.0
250	Papua New Guinea	0	5.0	4.0
251	Papua New Guinea	1	74.0	87.0
252	Paraguay	0	0.0	6.0
253	Paraguay	1	59.0	63.0
254	People's Republic of the Congo	0	0.0	0.0
255	People's Republic of the Congo	1	15.0	0.0
256	Peru	0	140.0	150.0
257	Peru	1	12631.0	3928.0
258	Philippines	0	270.0	512.0
259	Philippines	1	9289.0	12855.0
260	Poland	0	0.0	1.0
261	Poland	1	9.0	30.0
262	Portugal	0	1.0	9.0
263	Portugal	1	31.0	86.0
264	Qatar	0	0.0	0.0
265	Qatar	1	7.0	13.0
266	Republic of the Congo	0	0.0	0.0
267	Republic of the Congo	1	182.0	54.0
268	Rhodesia	0	0.0	0.0
269	Rhodesia	1	217.0	158.0
270	Romania	0	1.0	3.0
271	Romania	1	3.0	6.0
272	Russia	0	116.0	152.0
273	Russia	1	4192.0	7289.0
274	Rwanda	0	1.0	1.0
275	Rwanda	1	3235.0	921.0
276	Saudi Arabia	0	41.0	35.0
277	Saudi Arabia	1	631.0	1631.0
278	Senegal	0	0.0	1.0
279	Senegal	1	325.0	322.0
280	Serbia	0	0.0	0.0

	Country	success	Killed	Wounded
281	Serbia	1	3.0	9.0
282	Serbia-Montenegro	0	0.0	0.0
283	Serbia-Montenegro	1	3.0	5.0
284	Seychelles	1	0.0	0.0
285	Sierra Leone	0	7.0	0.0
286	Sierra Leone	1	833.0	122.0
287	Singapore	1	5.0	3.0
288	Slovak Republic	0	1.0	0.0
289	Slovak Republic	1	6.0	11.0
290	Slovenia	1	1.0	2.0
291	Solomon Islands	1	4.0	0.0
292	Somalia	0	449.0	364.0
293	Somalia	1	9824.0	8511.0
294	South Africa	0	27.0	87.0
295	South Africa	1	2647.0	4458.0
296	South Korea	0	0.0	1.0
297	South Korea	1	10.0	133.0
298	South Sudan	0	119.0	12.0
299	South Sudan	1	2515.0	1311.0
300	South Vietnam	1	81.0	0.0
301	South Yemen	1	0.0	2.0
302	Soviet Union	0	3.0	1.0
303	Soviet Union	1	93.0	149.0
304	Spain	0	34.0	261.0
305	Spain	1	1254.0	4674.0
306	Sri Lanka	0	153.0	368.0
307	Sri Lanka	1	15377.0	15193.0
308	St. Kitts and Nevis	1	0.0	9.0
309	St. Lucia	1	2.0	12.0
310	Sudan	0	82.0	22.0
311	Sudan	1	3801.0	2147.0
312	Suriname	0	3.0	0.0
313	Suriname	1	26.0	24.0
314	Swaziland	0	0.0	0.0
315	Swaziland	1	6.0	3.0
316	Sweden	0	1.0	3.0

	Country	success	Killed	Wounded
317	Sweden	1	21.0	77.0
318	Switzerland	0	1.0	13.0
319	Switzerland	1	73.0	81.0
320	Syria	0	110.0	62.0
321	Syria	1	15119.0	14047.0
322	Taiwan	0	0.0	9.0
323	Taiwan	1	60.0	78.0
324	Tajikistan	0	2.0	6.0
325	Tajikistan	1	305.0	1103.0
326	Tanzania	0	0.0	2.0
327	Tanzania	1	73.0	232.0
328	Thailand	0	23.0	164.0
329	Thailand	1	2719.0	7654.0
330	Togo	0	16.0	2.0
331	Togo	1	60.0	32.0
332	Trinidad and Tobago	0	0.0	1.0
333	Trinidad and Tobago	1	6.0	34.0
334	Tunisia	0	14.0	2.0
335	Tunisia	1	337.0	431.0
336	Turkey	0	183.0	197.0
337	Turkey	1	6705.0	9702.0
338	Turkmenistan	0	0.0	1.0
339	Turkmenistan	1	3.0	2.0
340	Uganda	0	8.0	34.0
341	Uganda	1	3057.0	1111.0
342	Ukraine	0	6.0	72.0
343	Ukraine	1	2255.0	2769.0
344	United Arab Emirates	0	0.0	2.0
345	United Arab Emirates	1	123.0	25.0
346	United Kingdom	0	110.0	243.0
347	United Kingdom	1	3300.0	5863.0
348	United States	0	13.0	68.0
349	United States	1	3758.0	20634.0
350	Uruguay	0	0.0	0.0
351	Uruguay	1	6.0	6.0
352	Uzbekistan	0	1.0	0.0

	Country	success	Killed	Wounded
353	Uzbekistan	1	67.0	200.0
354	Vanuatu	1	0.0	0.0
355	Vatican City	0	0.0	3.0
356	Venezuela	0	5.0	15.0
357	Venezuela	1	222.0	232.0
358	Vietnam	0	0.0	0.0
359	Vietnam	1	1.0	27.0
360	Wallis and Futuna	1	0.0	0.0
361	West Bank and Gaza Strip	0	178.0	206.0
362	West Bank and Gaza Strip	1	1322.0	2808.0
363	West Germany (FRG)	0	4.0	15.0
364	West Germany (FRG)	1	93.0	847.0
365	Western Sahara	1	1.0	4.0
366	Yemen	0	377.0	341.0
367	Yemen	1	8399.0	8987.0
368	Yugoslavia	0	5.0	7.0
369	Yugoslavia	1	114.0	274.0
370	Zaire	0	0.0	0.0
371	Zaire	1	324.0	211.0
372	Zambia	0	0.0	1.0
373	Zambia	1	70.0	61.0
374	Zimbabwe	0	2.0	2.0
375	Zimbabwe	1	152.0	220.0

In [55]:

```
# Group by Country and Year and sum the number of killings
country_killings = df_terr.groupby(['Country','success'])['Killed'].sum().reset_index()

# Sort the data by the number of killings in descending order
# top_countries=country_killings.nlargest(10,'Killed')
# top_countries
country_killings
```

Out[55]:

	Country	success	Killed
0	Afghanistan	0	2832.0
1	Afghanistan	1	36552.0
2	Albania	0	0.0
3	Albania	1	42.0
4	Algeria	0	58.0
5	Algeria	1	11008.0

	Country	success	Killed
6	Andorra	1	0.0
7	Angola	0	38.0
8	Angola	1	3005.0
9	Antigua and Barbuda	1	0.0
10	Argentina	0	6.0
11	Argentina	1	484.0
12	Armenia	0	0.0
13	Armenia	1	37.0
14	Australia	0	1.0
15	Australia	1	22.0
16	Austria	0	2.0
17	Austria	1	28.0
18	Azerbaijan	0	1.0
19	Azerbaijan	1	257.0
20	Bahamas	0	0.0
21	Bahamas	1	1.0
22	Bahrain	0	0.0
23	Bahrain	1	44.0
24	Bangladesh	0	17.0
25	Bangladesh	1	1227.0
26	Barbados	1	76.0
27	Belarus	1	14.0
28	Belgium	0	0.0
29	Belgium	1	79.0
30	Belize	0	0.0
31	Belize	1	3.0
32	Benin	1	0.0
33	Bhutan	1	9.0
34	Bolivia	0	2.0
35	Bolivia	1	40.0
36	Bosnia-Herzegovina	0	0.0
37	Bosnia-Herzegovina	1	79.0
38	Botswana	1	11.0
39	Brazil	0	2.0
40	Brazil	1	201.0
41	Brunei	0	0.0

	Country	success	Killed
42	Bulgaria	0	2.0
43	Bulgaria	1	26.0
44	Burkina Faso	0	1.0
45	Burkina Faso	1	133.0
46	Burundi	0	24.0
47	Burundi	1	4181.0
48	Cambodia	0	1.0
49	Cambodia	1	542.0
50	Cameroon	0	49.0
51	Cameroon	1	2298.0
52	Canada	0	0.0
53	Canada	1	365.0
54	Central African Republic	0	7.0
55	Central African Republic	1	1983.0
56	Chad	0	17.0
57	Chad	1	1102.0
58	Chile	0	21.0
59	Chile	1	207.0
60	China	0	6.0
61	China	1	1002.0
62	Colombia	0	317.0
63	Colombia	1	14381.0
64	Comoros	0	1.0
65	Comoros	1	0.0
66	Costa Rica	0	5.0
67	Costa Rica	1	12.0
68	Croatia	0	0.0
69	Croatia	1	248.0
70	Cuba	0	4.0
71	Cuba	1	4.0
72	Cyprus	0	3.0
73	Cyprus	1	42.0
74	Czech Republic	0	0.0
75	Czech Republic	1	6.0
76	Czechoslovakia	0	0.0
77	Czechoslovakia	1	27.0

	Country	success	Killed
78	Democratic Republic of the Congo	0	78.0
79	Democratic Republic of the Congo	1	3991.0
80	Denmark	0	0.0
81	Denmark	1	5.0
82	Djibouti	1	274.0
83	Dominica	0	0.0
84	Dominica	1	3.0
85	Dominican Republic	0	0.0
86	Dominican Republic	1	34.0
87	East Germany (GDR)	0	0.0
88	East Germany (GDR)	1	2.0
89	East Timor	0	2.0
90	East Timor	1	7.0
91	Ecuador	0	1.0
92	Ecuador	1	53.0
93	Egypt	0	281.0
94	Egypt	1	3588.0
95	El Salvador	0	49.0
96	El Salvador	1	12004.0
97	Equatorial Guinea	1	2.0
98	Eritrea	1	46.0
99	Estonia	1	3.0
100	Ethiopia	0	17.0
101	Ethiopia	1	1748.0
102	Falkland Islands	1	0.0
103	Fiji	0	0.0
104	Fiji	1	8.0
105	Finland	0	0.0
106	Finland	1	11.0
107	France	0	21.0
108	France	1	513.0
109	French Guiana	0	0.0
110	French Guiana	1	1.0
111	French Polynesia	1	0.0
112	Gabon	0	0.0
113	Gabon	1	6.0

	Country	success	Killed
114	Gambia	0	13.0
115	Gambia	1	0.0
116	Georgia	0	6.0
117	Georgia	1	272.0
118	Germany	0	0.0
119	Germany	1	84.0
120	Ghana	0	0.0
121	Ghana	1	19.0
122	Greece	0	4.0
123	Greece	1	321.0
124	Grenada	0	1.0
125	Grenada	1	8.0
126	Guadeloupe	0	0.0
127	Guadeloupe	1	8.0
128	Guatemala	0	34.0
129	Guatemala	1	5133.0
130	Guinea	0	0.0
131	Guinea	1	213.0
132	Guinea-Bissau	0	0.0
133	Guinea-Bissau	1	17.0
134	Guyana	0	1.0
135	Guyana	1	40.0
136	Haiti	0	1.0
137	Haiti	1	335.0
138	Honduras	0	3.0
139	Honduras	1	304.0
140	Hong Kong	0	2.0
141	Hong Kong	1	2.0
142	Hungary	0	0.0
143	Hungary	1	6.0
144	Iceland	1	0.0
145	India	0	222.0
146	India	1	19119.0
147	Indonesia	0	11.0
148	Indonesia	1	1227.0
149	International	1	1.0

	Country	success	Killed
150	Iran	0	26.0
151	Iran	1	1647.0
152	Iraq	0	5553.0
153	Iraq	1	73036.0
154	Ireland	0	2.0
155	Ireland	1	115.0
156	Israel	0	39.0
157	Israel	1	1664.0
158	Italy	0	13.0
159	Italy	1	407.0
160	Ivory Coast	0	15.0
161	Ivory Coast	1	253.0
162	Jamaica	0	1.0
163	Jamaica	1	41.0
164	Japan	0	0.0
165	Japan	1	66.0
166	Jordan	0	2.0
167	Jordan	1	131.0
168	Kazakhstan	0	2.0
169	Kazakhstan	1	37.0
170	Kenya	0	27.0
171	Kenya	1	1921.0
172	Kosovo	0	0.0
173	Kosovo	1	83.0
174	Kuwait	0	2.0
175	Kuwait	1	61.0
176	Kyrgyzstan	0	0.0
177	Kyrgyzstan	1	10.0
178	Laos	0	0.0
179	Laos	1	27.0
180	Latvia	0	0.0
181	Latvia	1	2.0
182	Lebanon	0	65.0
183	Lebanon	1	3996.0
184	Lesotho	0	1.0
185	Lesotho	1	45.0

	Country	success	Killed
186	Liberia	0	0.0
187	Liberia	1	177.0
188	Libya	0	91.0
189	Libya	1	2507.0
190	Lithuania	0	0.0
191	Lithuania	1	1.0
192	Luxembourg	0	0.0
193	Luxembourg	1	0.0
194	Macau	0	0.0
195	Macau	1	1.0
196	Macedonia	0	1.0
197	Macedonia	1	48.0
198	Madagascar	0	8.0
199	Madagascar	1	23.0
200	Malawi	1	33.0
201	Malaysia	0	0.0
202	Malaysia	1	152.0
203	Maldives	0	0.0
204	Maldives	1	20.0
205	Mali	0	20.0
206	Mali	1	1412.0
207	Malta	0	0.0
208	Malta	1	5.0
209	Martinique	1	0.0
210	Mauritania	0	1.0
211	Mauritania	1	42.0
212	Mauritius	0	0.0
213	Mexico	0	21.0
214	Mexico	1	759.0
215	Moldova	0	0.0
216	Moldova	1	13.0
217	Montenegro	1	1.0
218	Morocco	0	0.0
219	Morocco	1	292.0
220	Mozambique	0	22.0
221	Mozambique	1	2689.0

	Country	success	Killed
222	Myanmar	0	20.0
223	Myanmar	1	1260.0
224	Namibia	0	0.0
225	Namibia	1	220.0
226	Nepal	0	4.0
227	Nepal	1	1965.0
228	Netherlands	0	8.0
229	Netherlands	1	29.0
230	New Caledonia	0	0.0
231	New Caledonia	1	35.0
232	New Hebrides	1	0.0
233	New Zealand	0	0.0
234	New Zealand	1	1.0
235	Nicaragua	0	29.0
236	Nicaragua	1	10569.0
237	Niger	0	118.0
238	Niger	1	1356.0
239	Nigeria	0	454.0
240	Nigeria	1	22228.0
241	North Korea	1	3.0
242	North Yemen	0	1.0
243	North Yemen	1	2.0
244	Norway	0	0.0
245	Norway	1	79.0
246	Pakistan	0	528.0
247	Pakistan	1	23294.0
248	Panama	0	1.0
249	Panama	1	37.0
250	Papua New Guinea	0	5.0
251	Papua New Guinea	1	74.0
252	Paraguay	0	0.0
253	Paraguay	1	59.0
254	People's Republic of the Congo	0	0.0
255	People's Republic of the Congo	1	15.0
256	Peru	0	140.0
257	Peru	1	12631.0

