Global Terrorism Analysis Report

Presented by

Ahmed Elsaid Bayoumii

--Introduction

This report provides an analysis of global terrorism data using Python libraries such as pandas, NumPy, matplotlib, and others. The data used for the analysis is sourced from The Global Terrorism Database (GTD), which is an open-source database containing information on terrorist attacks worldwide from 1970 through 2017. The GTD includes data on both domestic and international terrorist incidents, totaling over 180,000 attacks.

The database is curated and maintained by researchers at the National Consortium for the Study of Terrorism and Responses to Terrorism (START), based at the University of Maryland.

The analysis explores various aspects of the data, including country statistics, the number of incidents, weapon types, target types, and attack types. The Python libraries mentioned, such as pandas, NumPy, and matplotlib, are commonly used for data manipulation, analysis, and visualization, respectively.

By leveraging these libraries, the report likely presents visualizations, statistical summaries, and insights derived from the GTD, providing a comprehensive understanding of global terrorism trends and patterns over the specified time period.

1. Data Acquisition and Preprocessing:

-Load the dataset & Handle missing values & data cleaning

bioumi_proj.ipynb

Code | Share | Open in... | Python 3 (ipykernel)

[70]:
```
df = pd.read_csv('globalterrorismdb_0718dist.csv', encoding='ISO-8859-1')
```

[71]:
```
df.head(2)
```

[71]:

	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	country_txt	region	...	addnotes	scite1	scite2	scite3	dbsource	INT_LOG	INT_
0	197000000001	1970	7	2	NaN	0	NaN	58	Dominican Republic	2	...	NaN	NaN	NaN	NaN	PGIS	0	
1	197000000002	1970	0	0	NaN	0	NaN	130	Mexico	1	...	NaN	NaN	NaN	NaN	PGIS	0	

2 rows × 135 columns

[74]:
```
df.describe()
```

[74]:

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

```
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)

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

df.head(2)
```

	Year	Month	Day	Country	state	Region	city	latitude	longitude	AttackType	Killed	Wounded	Target	Summary	Group	Target_type
0	1970	7	2	Dominican Republic	NaN	Central America & Caribbean	Santo Domingo	18.456792	-69.951164	Assassination	1.0	0.0	Julio Guzman	NaN	MANO-D	Private Citizens & Property

```
df.isnull().sum()
```

```
Year                     0
Month                    0
Day                      0
Country                  0
state                  421
Region                   0
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
dtype: int64
```

2. Data Analysis:

-Perform basic statistical analysis using Numpy to summarize the data

```
mean_val = np.mean(df.select_dtypes(include=[np.number]), axis=0)
median_val = np.median(df.select_dtypes(include=[np.number]), axis=0)
std_val = np.std(df.select_dtypes(include=[np.number]), axis=0)
```

```
frequent_val = df.select_dtypes(include=[object]).apply(lambda x: x.value_counts().idxmax())
```

-Use Pandas to:

```
attack_pyear = df.groupby('Year').size()
killed_pyear = df.groupby('Year')['Killed'].sum()
wounded_pyear = df.groupby('Year')['Wounded'].sum()
region_attack = df.groupby('Region').size()
count_attack = df.groupby('Country').size()
attacks = df['AttackType'].value_counts()
targets = df['Target_type'].value_counts()
killed_pregion = df.groupby('Region')['Killed'].sum()
wounded_pregion = df.groupby('Region')['Wounded'].sum()
```

```
df['Country'].value_counts().idxmax()
```

```
'Iraq'
```

```
df['Region'].value_counts().idxmax()
```

```
'Middle East & North Africa'
```

```
df['city'].value_counts().index[1]
```

```
'Baghdad'
```

```
df['Month'].value_counts().idxmax()
```

```
5
```

```
df['Year'].value_counts().idxmax()
```

```
2014
```

```
df['Group'].value_counts().index[1]
```

```
'Taliban'
```

