Title:
Global Terrorism Analysis Report

Topic:
Data visualization and analysis

Presented by

Amr Mohamed Yasser

Introduction

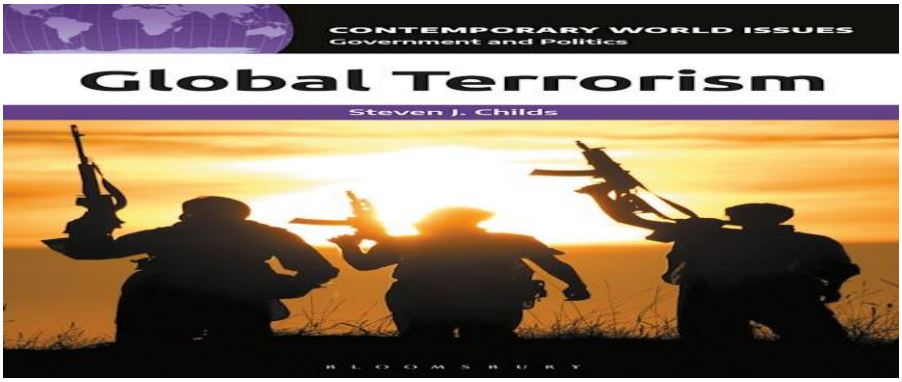
This report presents The Global Terrorism Database (GTD) is an open-source database including information on terrorist attacks around the world from 1970 through 2017. The GTD includes systematic data on domestic as well as international terrorist incidents that have occurred during this time and now includes more than 180,000 attacks. The database is maintained by researchers at the National Consortium for the Study of Terrorism and Responses to Terrorism (START), headquartered at the University of Maryland. an analysis of global data using Python libraries such as pandas, NumPy, matplotlib, and others. The data encompasses various aspects, including country statistics, number of incidents, weapon types, target types, and attack types.

Data Import and Basic Information

The dataset is imported using the pandas library, and basic information about the data structure is displayed, including the number of columns and rows.

Key Columns:

- **Country**: Represents different countries.
- **Killed**: Number of people killed.
- **Weapon Type**: Types of weapons used.
- **Target Type**: Types of targets.
- **Attack Type**: Types of attacks.



Data preparation and preprocessing:

```
df.describe()
```

	eventid	iyear	imonth	iday	extended	country	region	latitude	longitude	specificity	...	ransomamt	ransomamtus
count	1.816910e+05	181691.000000	181691.000000	181691.000000	181691.000000	181691.000000	181691.000000	177135.000000	1.771340e+05	181685.000000	...	1.350000e+03	5.630000e+02
mean	2.002705e+11	2002.638997	6.467277	15.505644	0.045346	131.968501	7.160938	23.498343	-4.586957e+02	1.451452	...	3.172530e+06	5.784865e+05
std	1.325957e+09	13.259430	3.388303	8.814045	0.208063	112.414535	2.933408	18.569242	2.047790e+05	0.995430	...	3.021157e+07	7.077924e+06
min	1.970000e+11	1970.000000	0.000000	0.000000	0.000000	4.000000	1.000000	-53.154613	-8.618590e+07	1.000000	...	-9.900000e+01	-9.900000e+01
25%	1.991021e+11	1991.000000	4.000000	8.000000	0.000000	78.000000	5.000000	11.510046	4.545640e+00	1.000000	...	0.000000e+00	0.000000e+00
50%	2.009022e+11	2009.000000	6.000000	15.000000	0.000000	98.000000	6.000000	31.467463	4.324651e+01	1.000000	...	1.500000e+04	0.000000e+00
75%	2.014081e+11	2014.000000	9.000000	23.000000	0.000000	160.000000	10.000000	34.685087	6.871033e+01	1.000000	...	4.000000e+05	0.000000e+00
max	2.017123e+11	2017.000000	12.000000	31.000000	1.000000	1004.000000	12.000000	74.633553	1.793667e+02	5.000000	...	1.000000e+09	1.320000e+08

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 181691 entries, 0 to 181690
Columns: 135 entries, eventid to related
dtypes: float64(55), int64(22), object(58)
memory usage: 187.1+ MB
```

```
n_rows,n_columns= df.shape
print(n_rows)
print(n_columns)
```

```
181691
135
```

1-Rename Columns

```
# Renaming Columns
df.rename(columns={'iyear':'Year','imonth':'Month','iday':'Day','country_txt':'Country','provstate':'state',
                   'region_txt':'Region','attacktype1_txt':'AttackType','target1':'Target','nkill':'Killed',
                   'nwound': 'Wounded','summary':'Summary','gname':'Group','targtype1_txt':'Target_type',
                   'weaptype1_txt':'Weapon_type','motive':'Motive'
                  },inplace=True)
```

```
dx = df[['eventid','Year','Month','Day','Country','Region','state','city','latitude','longitude','AttackType','Killed','Wounded',
         'Target','Summary','Group','Target_type','Weapon_type','Motive','success']]
```

2- To Identify Null Values and Handle Null Values