	Country	success	Killed
258	Philippines	0	270.0
259	Philippines	1	9289.0
260	Poland	0	0.0
261	Poland	1	9.0
262	Portugal	0	1.0
263	Portugal	1	31.0
264	Qatar	0	0.0
265	Qatar	1	7.0
266	Republic of the Congo	0	0.0
267	Republic of the Congo	1	182.0
268	Rhodesia	0	0.0
269	Rhodesia	1	217.0
270	Romania	0	1.0
271	Romania	1	3.0
272	Russia	0	116.0
273	Russia	1	4192.0
274	Rwanda	0	1.0
275	Rwanda	1	3235.0
276	Saudi Arabia	0	41.0
277	Saudi Arabia	1	631.0
278	Senegal	0	0.0
279	Senegal	1	325.0
280	Serbia	0	0.0
281	Serbia	1	3.0
282	Serbia-Montenegro	0	0.0
283	Serbia-Montenegro	1	3.0
284	Seychelles	1	0.0
285	Sierra Leone	0	7.0
286	Sierra Leone	1	833.0
287	Singapore	1	5.0
288	Slovak Republic	0	1.0
289	Slovak Republic	1	6.0
290	Slovenia	1	1.0
291	Solomon Islands	1	4.0
292	Somalia	0	449.0
293	Somalia	1	9824.0

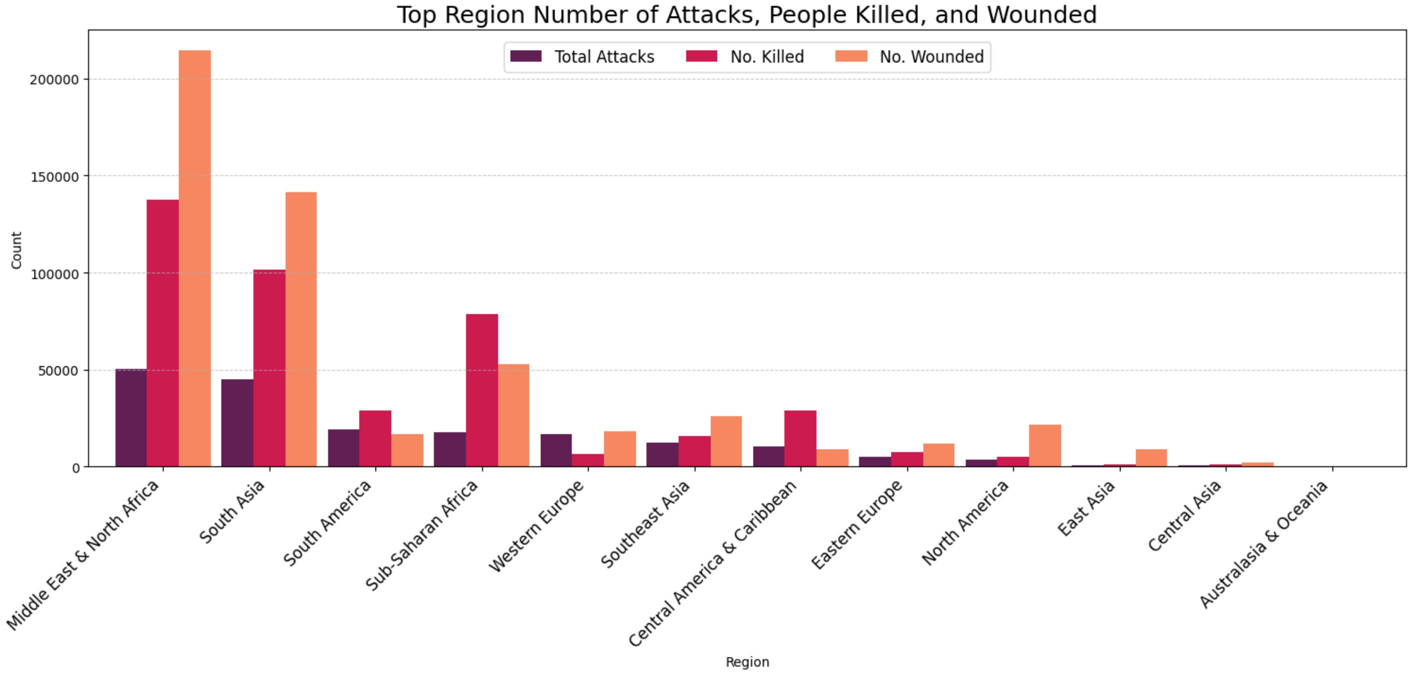
	Country	success	Killed
294	South Africa	0	27.0
295	South Africa	1	2647.0
296	South Korea	0	0.0
297	South Korea	1	10.0
298	South Sudan	0	119.0
299	South Sudan	1	2515.0
300	South Vietnam	1	81.0
301	South Yemen	1	0.0
302	Soviet Union	0	3.0
303	Soviet Union	1	93.0
304	Spain	0	34.0
305	Spain	1	1254.0
306	Sri Lanka	0	153.0
307	Sri Lanka	1	15377.0
308	St. Kitts and Nevis	1	0.0
309	St. Lucia	1	2.0
310	Sudan	0	82.0
311	Sudan	1	3801.0
312	Suriname	0	3.0
313	Suriname	1	26.0
314	Swaziland	0	0.0
315	Swaziland	1	6.0
316	Sweden	0	1.0
317	Sweden	1	21.0
318	Switzerland	0	1.0
319	Switzerland	1	73.0
320	Syria	0	110.0
321	Syria	1	15119.0
322	Taiwan	0	0.0
323	Taiwan	1	60.0
324	Tajikistan	0	2.0
325	Tajikistan	1	305.0
326	Tanzania	0	0.0
327	Tanzania	1	73.0
328	Thailand	0	23.0
329	Thailand	1	2719.0

	Country	success	Killed
330	Togo	0	16.0
331	Togo	1	60.0
332	Trinidad and Tobago	0	0.0
333	Trinidad and Tobago	1	6.0
334	Tunisia	0	14.0
335	Tunisia	1	337.0
336	Turkey	0	183.0
337	Turkey	1	6705.0
338	Turkmenistan	0	0.0
339	Turkmenistan	1	3.0
340	Uganda	0	8.0
341	Uganda	1	3057.0
342	Ukraine	0	6.0
343	Ukraine	1	2255.0
344	United Arab Emirates	0	0.0
345	United Arab Emirates	1	123.0
346	United Kingdom	0	110.0
347	United Kingdom	1	3300.0
348	United States	0	13.0
349	United States	1	3758.0
350	Uruguay	0	0.0
351	Uruguay	1	6.0
352	Uzbekistan	0	1.0
353	Uzbekistan	1	67.0
354	Vanuatu	1	0.0
355	Vatican City	0	0.0
356	Venezuela	0	5.0
357	Venezuela	1	222.0
358	Vietnam	0	0.0
359	Vietnam	1	1.0
360	Wallis and Futuna	1	0.0
361	West Bank and Gaza Strip	0	178.0
362	West Bank and Gaza Strip	1	1322.0
363	West Germany (FRG)	0	4.0
364	West Germany (FRG)	1	93.0
365	Western Sahara	1	1.0

	Country	success	Killed
366	Yemen	0	377.0
367	Yemen	1	8399.0
368	Yugoslavia	0	5.0
369	Yugoslavia	1	114.0
370	Zaire	0	0.0
371	Zaire	1	324.0
372	Zambia	0	0.0
373	Zambia	1	70.0
374	Zimbabwe	0	2.0
375	Zimbabwe	1	152.0

In [56]:

```
plt.style.use('default')
# Get the top 15 Region by number of attacks
reg_terror = df_terr['Region'].value_counts().to_frame()
reg_terror.columns = ['Attacks']

# Sum the number of people killed per Region
reg_kill = df_terr.groupby('Region')['Killed'].sum().to_frame()

# Sum the number of people wounded per Region
reg_wound = df_terr.groupby('Region')['Wounded'].sum().to_frame()

# Merge the information
reg_stats = reg_terror.merge(reg_kill, left_index=True, right_index=True, how='left')
reg_stats = reg_stats.merge(reg_wound, left_index=True, right_index=True, how='left')

# Use seaborn color palette
palette = sns.color_palette("rocket", 3)  # Choose a palette with 3 colors

# Plot the data
fig, ax = plt.subplots(figsize=(18, 6))

reg_stats.plot(kind='bar', width=0.9, ax=ax, color=palette)

ax.set_title('Top Region Number of Attacks, People Killed, and Wounded',fontsize=18)
ax.set_xlabel('Region')
ax.set_ylabel('Count')
# ax.legend( labels=['Total Attacks', 'No.Killed', 'No.Wounded'])
ax.legend(labels=['Total Attacks', 'No. Killed', 'No. Wounded'], loc='upper center', bbox_


plt.xticks(rotation=45,ha='right',fontsize=12)
plt.grid(axis='y', linestyle='--', linewidth=0.7, alpha=0.7)
# plt.tight_layout()
plt.show()
```

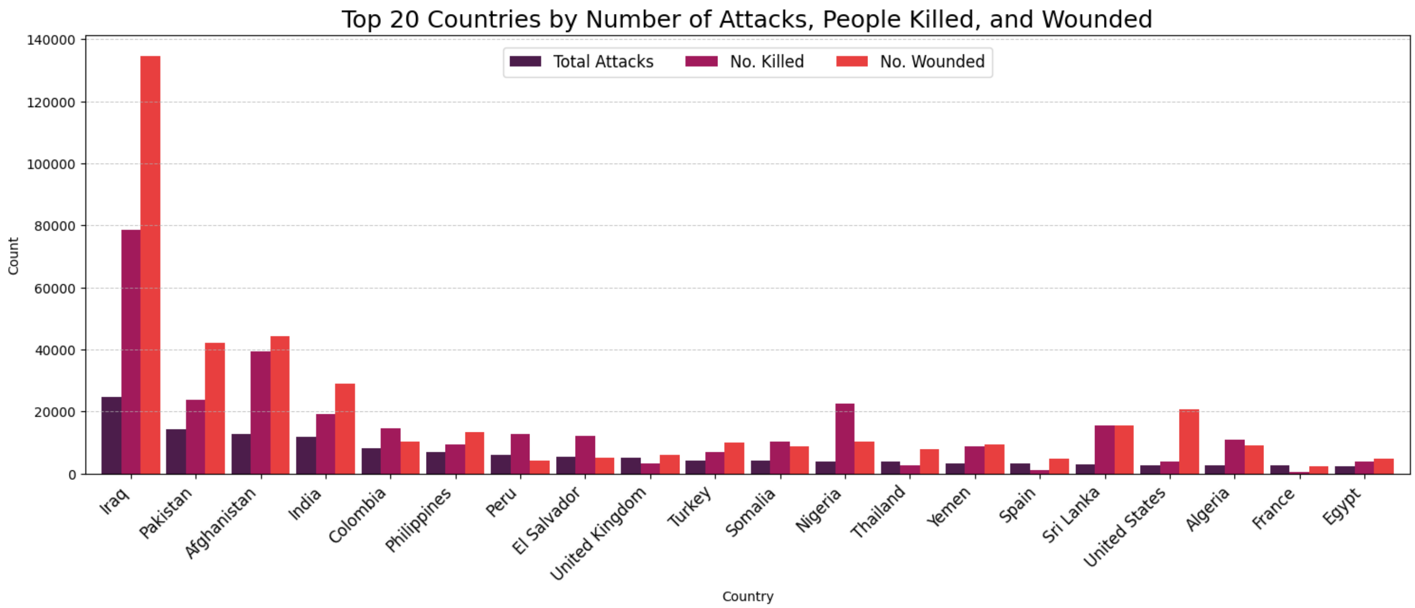


- > The `Middle East and North Africa` region is considered the region most vulnerable to terrorism, followed by `South Asia`, which has a large population density, which will lead to large numbers of deaths and injuries. Although the `Australian region` has witnessed only a very small number of terrorist incidents, there have been deaths and injuries there... We can say, in one way or another, that the continents of `Africa and Asia` are witnessing the highest terrorist attacks.

In [57]:

```
# Get the top 15 countries by number of attacks
coun_terror = df_terr['Country'].value_counts()[:20].to_frame()
coun_terror.columns = ['Attacks']

# Sum the number of people killed per country
coun_kill = df_terr.groupby('Country')['Killed'].sum().to_frame()

# Sum the number of people wounded per country
coun_wound = df_terr.groupby('Country')['Wounded'].sum().to_frame()

# Merge the information
coun_stats = coun_terror.merge(coun_kill, left_index=True, right_index=True, how='left')
coun_stats = coun_stats.merge(coun_wound, left_index=True, right_index=True, how='left')

# Use seaborn color palette
palette = sns.color_palette("rocket", 4)  # Choose a palette with 3 colors

# Plot the data
fig, ax = plt.subplots(figsize=(18, 6))

coun_stats.plot(kind='bar', width=0.9, ax=ax, color=palette)

ax.set_title('Top 20 Countries by Number of Attacks, People Killed, and Wounded',fontsize=
ax.set_xlabel('Country')
ax.set_ylabel('Count')
# ax.legend( labels=['Total Attacks', 'No.Killed', 'No.Wounded'])
ax.legend(labels=['Total Attacks', 'No. Killed', 'No. Wounded'], loc='upper center', bbox_


plt.xticks(rotation=45,ha='right',fontsize=12)
```

```
plt.grid(axis='y', linestyle='--', linewidth=0.7, alpha=0.7)
# plt.tight_layout()
plt.show()
```

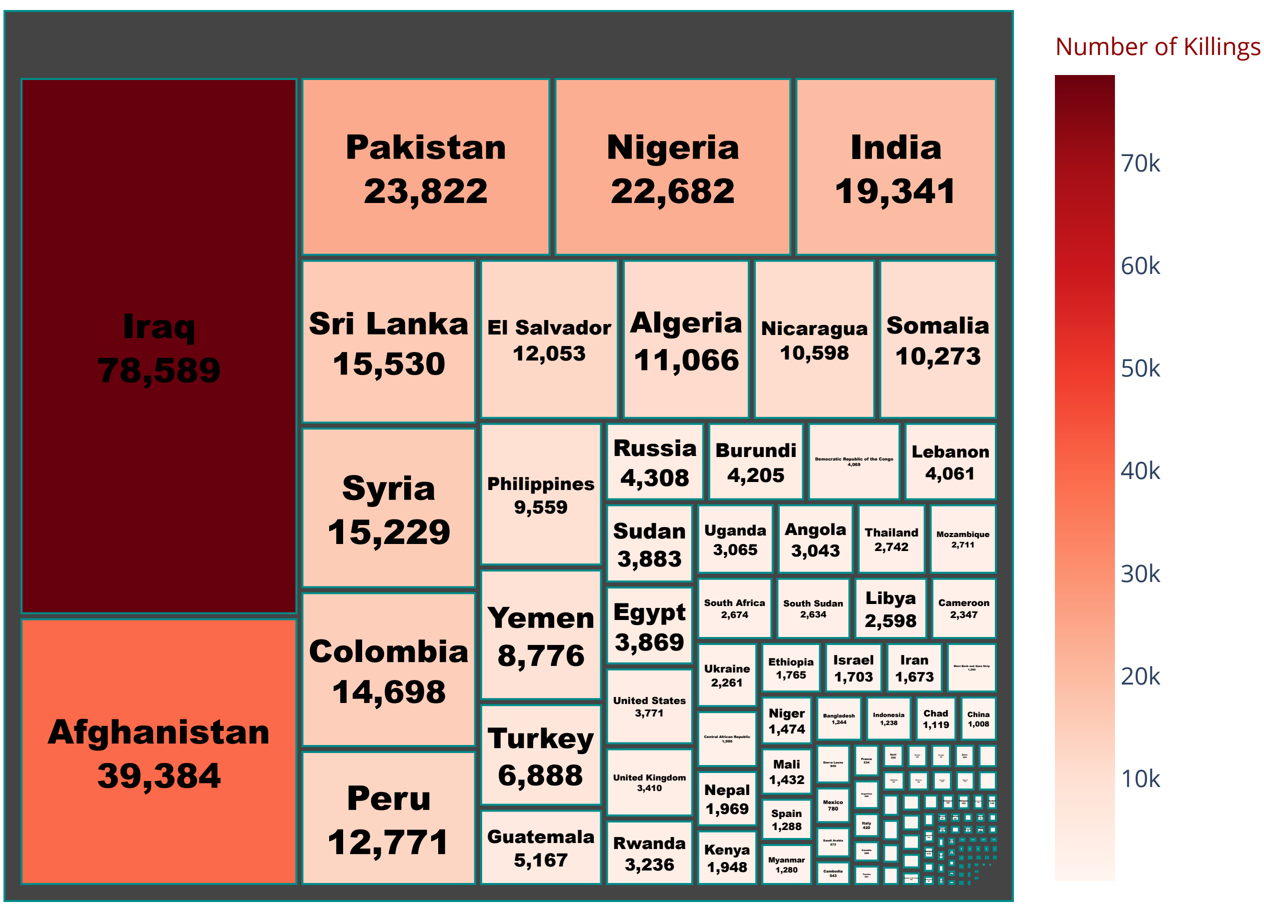


In [58]:

```
# its' okay
import plotly.express as px

# Group by Country and Year and sum the number of killings
country_killings = df_terr.groupby(['Country'])['Killed'].sum().reset_index()

# Sort the data by the number of killings in descending order
top_countries=country_killings.nlargest(150,'Killed')
top_countries

# Create a treemap with Plotly
fig = px.treemap(
    top_countries,
    path=['Country'],
    values='Killed',
    color='Killed',
    color_continuous_scale='Reds',
    title='Killings in Global Terrorism (Top 150 Countries)',
    labels={'Killed': 'Number of Killings'}
)

# Customize layout for better readability
fig.update_layout(
    title={
        'text': 'Killings in Global Terrorism (Top 150 Countries)',
        'font_size': 24,
        'font_family': 'Arial black',
        'x': 0.5,
        # 'y': 1,
        'xanchor': 'center'
    },
    margin=dict(t=60, l=10, r=10, b=20),
    coloraxis_colorbar=dict(
    #     title='Number of Killings',
        # tickvals=[0, top_countries['Killed'].max()],
        # ticktext=['Low', 'High'],
        # tickfont=dict(size=14, color='black', family='Arial',),
        title_font=dict(size=12, color='darkred'),
        lenmode='pixels',
    #     # len=3000
```

```python
    ),
    # uniformtext=dict(minsize=12, mode='hide'),  # Adjust text size and hiding mode
    showlegend=True   # Hide legend if not needed
)

# # Customize text for better readability
fig.update_traces(
    texttemplate='%{label}<br>%{value:,}',  # Format text for better readability
    textfont=dict(size=17, color='black', family='Arial black'),  # Customize text font a
    marker_line=dict(color='darkcyan', width=1),  # Add borders around the blocks
    hovertemplate='<b>%{label}</b><br>Number of Killings: %{value:,}<extra></extra>',  # I
    textposition='middle center', # Place text in the middle of the blocks

)

# # Add annotations to highlight specific data points (optional)
# annotations = [
#     dict(
#         x=1, y=1,
#         text='',
#         showarrow=False,
#         font=dict(size=20, color='white')
#     )
# ]
# fig.update_layout(annotations=annotations)

# Show plot
fig.show()
```

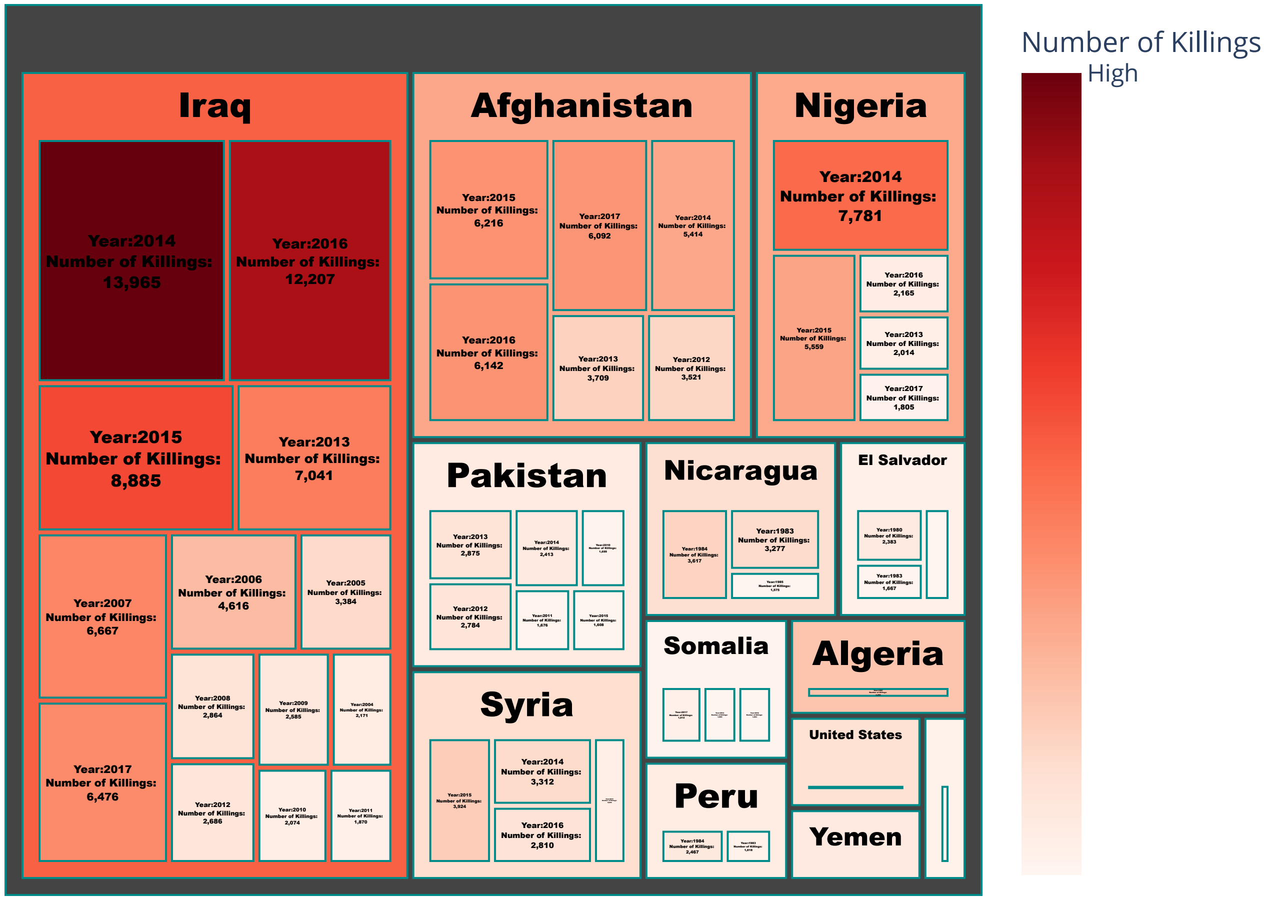
Killings in Global Terrorism (Top 150 Countries)



In [59]:

```
import plotly.express as px
# Group by Country and Year and sum the number of killings
country_year_killings = df_terr.groupby(['Country', 'Year'])['Killed'].sum().reset_index()

# Sort the data by the number of killings in descending order
top_countries_year=country_year_killings.nlargest(50,'Killed')
top_countries_year
# Create a treemap with Plotly
fig = px.treemap(
    top_countries_year,
    path=['Country', 'Year'],
    values='Killed',
    color='Killed',
    color_continuous_scale='Reds',
    title='Killings in Global Terrorism on Years (Top 50 Years under Countries)',
    labels={'Killed': 'Number of Killings'}
)

# Customize layout for better readability
fig.update_layout(
    title={
        # 'text': title,
        'font_size': 24,
        'font_family': 'Arial Black',
        'x': 0.5,
        'xanchor': 'center'
    },
    margin=dict(t=60, l=10, r=10, b=20),
    coloraxis_colorbar=dict(
    #     title='Number of Killings',
        tickvals=[0, top_countries_year['Killed'].max()],
        ticktext=['Low', 'High'],
    #     tickfont=dict(size=14, color='black', family='Arial'),
    #     title_font=dict(size=12, color='darkred'),
    #     lenmode='pixels',
    ),
    showlegend=True
)

# Customize text for better readability
fig.update_traces(
    texttemplate='Year:%{label}<br>Number of Killings: <br>%{value:,}',  # Format text fo
    textfont=dict(size=17, color='black', family='Arial Black'),  # Customize text font a
    marker_line=dict(color='darkcyan', width=1),  # Add borders around the blocks
    hovertemplate='Year:%{label}</b><br>Number of Killings: %{value:,}<extra></extra>',  #
    textposition='middle center',  # Place text in the middle of the blocks
)

# Add annotations to highlight specific data points (optional)
annotations = [
    dict(
        x=0.5, y=1.05,
        text=' ',
        showarrow=False,
        font=dict(size=14, color='grey'),
        xref='paper',
        yref='paper'
    )
]
fig.update_layout(annotations=annotations)

# Show plot
fig.show()
```

n Global Terrorism on Years (Top 50 Years under C



The type of attack and its impact on the number of casualties killed and wounded.

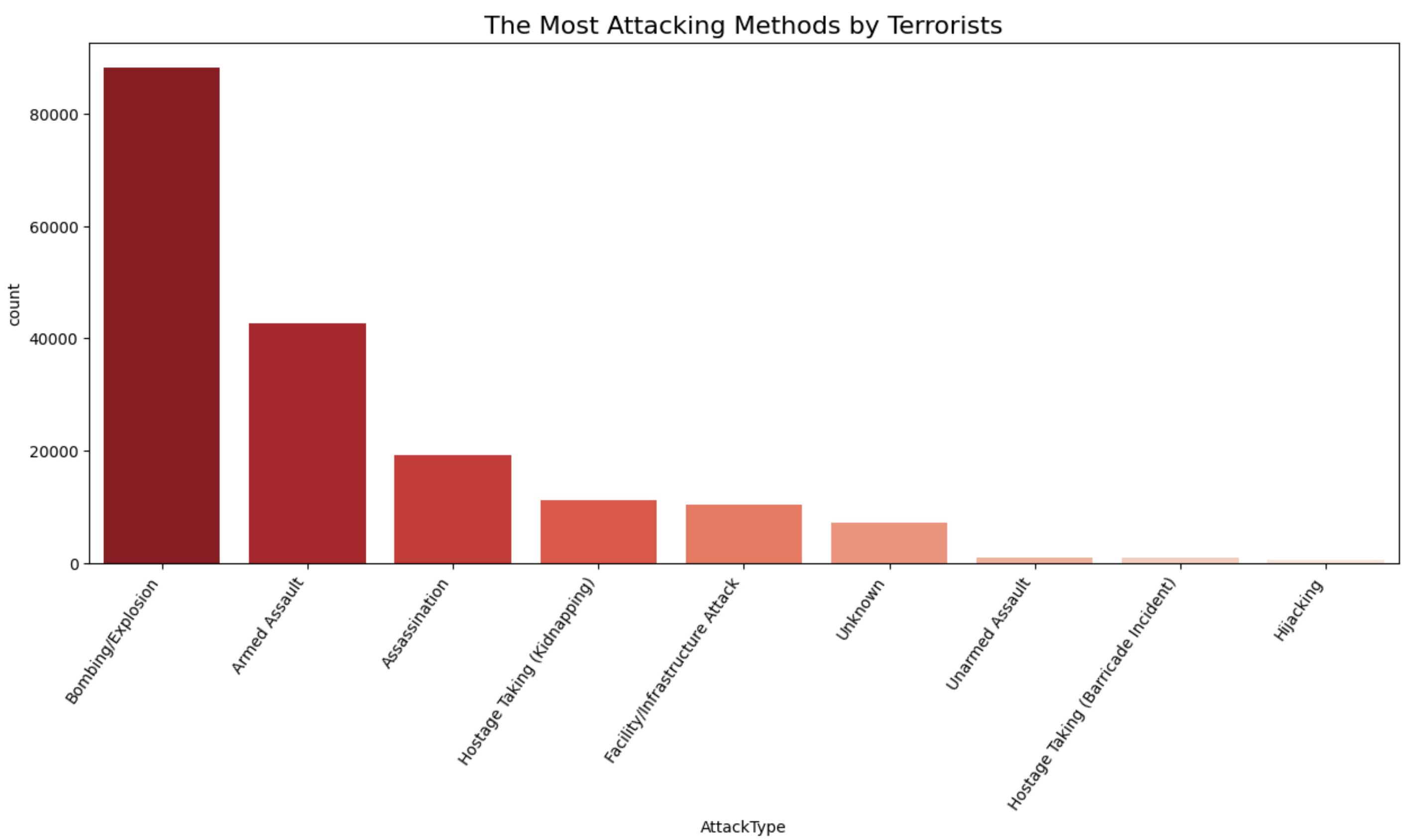
In [60]:

```
# type of attack has an effect on the number of people killed and wounded.
killed_wounded_attack_type = df_terr.groupby('AttackType').agg({'AttackType':'count','Kil
killed_wounded_attack_type.set_index('AttackType',inplace=True)
killed_wounded_attack_type = killed_wounded_attack_type.reset_index().sort_values(by='Num_
killed_wounded_attack_type.set_index('AttackType',inplace=True)
killed_wounded_attack_type
```