```
dx.isnull().sum()
```

```
eventid            0
Year               0
Month              0
Day                0
Country            0
Region             0
state            421
city             435
latitude        4556
longitude       4557
AttackType         0
Killed         10313
Wounded        16311
Target           638
Summary        66129
Group              0
Target_type        0
Weapon_type        0
Motive        131130
success            0
dtype: int64
```

```
dx['Killed'].fillna(0, inplace=True)
dx['Wounded'].fillna(0, inplace=True)
dx['state'].fillna('Unknown', inplace=True)
dx['city'].fillna('Unknown', inplace=True)
dx['Motive'].fillna('Unknown', inplace=True)
dx['Summary'].fillna('Unknown', inplace=True)
dx['Group'].fillna('Unknown', inplace=True)
dx['Target_type'].fillna('Unknown', inplace=True)
dx['Weapon_type'].fillna('Unknown', inplace=True)
dx['Target'].fillna('Unknown', inplace=True)
dx['success'].fillna(0, inplace=True)
dx['latitude'].fillna(0, inplace=True)
dx['longitude'].fillna(0, inplace=True)
```

```
dx.isna().sum()
```

```
eventid        0
Year           0
Month          0
Day            0
Country        0
Region         0
state          0
city           0
latitude       0
longitude      0
AttackType     0
Killed         0
Wounded        0
Target         0
Summary        0
Group          0
Target_type    0
Weapon_type    0
Motive         0
success        0
dtype: int64
```

Data analysis:

1- Calculate the mean, median, and standard deviation of relevant numeric column

```
Killed:
  Mean: 2.2668596683380025
  Median: 0.0
  Standard deviation: 11.227026186417314
Wounded:
  Mean: 2.8832963658078827
  Median: 0.0
  Standard deviation: 34.30965286998215
Success:
  Mean: 0.8895982739926579
  Median: 1.0
  Standard deviation: 0.31338982897015316
```

2- Group data by various categories (e.g., year, region, attack type) and calculate aggregate statistics.

```
Total Attacks per Year:
Year
1970    651
1971    471
1972    568
1973    473
1974    581
dtype: int64
Total Attacks per Region:
Region
Australasia & Oceania            282
Central America & Caribbean    10344
Central Asia                     563
East Asia                        802
Eastern Europe                  5144
dtype: int64
Total Killed per Year:
Year
1970    174.0
1971    173.0
1972    566.0
1973    370.0
1974    539.0
Total Wounded per Year:
Year
1970     212.0
1971      82.0
1972     409.0
1973     495.0
1974     865.0
Name: Wounded, dtype: float64
Total Killed per Region:
Region
Australasia & Oceania             150.0
Central America & Caribbean     28708.0
Central Asia                     1000.0
East Asia                        1152.0
Eastern Europe                   7415.0
Name: Killed, dtype: float64
Total Wounded per Region:
Region
Australasia & Oceania             260.0
Central America & Caribbean      8991.0
Central Asia                     2009.0
East Asia                        9213.0
Eastern Europe                  12045.0
Name: Wounded, dtype: float64
```

3- Identify trends over time (e.g., number of attacks per year).