Out[60]:

AttackType	Num_Attacks	Killed	Wounded
Bombing/Explosion	88255	157321.0	372686.0
Armed Assault	42669	160297.0	77366.0
Assassination	19312	24920.0	13887.0
Hostage Taking (Kidnapping)	11158	24231.0	6446.0
Facility/Infrastructure Attack	10356	3642.0	3765.0
Unknown	7276	32381.0	14725.0
Unarmed Assault	1015	880.0	14027.0
Hostage Taking (Barricade Incident)	991	4478.0	3966.0
Hijacking	659	3718.0	17001.0

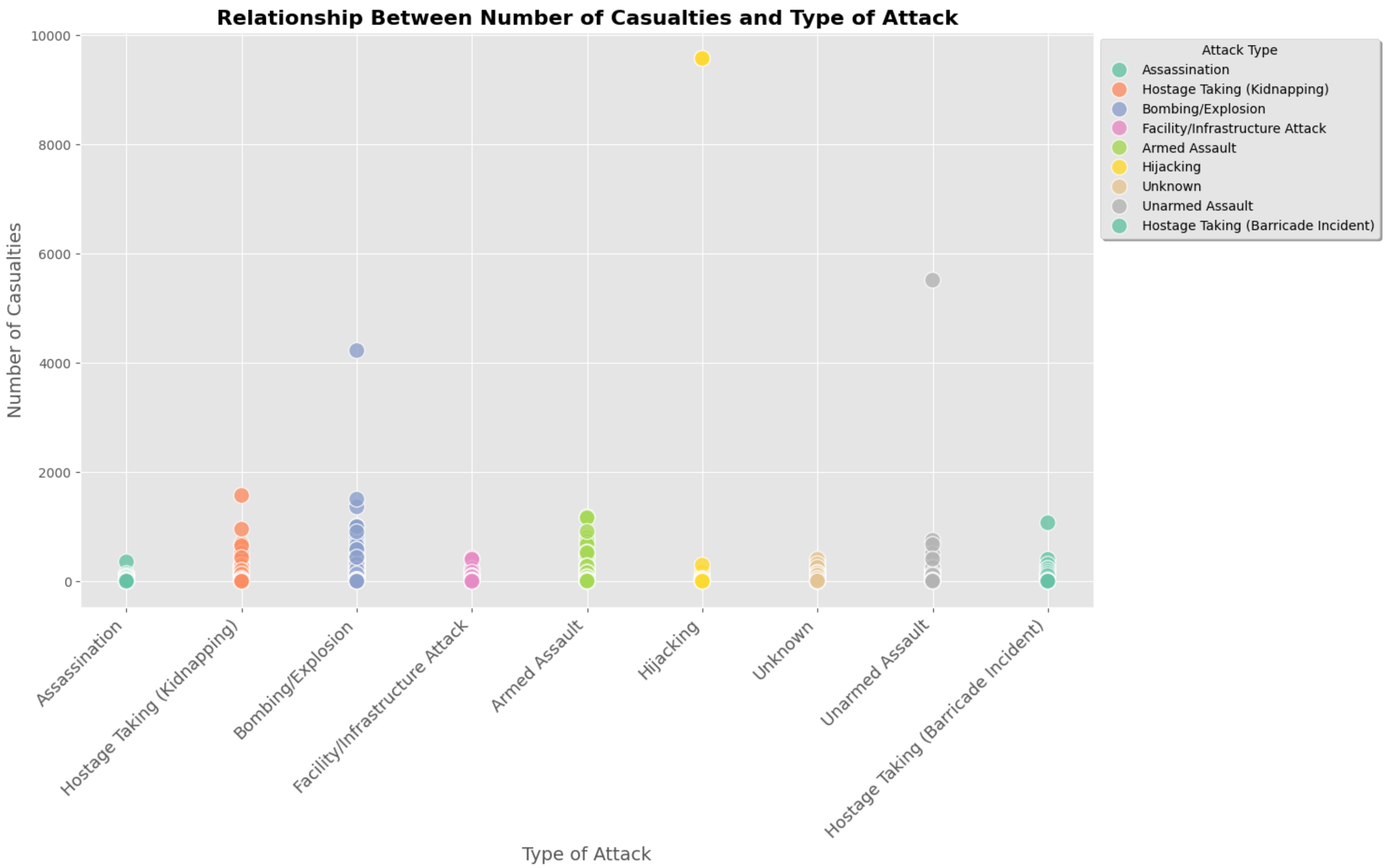
In [61]:
```
# Set the figure size
plt.figure(figsize=(15, 6))

# Create a count plot with Seaborn
sns.countplot(
    x='AttackType',
    data=df_terr,
    palette='Reds_r',
    order=df_terr['AttackType'].value_counts().index
)

# Rotate x-axis labels for better readability
plt.xticks(rotation=55,ha='right')

# Add a title to the plot
plt.title('The Most Attacking Methods by Terrorists', fontsize=16)

# Show the plot
# plt.tight_layout()  # Adjust layout to fit labels and title
plt.show()
```



to get on overall casualties killing and wounding. df_terr['casualties'] = df_terr['Killed'].fillna(0) + df_terr['Wounded'].fillna(0)

In [62]:
```
# df_terr['casualties'] = df_terr['Killed'].fillna(0) + df_terr['Wounded'].fillna(0)

# Set the style of the visualization
plt.style.use('ggplot')

# Create the scatter plot
plt.figure(figsize=(14, 8))
scatter_plot = sns.scatterplot(data=df_terr, x='AttackType', y='casualties', hue='AttackTy

# Customize the plot
scatter_plot.set_title('Relationship Between Number of Casualties and Type of Attack', for
scatter_plot.set_xlabel('Type of Attack', fontsize=14)
```

```python
scatter_plot.set_ylabel('Number of Casualties', fontsize=14)

plt.xticks(rotation=45, ha='right', fontsize=13)

scatter_plot.legend(title='Attack Type', loc='upper left', bbox_to_anchor=(1, 1), ncol=1,

# Adjust x-axis labels for better readability
# scatter_plot.set_xticklabels(scatter_plot.get_xticklabels(), rotation=45, horizontalalig

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



In [63]:

```python
killed_wounded_attack_type.index
```

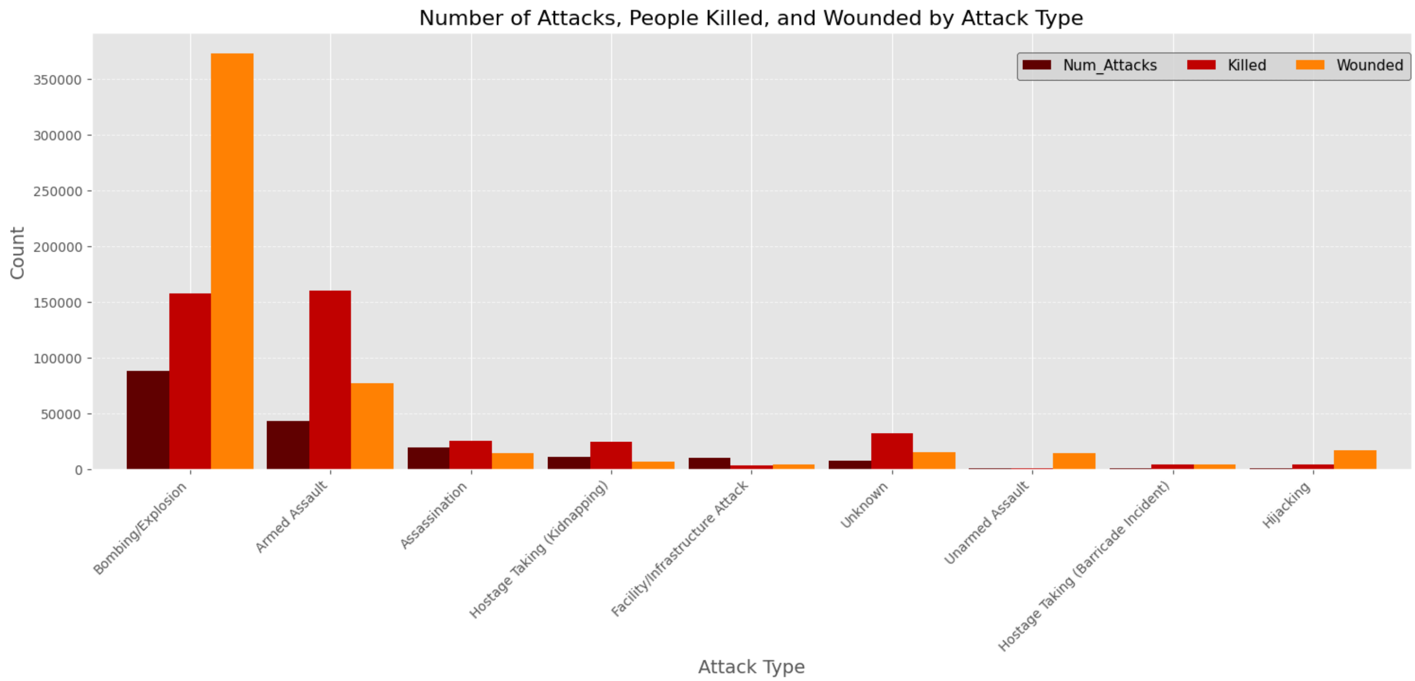
Out[63]:

```
Index(['Bombing/Explosion', 'Armed Assault', 'Assassination',
       'Hostage Taking (Kidnapping)', 'Facility/Infrastructure Attack',
       'Unknown', 'Unarmed Assault', 'Hostage Taking (Barricade Incident)',
       'Hijacking'],
      dtype='object', name='AttackType')
```

In [64]:

```python
# Use seaborn color palette
palette = sns.color_palette("gist_heat",3 )   # Choose a palette with 3 colors

# Plot the data
fig, ax = plt.subplots(figsize=(18, 6))
killed_wounded_attack_type.plot(kind='bar', width=0.9, ax=ax, color=palette)

ax.set_title('Number of Attacks, People Killed, and Wounded by Attack Type', fontsize=16)
ax.set_xlabel('Attack Type', fontsize=14)
ax.set_ylabel('Count', fontsize=14)

# Set x-axis ticks and labels
ax.set_xticklabels(killed_wounded_attack_type.index, rotation=45, ha='right')
```

```
# Add legend
legend = ax.legend(loc='upper center', bbox_to_anchor=(0.85, .97), fontsize=11, ncol=3, fr
frame = legend.get_frame()
frame.set_facecolor('lightgrey')  # Set the background color of the legend box
frame.set_edgecolor('black')

# Add gridlines for better readability
ax.grid(axis='y', linestyle='--', linewidth=0.7, alpha=0.7)

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



Now let us check out which Terrorist organizations have carried out their operations in each country.

In [65]:

```
((df_terr.value_counts('Group')/len(df_terr))*100).reset_index()
```

Out[65]:

	Group	count
0	Unknown	45.561971
1	Taliban	4.115779
2	Islamic State of Iraq and the Levant (ISIL)	3.089311
3	Shining Path (SL)	2.507004
4	Farabundo Marti National Liberation Front (FMLN)	1.844340
...	...	...
3532	Jaish al-Muhajireen wal-Ansar (Muhajireen Army)	0.000550
3533	Jaish al-Islam (Libya)	0.000550
3534	Jaish Tahkim al-Din	0.000550
3535	Jaish Al-Umma (Army of the Nation)	0.000550

	Group	count
3536	leftist guerrillas-Bolivarian militia	0.000550

3537 rows × 2 columns

By looking at the data and performing some mathematical procedures, we find that `unknown data` represents `45%` of the total data that we have, and we will work to solve it.

In [66]:

```
df_terr.loc[df_terr['Group'] == 'Unknown', 'Group'] = 'Israel Security Intelligence Servic
```

In [67]:

```
df_terr['Group'].value_counts().to_frame()
```

Out[67]:

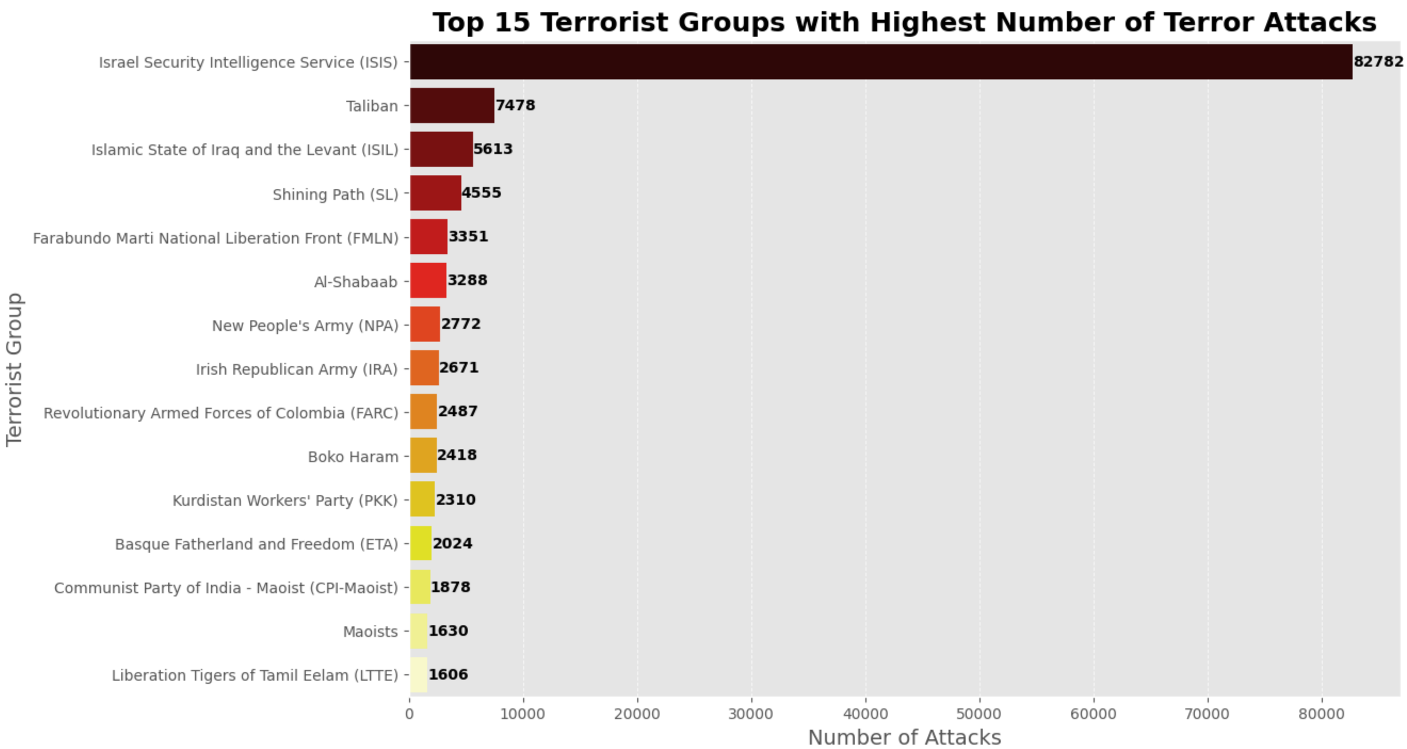
Group	count
Israel Security Intelligence Service (ISIS)	82782
Taliban	7478
Islamic State of Iraq and the Levant (ISIL)	5613
Shining Path (SL)	4555
Farabundo Marti National Liberation Front (FMLN)	3351
...	...
Ansar Sarallah	1
Sword of Islam	1
Support of Ocalan-The Hawks of Thrace	1
Arab Revolutionary Front	1
MANO-D	1

3537 rows × 1 columns

In [68]:

```
palette = sns.color_palette("hot",15)

# Create the plot
plt.figure(figsize=(12, 8))
bar_plot = sns.barplot(x=df_terr['Group'].value_counts()[:15].values, y=df_terr['Group'].v

# Enhance the plot with annotations
for p in bar_plot.patches:
    width = p.get_width()
    bar_plot.text(width + 0.5, p.get_y() + p.get_height() / 2, f'{int(width)}',
                  va='center', ha='left', fontsize=10, color='black', weight='bold')

# Set plot title and labels
plt.title('Top 15 Terrorist Groups with Highest Number of Terror Attacks', fontsize=18, we
plt.xlabel('Number of Attacks', fontsize=14)
plt.ylabel('Terrorist Group', fontsize=14)
```

```
# Add gridlines for better readability
plt.grid(axis='x', linestyle='--', linewidth=0.7, alpha=0.7)

# Adjust layout for better visualization
# plt.tight_layout()

# Show the plot
plt.show()
```



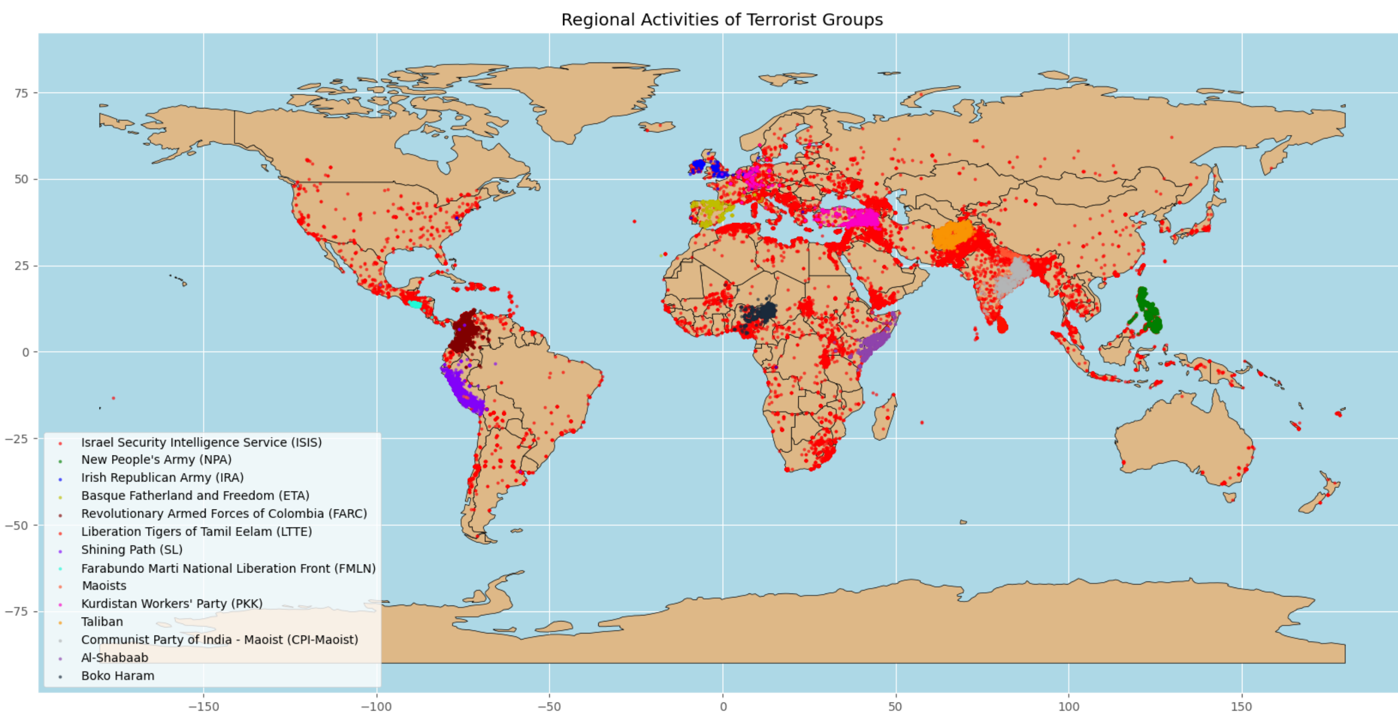
In [69]:

```
# If you don't have geopandas installed, uncomment the line below to install it:
# pip install geopandas
```

In [70]:

```
# its okay
import geopandas as gpd
from shapely.geometry import Point

# Ensure your df_terr DataFrame has 'longitude' and 'latitude' columns
df_terr['Coordinates'] = df_terr.apply(lambda row: Point(row['longitude'], row['latitude']

# Create a GeoDataFrame
gdf_terr = gpd.GeoDataFrame(df_terr, geometry='Coordinates')

# Filter top 15 groups by number of attacks
top_groups = df_terr[df_terr['Group'].isin(df_terr['Group'].value_counts()[:15].index)]

# Create a GeoDataFrame for top groups
gdf_top_groups = gpd.GeoDataFrame(top_groups, geometry='Coordinates')

# the world map shapefile ()
#-----------------------------------------------------read it its important -------------

# To run it you must download it (Admin 0 - Countries) from [https://www.naturalearthdata.
# i provided it to you if u run it from jupiter
#---can you run it from colab without using package just uncomment the line below to run i
# but comment next line ok :)
# world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))

world = gpd.read_file("ne_110m_admin_0_countries\\ne_110m_admin_0_countries.shp")
```

```
# Plot the map
fig, ax = plt.subplots(figsize=(22, 10))
ax.patch.set_facecolor('lightblue')

world.plot(ax=ax, color='burlywood', edgecolor='black')

# Define colors and groups
colors = ['r', 'g', 'b', 'y', '#800000', '#ff1100', '#8202fa', '#20fad9', '#ff5733', '#fa
groups = list(gdf_top_groups['Group'].unique())

# Plot points for each group
for group, color in zip(groups, colors):
    group_data = gdf_top_groups[gdf_top_groups['Group'] == group]
    group_data.plot(ax=ax, marker='o', color=color, markersize=5, label=group, alpha=0.6)

# Add legend
legend = plt.legend(loc='lower left', frameon=True, prop={'size': 10})
frame = legend.get_frame()
frame.set_facecolor('white')

# Set plot title
plt.title('Regional Activities of Terrorist Groups')

# Show the plot
plt.show()
```



In [71]:
```
df_terr.Target_type.head(10)
```

Out[71]:
```
0    Private Citizens & Property
1        Government (Diplomatic)
2            Journalists & Media
3        Government (Diplomatic)
4        Government (Diplomatic)
5                         Police
6                         Police
7                      Utilities
8                       Military
9           Government (General)
Name: Target_type, dtype: object
```

In [72]:

```
(df_terr.Target_type.value_counts()/len(df_terr))*100
```

Out[72]:

```
Target_type
Private Citizens & Property       23.947801
Military                          15.401974
Police                            13.487735
Government (General)              11.713844
Business                          11.375907
Transportation                     3.742068
Utilities                          3.314969
Unknown                            3.246171
Religious Figures/Institutions     2.443709
Educational Institution            2.378764
Government (Diplomatic)            1.966526
Terrorists/Non-State Militia       1.672620
Journalists & Media                1.622535
Violent Political Party            1.027018
Airports & Aircraft                0.739167
Telecommunication                  0.555338
NGO                                0.533873
Tourists                           0.242169
Maritime                           0.193185
Food or Water Supply               0.174472
Abortion Related                   0.144751
Other                              0.075403
Name: count, dtype: float64
```

Looking at the `unknown data`, it represents a `3.24%` percentage overall data that we can ignore without effect on data.

In [73]:

```
# Filter out rows where 'Target_type' is 'Unknown'
Target_type_filter = df_terr[df_terr['Target_type'] != 'Unknown']
Target_type_filter.head(10)
```

Out[73]:

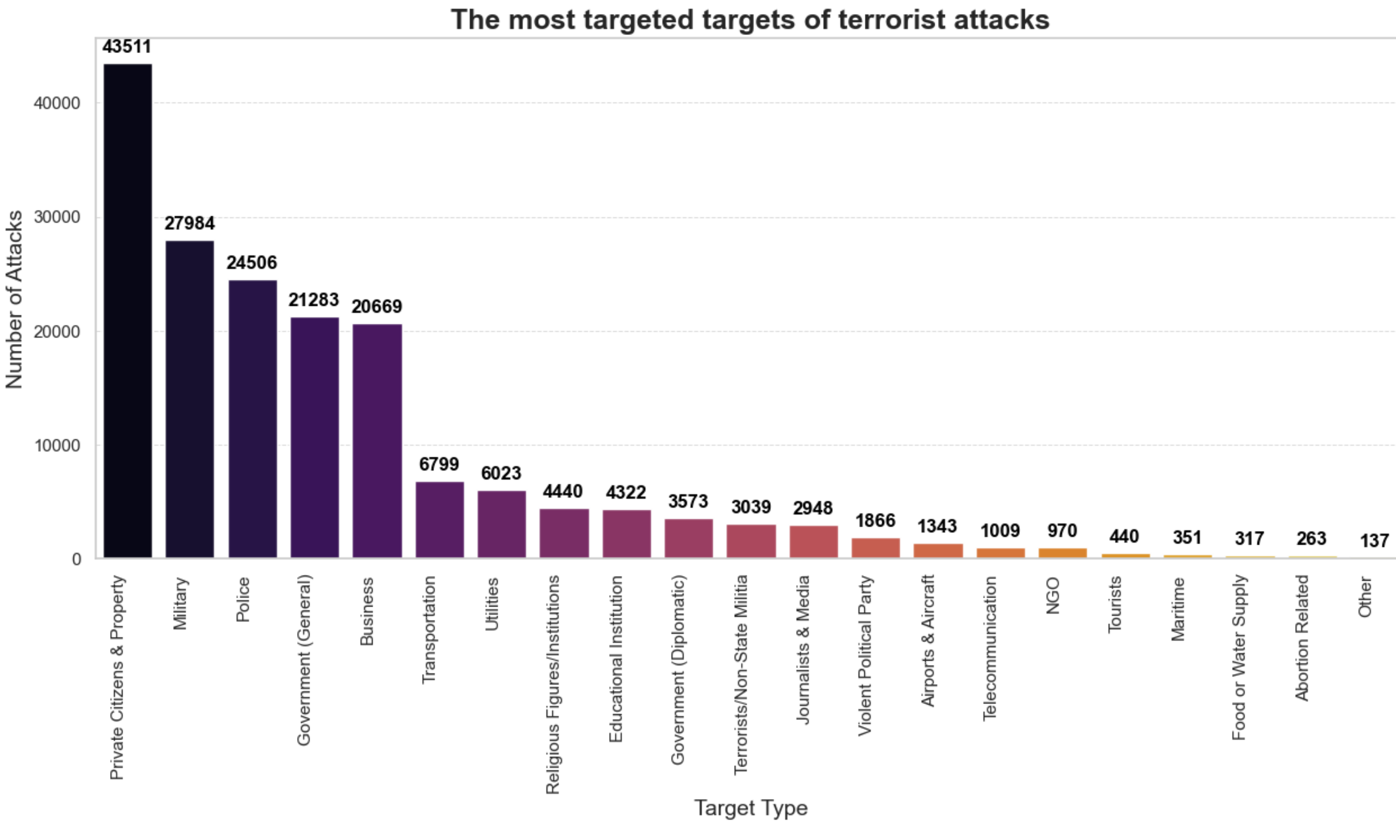
	id	Year	Month	Day	Country	Region	State	city	latitude	longitude	
0	197000000001	1970	7	2	Dominican Republic	Central America & Caribbean	NaN	Santo Domingo	18.456792	-69.951164	
1	197000000002	1970	0	0	Mexico	North America	Federal	Mexico city	19.371887	-99.086624	H
2	197001000001	1970	1	0	Philippines	Southeast Asia	Tarlac	Unknown	15.478598	120.599741	
3	197001000002	1970	1	0	Greece	Western Europe	Attica	Athens	37.997490	23.762728	Boml

	id	Year	Month	Day	Country	Region	State	city	latitude	longitude	
4	197001000003	1970	1	0	Japan	East Asia	Fukouka	Fukouka	33.580412	130.396361	Facility
5	197001010002	1970	1	1	United States	North America	Illinois	Cairo	37.005105	-89.176269	
6	197001020001	1970	1	2	Uruguay	South America	Montevideo	Montevideo	-34.891151	-56.187214	
7	197001020002	1970	1	2	United States	North America	California	Oakland	37.791927	-122.225906	Boml
8	197001020003	1970	1	2	United States	North America	Wisconsin	Madison	43.076592	-89.412488	Facility
9	197001030001	1970	1	3	United States	North America	Wisconsin	Madison	43.072950	-89.386694	Facility

In [74]:

```
# Set the size and style of the plot
plt.figure(figsize=(15, 6))
sns.set(style="whitegrid")  # Set the style to 'whitegrid' for a cleaner background

# Create the countplot
ax = sns.countplot(
    x='Target_type',
    data=Target_type_filter,
    palette='inferno',
    order=Target_type_filter['Target_type'].value_counts().index
)

# Add annotations on top of the bars
for p in ax.patches:
    height = p.get_height()
    ax.annotate(f'{int(height)}',
                (p.get_x() + p.get_width() / 2., height),
                ha='center',
                va='center',
                xytext=(0, 10),
                textcoords='offset points',
                fontsize=12,
                weight='bold',
                color='black')
```

```python
# Rotate x-axis labels for better readability
plt.xticks(rotation=90, ha='right')

# Set plot title and labels
plt.title('The most targeted targets of terrorist attacks', fontsize=18, weight='bold')
plt.xlabel('Target Type', fontsize=14)
plt.ylabel('Number of Attacks', fontsize=14)

# Add gridlines for y-axis
plt.grid(axis='y', linestyle='--', linewidth=0.7, alpha=0.7)

# Adjust layout for better spacing
# plt.tight_layout()

# Show the plot
plt.show()
```



Pillow # imagemagick # !apt-get install imagemagick # !conda install -c conda-forge imagemagick

In [75]:
```python
# If you don't have cartopy installed, uncomment the line below to install it:
# !pip install cartopy
```

Animation shows the spread of terrorist activities in the country over the past years.