```
Year
1971        471
1973        473
1972        568
1974        581
1970        651
1975        740
1976        923
1998        934
2004       1166
2003       1278
```

4- Determine the most a@ected regions and countries.

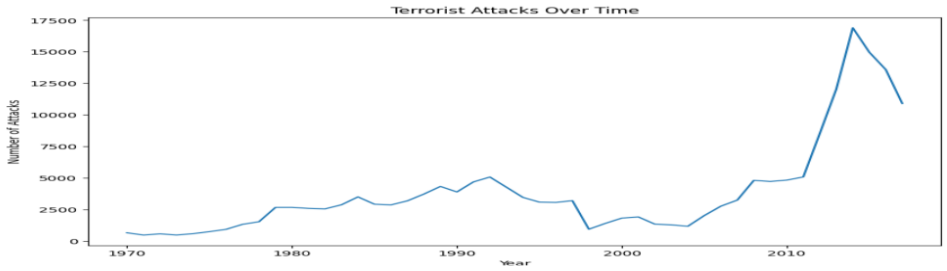
```
Most affected regions:
        Country  Number of Attacks
0          Iraq              24636
1      Pakistan              14368
2   Afghanistan              12731
3         India              11960
4      Colombia               8306
Most affected regions:
                      Region  Number of Attacks
0  Middle East & North Africa              50474
1                  South Asia              44974
2               South America              18978
3          Sub-Saharan Africa              17550
4              Western Europe              16639
```

5- identify the most common attack types and targets.

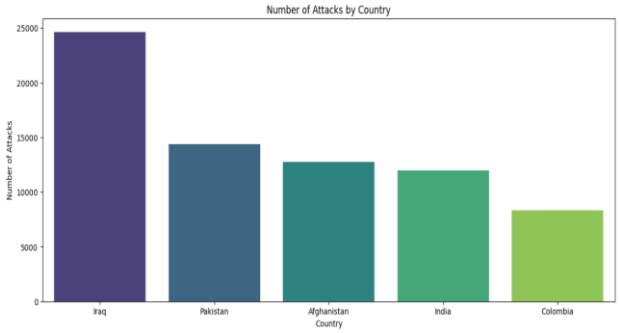
```
AttackType
Bombing/Explosion                    88255
Armed Assault                        42669
Assassination                        19312
Hostage Taking (Kidnapping)          11158
Facility/Infrastructure Attack       10356
Name: count, dtype: int64
Target
Unknown        6556
Civilians      6461
Soldiers       3157
Patrol         2942
Checkpoint     2905
```

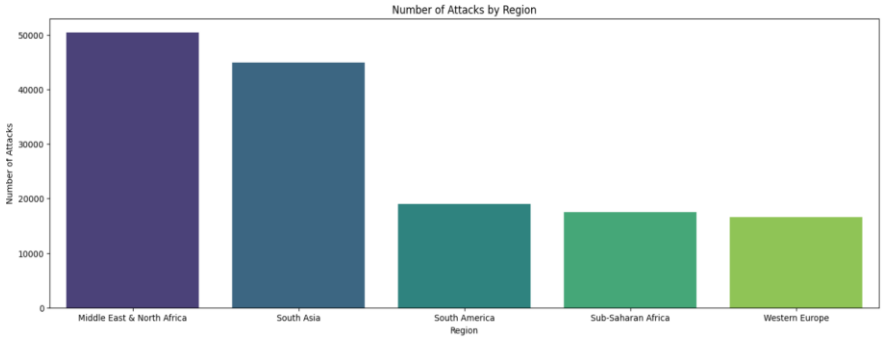
Data visualization:

1- Line plot showing the trend of terrorist attacks over the years



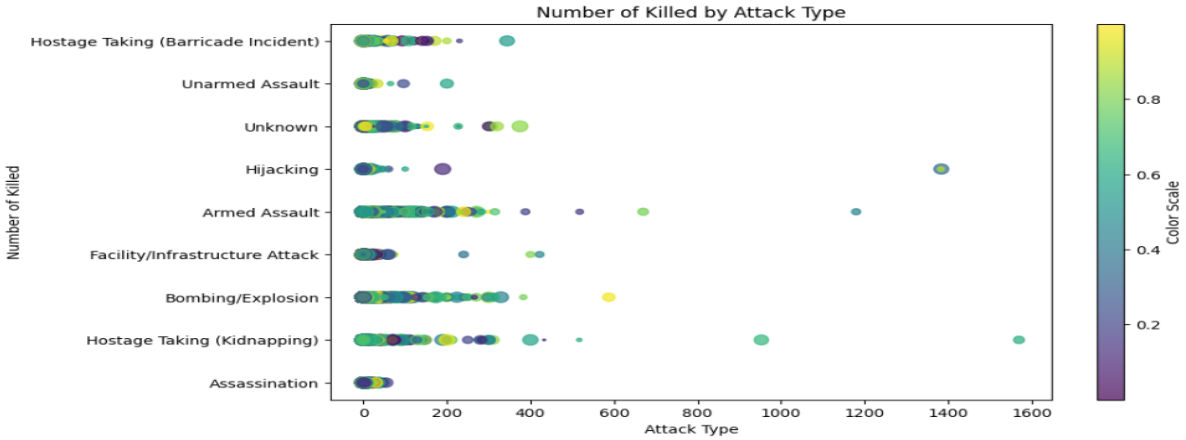
2- Bar plot of the number of attacks by region and by country.





3- Heatmap to visualize the correlation between di@erent features

4- Scatter plot showing the relationship between the number of casualties and the type of attack



Comparison between Dask and pandas

```
start_time = time.time()
import pandas as pd
test_pd = df.groupby('country').count()
endtime = time.time()
print("--- %s seconds ---" % (endtime - start_time))
import dask.dataframe as dd
dd=dd.from_pandas(df,npartitions=30)
start_time=time.time()
import pandas as pd
test_pd=dd.groupby('country').count()
endtime = time.time()
print("--- %s seconds ---" % (endtime - start_time))
```

```
--- 0.4219534397125244 seconds ---
--- 0.05566096305847168 seconds ---
```

Pandas is library to read files as csv and xml and excel and preprocessing data frames and take more time when we use to load data take time in 0.422195 Seconds

Dask is library used in bigdata and machine learning and handle large data take less time than pandas

Takes 0.055660943 Seconds

```
print("Most Country affected",df['Country'].value_counts().index[0])
print("Most City affected:",df['city'].value_counts().index[1])
print("Most Region affected:",df['Region'].value_counts().index[0])
print("Most Year affected",df['Year'].value_counts().index[0])
print("Most Month affected:",df['Month'].value_counts().index[0])
print("Most Group affected",df['Group'].value_counts().index[2])
print("Most Attack Types:",df['AttackType'].value_counts().index[0])
```

```
Most Country affected Iraq
Most City affected: Baghdad
Most Region affected: Middle East & North Africa
Most Year affected 2014
Most Month affected: 5
Most Group affected Islamic State of Iraq and the Levant (ISIL)
Most Attack Types: Bombing/Explosion
```