In [76]:
```python
import matplotlib.animation as animation
import cartopy.crs as ccrs
import cartopy.feature as cfeature
import base64
import io
from IPython.display import HTML

# Sample data creation (replace with your actual DataFrame)
```

```
# Assuming df_terr has columns 'Year', 'latitude', 'longitude', 'Killed', 'Wounded'
data = df_terr.copy()  # Create a copy to avoid modifying the original DataFrame
data['casualties'] = data['Killed'] + data['Wounded']  # Combine casualties

# Create a figure with Cartopy projection
fig, ax = plt.subplots(figsize=(10, 6), subplot_kw={'projection': ccrs.PlateCarree()})

# Add land and ocean with specified colors
ax.add_feature(cfeature.LAND, facecolor='burlywood')
ax.add_feature(cfeature.OCEAN, facecolor='lightblue')

# Initialize scatter object to None outside the function
scatter = None

def animate(year):
    global scatter  # Use global scatter object

    # Clear the axis to redraw
    ax.clear()

    # Add land and ocean with specified colors again after clearing
#     ax.add_feature(cfeature.LAND, facecolor='burlywood')
#     ax.add_feature(cfeature.OCEAN, facecolor='lightblue')
    ax.add_feature(cfeature.BORDERS, linestyle='--')
    ax.coastlines()

    # Filter data for the current year
    year_data = data[data['Year'] == year]

    # Create or update scatter plot data
    scatter = ax.scatter(
        year_data['longitude'],
        year_data['latitude'],
        s=year_data['casualties'] * 0.1,  # Adjust marker size based on casualties
        color='red',
        alpha=0.7,
        transform=ccrs.PlateCarree()  # Use PlateCarree projection
    )

    # Update title
    ax.set_title(f'Animation of Attack Terrorist Activities on The Country\nYear: {year}',
    ax.set_global()

    return scatter

# Create the animation
ani = animation.FuncAnimation(fig, animate, frames=sorted(data['Year'].unique()), interval

# Save the animation as a gif
ani.save('animation.gif', writer='imagemagick', fps=1)
plt.close()

# Display the gif in a Jupyter notebook
filename = 'animation.gif'
video = io.open(filename, 'r+b').read()
encoded = base64.b64encode(video)
HTML(data=f'<img src="data:image/gif;base64,{encoded.decode("ascii")}" type="gif" />')
```

```
MovieWriter imagemagick unavailable; using Pillow instead.
```

Out[76]:

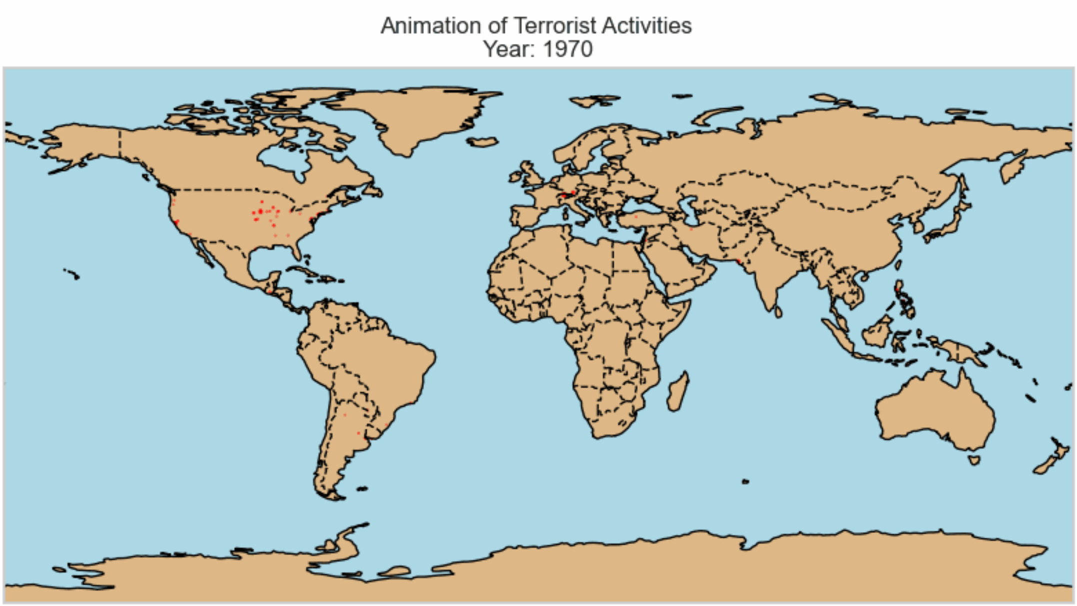
In [77]:

```
import matplotlib.animation as animation
import cartopy.crs as ccrs
import cartopy.feature as cfeature
import base64
import io
from IPython.display import HTML

# Sample data creation (replace with your actual DataFrame)
# Assuming df_terr has columns 'Year', 'latitude', 'longitude', 'Killed', 'Wounded'
data = df_terr.copy()  # Create a copy to avoid modifying the original DataFrame
data['casualties'] = data['Killed'] + data['Wounded']  # Combine casualties

# Create a figure with Cartopy projection
fig, ax = plt.subplots(figsize=(10, 6), subplot_kw={'projection': ccrs.PlateCarree()})

# Add land and ocean with specified colors
ax.add_feature(cfeature.LAND, facecolor='burlywood')
ax.add_feature(cfeature.OCEAN, facecolor='lightblue')

# Initialize scatter object to None outside the function
scatter = None

def animate(year):
    global scatter  # Use global scatter object

    # Clear the axis to redraw
    ax.clear()

    # Add land and ocean with specified colors again after clearing
    ax.add_feature(cfeature.LAND, facecolor='burlywood')
    ax.add_feature(cfeature.OCEAN, facecolor='lightblue')
    ax.add_feature(cfeature.BORDERS, linestyle='--')
    ax.coastlines()

    # Filter data for the current year
    year_data = data[data['Year'] == year]
```

```
    # Create or update scatter plot data
    scatter = ax.scatter(
        year_data['longitude'],
        year_data['latitude'],
        s=year_data['casualties'] * 0.1,  # Adjust marker size based on casualties
        color='red',
        alpha=0.7,
        transform=ccrs.PlateCarree()  # Use PlateCarree projection
    )

    # Update title
    ax.set_title(f'Animation of Terrorist Activities each Country \nYear: {year}', fontsiz
    ax.set_global()

    return scatter

# Create the animation
ani = animation.FuncAnimation(fig, animate, frames=sorted(data['Year'].unique()), interval

# Save the animation as a gif
ani.save('animation.gif', writer='imagemagick', fps=1)
plt.close()

# Display the gif in a Jupyter notebook
filename = 'animation.gif'
video = io.open(filename, 'r+b').read()
encoded = base64.b64encode(video)
HTML(data=f'<img src="data:image/gif;base64,{encoded.decode("ascii")}" type="gif" />')
```

```
MovieWriter imagemagick unavailable; using Pillow instead.
```

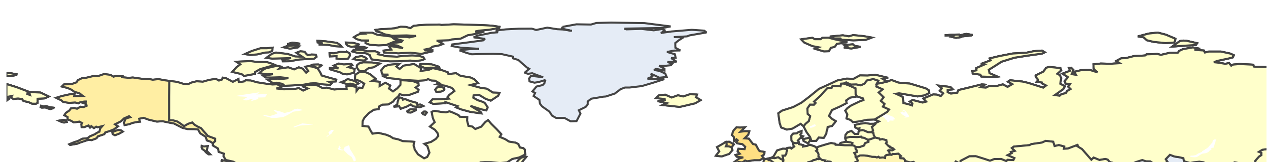
Out[77]:

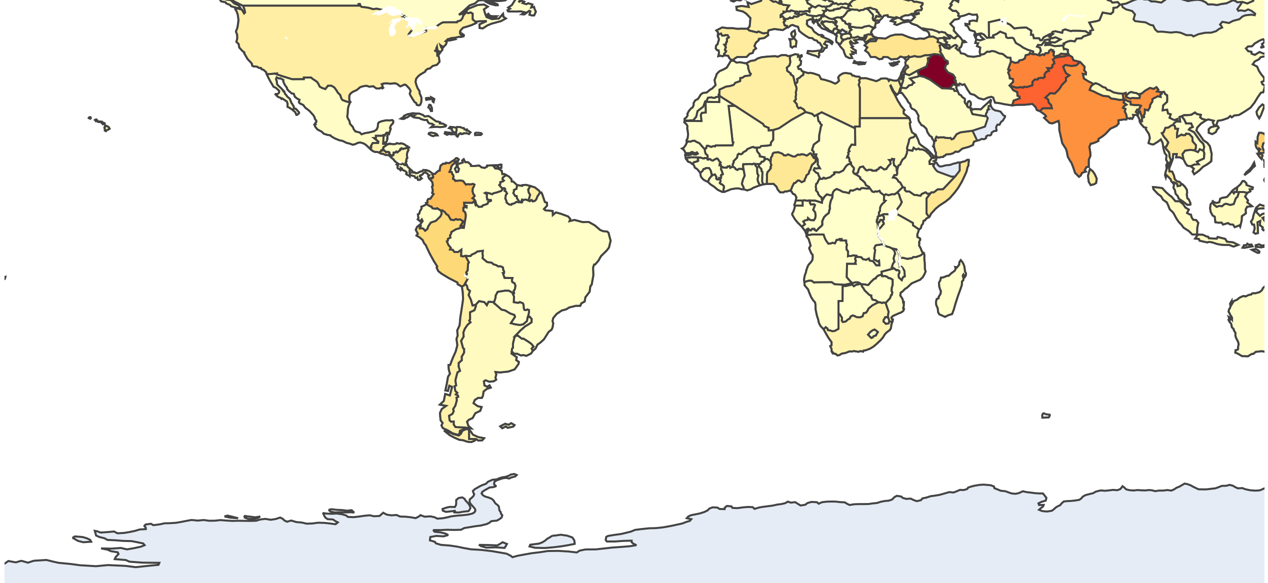


Distribution showing the total terrorist attacks by country during the past years and the most affected countries.

In [78]:
```python
import plotly.express as px

# Group the data by country and count the number of attacks
country_counts = df_terr['Country'].value_counts().reset_index()
country_counts.columns = ['Country', 'Attack Count']

# Create the choropleth map
fig = px.choropleth(
    country_counts,
    locations='Country',
    locationmode='country names',
    color='Attack Count',
    title='Distrbution Terrorist Attacks by Country',
    labels={'Attack Count': 'Number of Attacks'},
    hover_name='Country',
    color_continuous_scale='YlOrRd'
)

# Customize the color scale
fig.update_layout(
    title=dict(
        text='Distrbution Terrorist Attacks by Country Over The Past Years',
        font=dict(size=24, family='Arial', color='black'),
        x=0.5,
        xanchor='center'
    ),
    geo=dict(
      showframe=False,
        showcoastlines=True,
        projection_type='equirectangular',
        # center=dict(lat=0, lon=0),  # Center the map on a specific latitude and longitud
        # projection_scale=5,
    ),
    coloraxis_colorbar=dict(
        title='Number of Attacks',
        ticks='outside',
        ticklen=5,
        tickcolor='black',
        showticksuffix='all'
    ),
      # paper_bgcolor='lightgrey'  # Change the background color of the entire figure
    # plot_bgcolor='black'   # Change the background color of the plot area
     width=1000,  # Width of the figure in pixels
    height=600  # Height of the figure in pixels
)

# Add a hover template
fig.update_traces(
    hovertemplate='<b>%{hovertext}</b><br>Number of Attacks: %{z}<extra></extra>'
)

fig.show()
```

Distrbution Terrorist Attacks by Country Over The I





-------------------> Performance Comparison with Dask <-------------------------

- Demonstrate how to use Dask to perform similar operations with large datasets.
- Compare the performance and memory usage of Dask operations with Pandas.

!python -m pip install "dask[dataframe]" --upgrade!pip install --upgrade dask

In [79]:

```
import dask.dataframe as dd
import time
#dask
dtype={'approxdate': 'object',
       'attacktype2_txt': 'object',
       'attacktype3_txt': 'object',
       'claimmode2_txt': 'object',
       'claimmode3_txt': 'object',
       'corp2': 'object',
       'corp3': 'object',
       'divert': 'object',
       'doubtterr': 'float64',
       'gname2': 'object',
       'gname3': 'object',
       'gsubname': 'object',
       'gsubname2': 'object',
       'gsubname3': 'object',
       'hostkidoutcome_txt': 'object',
       'multiple': 'float64',
       'natlty1': 'float64',
       'natlty2_txt': 'object',
       'natlty3_txt': 'object',
       'ransom': 'float64',
       'ransomnote': 'object',
       'related': 'object',
       'target2': 'object',
       'target3': 'object',
```

```
        'targsubtype1': 'float64',
        'targsubtype2_txt': 'object',
        'targsubtype3_txt': 'object',
        'targtype2_txt': 'object',
        'targtype3_txt': 'object',
        'weapsubtype2_txt': 'object',
        'weapsubtype3_txt': 'object',
        'weaptype2_txt': 'object',
        'weaptype3_txt': 'object',
        'guncertain1': 'float64',
        'ishostkid': 'float64',
        'resolution': 'object',
        'specificity': 'float64',
        'weapsubtype4_txt': 'object',
        'weaptype4_txt': 'object'}
start_time = time.time()
df_dd = dd.read_csv('globalterrorismdb_0718dist.csv', encoding='ISO-8859-1', low_memory=Tr
# low_memory=True // helps manage memory usage by processing files in chunks and is usefu

dask_duration = time.time() - start_time

print(f"Time taken to read CSV into Dask DataFrame: {dask_duration:.2f} seconds")
```

```
Time taken to read CSV into Dask DataFrame: 0.61 seconds
```

In [80]:

```
# [persist()] This method computes the DataFrame and caches it in memory, which can speed
df_dd = df_dd.persist()    # Persist the DataFrame memory to cache it

df_dd = df_dd.compute()   # Compute the DataFrame on the cluster
df_dd.head()
```

Out[80]:

	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	country_txt	region	region_txt
0	197000000001	1970	7	2	NaN	0	NaN	58	Dominican Republic	2	Central America & Caribbean
1	197000000002	1970	0	0	NaN	0	NaN	130	Mexico	1	North America
2	197001000001	1970	1	0	NaN	0	NaN	160	Philippines	5	Southeast Asia
3	197001000002	1970	1	0	NaN	0	NaN	78	Greece	8	Western Europe
4	197001000003	1970	1	0	NaN	0	NaN	101	Japan	4	East Asia

In [81]:

```
df_dd.describe()
```

Out[81]:

	eventid	iyear	imonth	iday	extended	country	region	
count	1.816910e+05	181691.000000	181691.000000	181691.000000	181691.000000	181691.000000	181691.000000	17
mean	2.002705e+11	2002.638997	6.467277	15.505644	0.045346	131.968501	7.160938	
std	1.325957e+09	13.259430	3.388303	8.814045	0.208063	112.414535	2.933408	
min	1.970000e+11	1970.000000	0.000000	0.000000	0.000000	4.000000	1.000000	
25%	1.991021e+11	1991.000000	4.000000	8.000000	0.000000	78.000000	5.000000	

	eventid	iyear	imonth	iday	extended	country	region
50%	2.009022e+11	2009.000000	6.000000	15.000000	0.000000	98.000000	6.000000
75%	2.014081e+11	2014.000000	9.000000	23.000000	0.000000	160.000000	10.000000
max	2.017123e+11	2017.000000	12.000000	31.000000	1.000000	1004.000000	12.000000

In [82]:

```
df_dd.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 181691 entries, 0 to 74206
Columns: 135 entries, eventid to related
dtypes: float64(55), int64(22), object(58)
memory usage: 188.5+ MB
```

In [83]:

```
df_dd
```

Out[83]:

	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	country_txt	region	region_
0	197000000001	1970	7	2	NaN	0	NaN	58	Dominican Republic	2	Cent America Caribbe
1	197000000002	1970	0	0	NaN	0	NaN	130	Mexico	1	No Amer
2	197001000001	1970	1	0	NaN	0	NaN	160	Philippines	5	Southe A
3	197001000002	1970	1	0	NaN	0	NaN	78	Greece	8	Weste Euro
4	197001000003	1970	1	0	NaN	0	NaN	101	Japan	4	East A
...	...	...	...	...	...	...	...	...	...	...	
74202	201712310022	2017	12	31	NaN	0	NaN	182	Somalia	11	Su Sahar Afr
74203	201712310029	2017	12	31	NaN	0	NaN	200	Syria	10	Mid East No Afr
74204	201712310030	2017	12	31	NaN	0	NaN	160	Philippines	5	Southe A
74205	201712310031	2017	12	31	NaN	0	NaN	92	India	6	South A

	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	country_txt	region	region_
74206	201712310032	2017	12	31	NaN	0	NaN	160	Philippines	5	Southe A

181691 rows × 135 columns

In [84]:

```
df_dd["country_txt"].value_counts()
```

Out[84]:

```
country_txt
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
Lebanon                              2478
Chile                                2365
Libya                                2249
West Bank and Gaza Strip             2227
Syria                                2201
Russia                               2194
Israel                               2183
Guatemala                            2050
South Africa                         2016
Nicaragua                            1970
Ukraine                              1709
Bangladesh                           1648
Italy                                1565
Greece                               1275
Nepal                                1215
Sudan                                 967
Argentina                             815
Democratic Republic of the Congo      775
Indonesia                             761
Germany                               735
Iran                                  684
Kenya                                 683
Burundi                               613
Mali                                  566
Myanmar                               546
West Germany (FRG)                    541
Mexico                                524
Angola                                499
```

Japan	402
Uganda	394
Saudi Arabia	371
Mozambique	363
Cameroon	332
Honduras	323
Bolivia	314
Ireland	307
Venezuela	293
Central African Republic	283
Brazil	273
Cambodia	259
China	252
South Sudan	225
Ecuador	220
Georgia	217
Haiti	213
Bahrain	207
Yugoslavia	203
Kosovo	196
Ethiopia	190
Tajikistan	188
Bosnia-Herzegovina	159
Rwanda	159
Niger	154
Belgium	154
Namibia	151
Portugal	140
Sweden	132
Cyprus	132
Netherlands	130
Panama	127
Senegal	118
Macedonia	118
Austria	115
Australia	114
Paraguay	114
Jordan	113
Switzerland	111
Tunisia	109
Zimbabwe	101
Sierra Leone	101
Malaysia	99
Canada	96
Chad	91
Dominican Republic	90
Papua New Guinea	89
Rhodesia	83
Uruguay	82
Albania	80
Soviet Union	78
Kuwait	76
Ivory Coast	74
Costa Rica	67
Suriname	66
Zambia	62
Tanzania	59
Croatia	57
Guadeloupe	56
Bulgaria	52
Burkina Faso	52
Zaire	50
Taiwan	50
Azerbaijan	49
Togo	48
Hungary	46

Denmark	41
Poland	39
South Korea	38
East Germany (GDR)	38
Morocco	36
Jamaica	36
Republic of the Congo	36
Kyrgyzstan	35
Liberia	34
Macau	33
Czech Republic	32
New Caledonia	31
Cuba	30
Lesotho	29
Kazakhstan	27
Madagascar	27
Laos	27
Guyana	26
Hong Kong	26
Guinea	25
Armenia	24
Malta	23
Maldives	22
United Arab Emirates	22
Trinidad and Tobago	22
Djibouti	22
Uzbekistan	21
Moldova	21
Finland	20
New Zealand	20
Ghana	19
Norway	19
Mauritania	18
Slovak Republic	18
Fiji	17
Latvia	17
Swaziland	16
Estonia	16
Luxembourg	16
Belarus	13
Vietnam	12
Martinique	12
Serbia	12
Serbia-Montenegro	11
East Timor	10
Botswana	10
Eritrea	10
Czechoslovakia	10
Guinea-Bissau	9
Gabon	8
Lithuania	8
Benin	8
Belize	8
Qatar	7
Singapore	7
French Guiana	7
Romania	6
Bhutan	6
North Yemen	6
Slovenia	6
Brunei	6
Grenada	5
Bahamas	5
Montenegro	5
Western Sahara	5
Comoros	5

```
Malawi                                  5
People's Republic of the Congo          4
Iceland                                 4
Solomon Islands                         4
Gambia                                  3
Dominica                                3
French Polynesia                        3
Barbados                                3
Vanuatu                                 2
Turkmenistan                            2
Seychelles                              2
Mauritius                               2
St. Kitts and Nevis                     2
Equatorial Guinea                       2
South Yemen                             2
Vatican City                            1
Falkland Islands                        1
St. Lucia                               1
North Korea                             1
New Hebrides                            1
International                           1
Wallis and Futuna                       1
South Vietnam                           1
Andorra                                 1
Antigua and Barbuda                     1
Name: count, dtype: int64
```

In [85]:

```
df_dd.groupby("country_txt").count().sort_values(by="nkill",ascending=False)
```

Out[85]:

	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	region	region_txt	pro
country_txt											
Iraq	24636	24636	24636	24636	4558	24636	83	24636	24636	24636	
Pakistan	14368	14368	14368	14368	356	14368	107	14368	14368	14368	
Afghanistan	12731	12731	12731	12731	1099	12731	102	12731	12731	12731	
India	11960	11960	11960	11960	196	11960	212	11960	11960	11960	
Colombia	8306	8306	8306	8306	182	8306	384	8306	8306	8306	
Philippines	6908	6908	6908	6908	92	6908	147	6908	6908	6908	
Peru	6096	6096	6096	6096	5	6096	46	6096	6096	6096	
United Kingdom	5235	5235	5235	5235	16	5235	13	5235	5235	5235	
Turkey	4292	4292	4292	4292	90	4292	32	4292	4292	4292	
El Salvador	5320	5320	5320	5320	0	5320	46	5320	5320	5320	
Thailand	3849	3849	3849	3849	20	3849	13	3849	3849	3849	
Nigeria	3907	3907	3907	3907	154	3907	43	3907	3907	3907	
Somalia	4142	4142	4142	4142	323	4142	49	4142	4142	4142	
Yemen	3347	3347	3347	3347	590	3347	38	3347	3347	3347	
Sri Lanka	3022	3022	3022	3022	11	3022	24	3022	3022	3022	
Spain	3249	3249	3249	3249	4	3249	41	3249	3249	3249	
United States	2836	2836	2836	2836	37	2836	12	2836	2836	2836	
Algeria	2743	2743	2743	2743	77	2743	53	2743	2743	2743	

country_txt	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	region	region_txt	pro
France	2693	2693	2693	2693	18	2693	15	2693	2693	2693	
Egypt	2479	2479	2479	2479	230	2479	0	2479	2479	2479	
Lebanon	2478	2478	2478	2478	15	2478	61	2478	2478	2478	
Russia	2194	2194	2194	2194	30	2194	51	2194	2194	2194	
West Bank and Gaza Strip	2227	2227	2227	2227	30	2227	31	2227	2227	2227	
Israel	2183	2183	2183	2183	10	2183	12	2183	2183	2183	
Libya	2249	2249	2249	2249	251	2249	0	2249	2249	2249	
Chile	2365	2365	2365	2365	4	2365	13	2365	2365	2365	
South Africa	2016	2016	2016	2016	4	2016	1	2016	2016	2016	
Syria	2201	2201	2201	2201	170	2201	3	2201	2201	2201	
Guatemala	2050	2050	2050	2050	0	2050	66	2050	2050	2050	
Bangladesh	1648	1648	1648	1648	26	1648	12	1648	1648	1648	
Ukraine	1709	1709	1709	1709	211	1709	1	1709	1709	1709	
Italy	1565	1565	1565	1565	1	1565	29	1565	1565	1565	
Nicaragua	1970	1970	1970	1970	0	1970	21	1970	1970	1970	
Greece	1275	1275	1275	1275	7	1275	7	1275	1275	1275	
Nepal	1215	1215	1215	1215	29	1215	21	1215	1215	1215	
Sudan	967	967	967	967	43	967	25	967	967	967	
Indonesia	761	761	761	761	7	761	9	761	761	761	
Germany	735	735	735	735	10	735	7	735	735	735	
Argentina	815	815	815	815	1	815	39	815	815	815	
Kenya	683	683	683	683	24	683	6	683	683	683	
Democratic Republic of the Congo	775	775	775	775	70	775	16	775	775	775	
Burundi	613	613	613	613	5	613	3	613	613	613	
Iran	684	684	684	684	2	684	7	684	684	684	
Mali	566	566	566	566	15	566	5	566	566	566	
Myanmar	546	546	546	546	20	546	3	546	546	546	
Mexico	524	524	524	524	3	524	27	524	524	524	
Angola	499	499	499	499	3	499	7	499	499	499	
West Germany (FRG)	541	541	541	541	0	541	4	541	541	541	
Uganda	394	394	394	394	4	394	4	394	394	394	
Japan	402	402	402	402	1	402	3	402	402	402	
Saudi Arabia	371	371	371	371	41	371	0	371	371	371	

	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	region	region_txt	prov
country_txt											
Mozambique	363	363	363	363	8	363	8	363	363	363	
Cameroon	332	332	332	332	22	332	2	332	332	332	
Ireland	307	307	307	307	6	307	8	307	307	307	
Honduras	323	323	323	323	0	323	6	323	323	323	
Bolivia	314	314	314	314	0	314	5	314	314	314	
Venezuela	293	293	293	293	2	293	20	293	293	293	
Central African Republic	283	283	283	283	20	283	0	283	283	283	
Brazil	273	273	273	273	1	273	20	273	273	273	
Cambodia	259	259	259	259	0	259	11	259	259	259	
China	252	252	252	252	1	252	4	252	252	252	
Georgia	217	217	217	217	0	217	11	217	217	217	
Haiti	213	213	213	213	0	213	1	213	213	213	
Ecuador	220	220	220	220	0	220	10	220	220	220	
Bahrain	207	207	207	207	7	207	0	207	207	207	
South Sudan	225	225	225	225	21	225	0	225	225	225	
Yugoslavia	203	203	203	203	0	203	3	203	203	203	
Kosovo	196	196	196	196	2	196	2	196	196	196	
Tajikistan	188	188	188	188	0	188	7	188	188	188	
Ethiopia	190	190	190	190	2	190	23	190	190	190	
Rwanda	159	159	159	159	0	159	2	159	159	159	
Bosnia-Herzegovina	159	159	159	159	2	159	0	159	159	159	
Niger	154	154	154	154	5	154	6	154	154	154	
Belgium	154	154	154	154	0	154	3	154	154	154	
Namibia	151	151	151	151	0	151	1	151	151	151	
Cyprus	132	132	132	132	1	132	2	132	132	132	
Sweden	132	132	132	132	1	132	1	132	132	132	
Netherlands	130	130	130	130	0	130	9	130	130	130	
Panama	127	127	127	127	1	127	1	127	127	127	
Portugal	140	140	140	140	0	140	0	140	140	140	
Senegal	118	118	118	118	0	118	1	118	118	118	
Macedonia	118	118	118	118	0	118	2	118	118	118	
Paraguay	114	114	114	114	3	114	1	114	114	114	
Australia	114	114	114	114	0	114	1	114	114	114	
Austria	115	115	115	115	2	115	3	115	115	115	

country_txt	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	region	region_txt	prov
Jordan	113	113	113	113	3	113	6	113	113	113	
Tunisia	109	109	109	109	1	109	2	109	109	109	
Switzerland	111	111	111	111	1	111	4	111	111	111	
Zimbabwe	101	101	101	101	2	101	1	101	101	101	
Sierra Leone	101	101	101	101	6	101	14	101	101	101	
Canada	96	96	96	96	1	96	2	96	96	96	
Malaysia	99	99	99	99	1	99	4	99	99	99	
Dominican Republic	90	90	90	90	0	90	3	90	90	90	
Papua New Guinea	89	89	89	89	1	89	3	89	89	89	
Chad	91	91	91	91	1	91	4	91	91	91	
Uruguay	82	82	82	82	0	82	6	82	82	82	
Soviet Union	78	78	78	78	0	78	3	78	78	78	
Albania	80	80	80	80	0	80	0	80	80	80	
Kuwait	76	76	76	76	1	76	3	76	76	76	
Ivory Coast	74	74	74	74	1	74	2	74	74	74	
Rhodesia	83	83	83	83	0	83	0	83	83	83	
Suriname	66	66	66	66	0	66	1	66	66	66	
Zambia	62	62	62	62	0	62	0	62	62	62	
Croatia	57	57	57	57	0	57	0	57	57	57	
Tanzania	59	59	59	59	4	59	1	59	59	59	
Costa Rica	67	67	67	67	0	67	0	67	67	67	
Zaire	50	50	50	50	0	50	0	50	50	50	
Taiwan	50	50	50	50	0	50	1	50	50	50	
Bulgaria	52	52	52	52	0	52	1	52	52	52	
Burkina Faso	52	52	52	52	3	52	0	52	52	52	
Togo	48	48	48	48	0	48	0	48	48	48	
Azerbaijan	49	49	49	49	1	49	1	49	49	49	
Hungary	46	46	46	46	0	46	0	46	46	46	
Guadeloupe	56	56	56	56	0	56	0	56	56	56	
Denmark	41	41	41	41	0	41	0	41	41	41	
Poland	39	39	39	39	0	39	0	39	39	39	
Jamaica	36	36	36	36	0	36	1	36	36	36	
South Korea	38	38	38	38	0	38	0	38	38	38	
Morocco	36	36	36	36	0	36	0	36	36	36	

country_txt	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	region	region_txt	prov
Republic of the Congo	36	36	36	36	1	36	0	36	36	36	
Kyrgyzstan	35	35	35	35	2	35	3	35	35	35	
Czech Republic	32	32	32	32	1	32	0	32	32	32	
Macau	33	33	33	33	0	33	0	33	33	33	
Liberia	34	34	34	34	1	34	6	34	34	34	
New Caledonia	31	31	31	31	0	31	1	31	31	31	
Cuba	30	30	30	30	0	30	0	30	30	30	
Lesotho	29	29	29	29	0	29	1	29	29	29	
Kazakhstan	27	27	27	27	0	27	1	27	27	27	
Madagascar	27	27	27	27	0	27	2	27	27	27	
Guyana	26	26	26	26	0	26	0	26	26	26	
Hong Kong	26	26	26	26	0	26	0	26	26	26	
Laos	27	27	27	27	0	27	0	27	27	27	
Armenia	24	24	24	24	0	24	1	24	24	24	
Guinea	25	25	25	25	1	25	2	25	25	25	
East Germany (GDR)	38	38	38	38	0	38	0	38	38	38	
Malta	23	23	23	23	0	23	0	23	23	23	
Maldives	22	22	22	22	0	22	1	22	22	22	
Moldova	21	21	21	21	0	21	0	21	21	21	
Uzbekistan	21	21	21	21	0	21	1	21	21	21	
United Arab Emirates	22	22	22	22	0	22	1	22	22	22	
New Zealand	20	20	20	20	1	20	1	20	20	20	
Djibouti	22	22	22	22	0	22	0	22	22	22	
Finland	20	20	20	20	0	20	0	20	20	20	
Trinidad and Tobago	22	22	22	22	1	22	0	22	22	22	
Norway	19	19	19	19	0	19	0	19	19	19	
Ghana	19	19	19	19	0	19	0	19	19	19	
Slovak Republic	18	18	18	18	0	18	0	18	18	18	
Fiji	17	17	17	17	1	17	0	17	17	17	
Mauritania	18	18	18	18	0	18	1	18	18	18	
Latvia	17	17	17	17	0	17	0	17	17	17	
Estonia	16	16	16	16	0	16	0	16	16	16	

	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	region	region_txt	prov
country_txt											
Luxembourg	16	16	16	16	0	16	0	16	16	16	
Swaziland	16	16	16	16	0	16	0	16	16	16	
Serbia	12	12	12	12	0	12	0	12	12	12	
Vietnam	12	12	12	12	0	12	0	12	12	12	
Belarus	13	13	13	13	0	13	0	13	13	13	
Serbia-Montenegro	11	11	11	11	0	11	0	11	11	11	
Czechoslovakia	10	10	10	10	0	10	0	10	10	10	
East Timor	10	10	10	10	0	10	0	10	10	10	
Guinea-Bissau	9	9	9	9	0	9	1	9	9	9	
Eritrea	10	10	10	10	1	10	1	10	10	10	
Botswana	10	10	10	10	0	10	0	10	10	10	
Gabon	8	8	8	8	0	8	0	8	8	8	
Belize	8	8	8	8	0	8	3	8	8	8	
Lithuania	8	8	8	8	0	8	0	8	8	8	
Benin	8	8	8	8	0	8	0	8	8	8	
Martinique	12	12	12	12	0	12	0	12	12	12	
Qatar	7	7	7	7	0	7	0	7	7	7	
Singapore	7	7	7	7	0	7	0	7	7	7	
Romania	6	6	6	6	0	6	0	6	6	6	
Brunei	6	6	6	6	0	6	0	6	6	6	
North Yemen	6	6	6	6	0	6	0	6	6	6	
Slovenia	6	6	6	6	0	6	0	6	6	6	
Comoros	5	5	5	5	0	5	0	5	5	5	
Malawi	5	5	5	5	0	5	0	5	5	5	
French Guiana	7	7	7	7	0	7	0	7	7	7	
Montenegro	5	5	5	5	0	5	0	5	5	5	
Bhutan	6	6	6	6	0	6	0	6	6	6	
Grenada	5	5	5	5	0	5	0	5	5	5	
Bahamas	5	5	5	5	0	5	0	5	5	5	
Iceland	4	4	4	4	0	4	0	4	4	4	
Western Sahara	5	5	5	5	0	5	1	5	5	5	
Solomon Islands	4	4	4	4	0	4	0	4	4	4	
French Polynesia	3	3	3	3	0	3	0	3	3	3	

country_txt	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	region	region_txt	pro
People's Republic of the Congo	4	4	4	4	0	4	1	4	4	4	
Gambia	3	3	3	3	0	3	0	3	3	3	
Barbados	3	3	3	3	0	3	0	3	3	3	
Vanuatu	2	2	2	2	0	2	0	2	2	2	
Seychelles	2	2	2	2	0	2	0	2	2	2	
Dominica	3	3	3	3	0	3	0	3	3	3	
Turkmenistan	2	2	2	2	0	2	0	2	2	2	
South Yemen	2	2	2	2	0	2	0	2	2	2	
St. Kitts and Nevis	2	2	2	2	0	2	0	2	2	2	
Equatorial Guinea	2	2	2	2	0	2	0	2	2	2	
Mauritius	2	2	2	2	0	2	0	2	2	2	
Vatican City	1	1	1	1	0	1	0	1	1	1	
International	1	1	1	1	0	1	0	1	1	1	
South Vietnam	1	1	1	1	0	1	0	1	1	1	
Wallis and Futuna	1	1	1	1	0	1	0	1	1	1	
North Korea	1	1	1	1	0	1	0	1	1	1	
New Hebrides	1	1	1	1	0	1	0	1	1	1	
St. Lucia	1	1	1	1	0	1	0	1	1	1	
Antigua and Barbuda	1	1	1	1	0	1	0	1	1	1	
Falkland Islands	1	1	1	1	0	1	0	1	1	1	
Andorra	1	1	1	1	0	1	0	1	1	1	

Summary !!

- Dask :>is particularly effective for handling large datasets as it processes data in chunks and executes operations lazily, meaning computations are performed only when needed. It manages large datasets efficiently by reading data in parts and parallelizing tasks, making it well-suited for datasets that exceed memory capacity.
- Pandas :> is simpler and often adequate for smaller datasets that can fit into memory, but it may encounter difficulties with larger datasets.

---

[2]Compare the performance and memory usage of Dask

operations with Pandas.**

1. Setup and Create a Large Synthetic Dataset.

In [86]:

```
import pandas as pd
import numpy as np

# Parameters
n_rows = 10**7  # Number of rows
n_folders = 200  # Number of folders

# Generate synthetic data
np.random.seed(0)  # Set a seed for reproducibility
data = {
    'folders': np.random.choice(['folders_' + str(i) for i in range(n_folders)], size=n_ro
    'files': np.random.randint(1, 1000, size=n_rows)  # Generating random integers betweer
}

# Create a DataFrame
dump_df = pd.DataFrame(data)

# Save to CSV for use with Dask
dump_df.to_csv('lrg_dataset.csv', index=False)
```

In [87]:

```
dump_df
```

Out[87]:

	folders	files
0	folders_172	909
1	folders_47	312
2	folders_117	115
3	folders_192	424
4	folders_67	678
...	...	...
9999995	folders_41	342
9999996	folders_51	591
9999997	folders_14	239
9999998	folders_39	551
9999999	folders_139	486

10000000 rows × 2 columns

import dask.dataframe as dd import time # Measure performance with Dask # Load data start_time = time.time() dump_df_dask = dd.read_csv('lrg_dataset.csv') load_time_dask = time.time() - start_time # Group by 'folders' and compute sum of 'files' start_time = time.time() df_grouped_dask = dump_df_dask.groupby('folders')['files'].sum().compute() groupby_time_dask = time.time() - start_time # Print performance results print(f"Dask Load Time: {load_time_dask:.2f} seconds") print(f"Dask Groupby Time: {groupby_time_dask:.2f} seconds") # Estimate memory usage by converting a sample to Pandas sample_size = 10000000 # Number of rows to sample sample = dump_df_dask.head(sample_size) sample_memory_usage = sample.memory_usage(deep=True).sum() estimated_memory_usage = (sample_memory_usage / sample_size) * len(dump_df_dask) # Estimate memory usage for the grouped data grouped_sample_memory_usage = df_grouped_dask.memory_usage(deep=True).sum() if isinstance(df_grouped_dask, pd.DataFrame) else df_grouped_dask.nbytes estimated_grouped_memory_usage = grouped_sample_memory_usage # Print memory usage results print(f"Estimated Dask

Memory Usage: {estimated_memory_usage / 1e6:.2f} MB") print(f"Estimated Dask Grouped Data Memory Usage: {estimated_grouped_memory_usage / 1e6:.2f} MB") import pandas as pd import time # Measure performance and memory usage with Pandas # Load data start_time = time.time() dump_df_pandas = pd.read_csv('lrg_dataset.csv') load_time_pandas = time.time() - start_time # Group by 'Country' and compute sum of 'Value' start_time = time.time() df_grouped_pandas = dump_df_pandas.groupby('folders')['files'].sum().reset_index() groupby_time_pandas = time.time() - start_time # Print results print(f"Pandas Load Time: {load_time_pandas:.2f} seconds") print(f"Pandas Groupby Time: {groupby_time_pandas:.2f} seconds") # Memory usage print(f"Pandas Memory Usage: {dump_df_pandas.memory_usage(deep=True).sum() / 1e6:.2f} MB") print(f"Pandas Grouped Data Memory Usage: {df_grouped_pandas.memory_usage(deep=True).sum() / 1e6:.2f} MB")

In [88]:

```
import dask.dataframe as dd
import os
import time

# Parameters
n_rows = 10**7  # Number of rows
n_folders = 200  # Number of folders

# Path to the dataset
file_path = 'lrg_dataset.csv'

# Timing and memory usage comparison
start_time = time.time()
# Read the CSV file into a Pandas DataFrame
df_pandas = pd.read_csv(file_path)
pandas_duration = time.time() - start_time

# Perform groupby operation with Pandas
start_time = time.time()
df_grouped_pandas = df_pandas.groupby('folders').agg({'files': 'sum'}).reset_index()
pandas_groupby_duration = time.time() - start_time

# Estimate memory usage for Pandas DataFrame
pandas_memory_usage = df_pandas.memory_usage(deep=True).sum() / 1e6
pandas_grouped_memory_usage = df_grouped_pandas.memory_usage(deep=True).sum() / 1e6

# ------------------------------------------------------------------------------------------

# Dask operations
start_time = time.time()
# Read the CSV file into a Dask DataFrame
df_dask = dd.read_csv(file_path)
dask_duration = time.time() - start_time

# Perform groupby operation with Dask
start_time = time.time()
df_grouped_dask = df_dask.groupby('folders').agg({'files': 'sum'}).compute()
dask_groupby_duration = time.time() - start_time

# Estimate memory usage for Dask DataFrame (approximation)
sample = df_dask.head(10000000)
sample_memory_usage = sample.memory_usage(deep=True).sum()
dask_memory_usage = sample_memory_usage * (len(df_dask) / len(sample)) / 1e6

# Print results
print(f"Pandas Load Duration: {pandas_duration:.2f} seconds")
print(f"Pandas Groupby Duration: {pandas_groupby_duration:.2f} seconds")
print(f"Pandas Memory Usage: {pandas_memory_usage:.2f} MB")
print(f"Pandas Grouped Data Memory Usage: {pandas_grouped_memory_usage:.2f} MB")
print("--"*50)
print(f"Dask Load Duration: {dask_duration:.2f} seconds")
print(f"Dask Groupby Duration: {dask_groupby_duration:.2f} seconds")
print(f"Dask Memory Usage (approx): {dask_memory_usage:.2f} MB")
```

```
Pandas Load Duration: 4.02 seconds
Pandas Groupby Duration: 1.05 seconds
Pandas Memory Usage: 754.50 MB
Pandas Grouped Data Memory Usage: 0.02 MB
------------------------------------------------------------------------------------------------
Dask Load Duration: 0.03 seconds
Dask Groupby Duration: 5.12 seconds
Dask Memory Usage (approx): 754.50 MB
```

The End :)