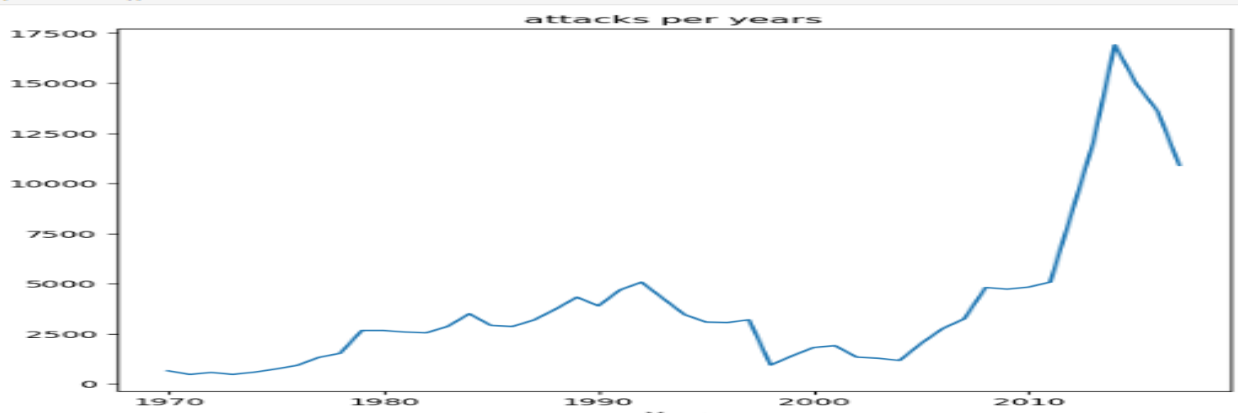
```
df['AttackType'].value_counts().idxmax()
```

```
'Bombing/Explosion'
```

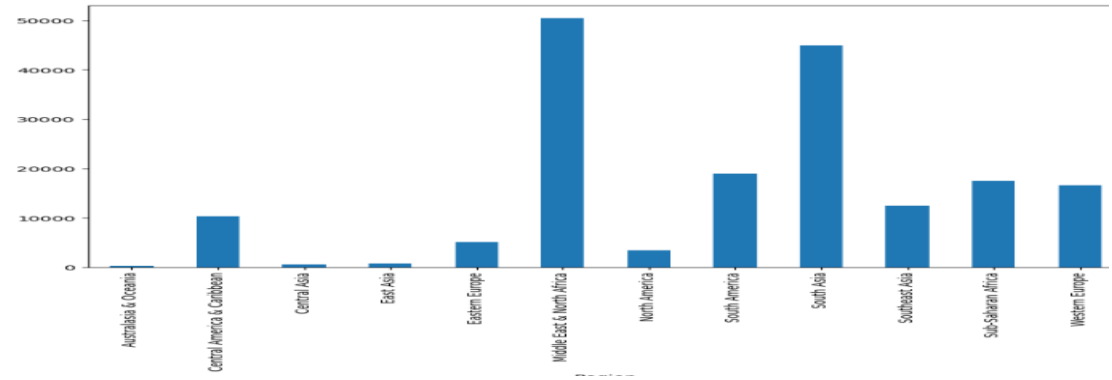
3. Data Visualization:

Line plot showing the trend of terrorist attacks over the years

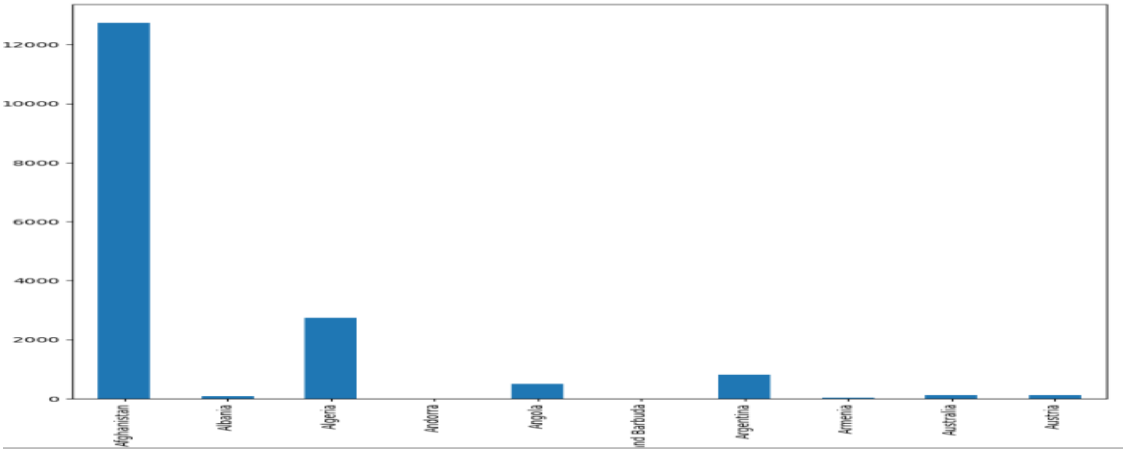
```
plt.figure(figsize=(7, 7))
sns.lineplot(data=attack_pyear)
plt.title(' attacks per years ')
plt.show()
```



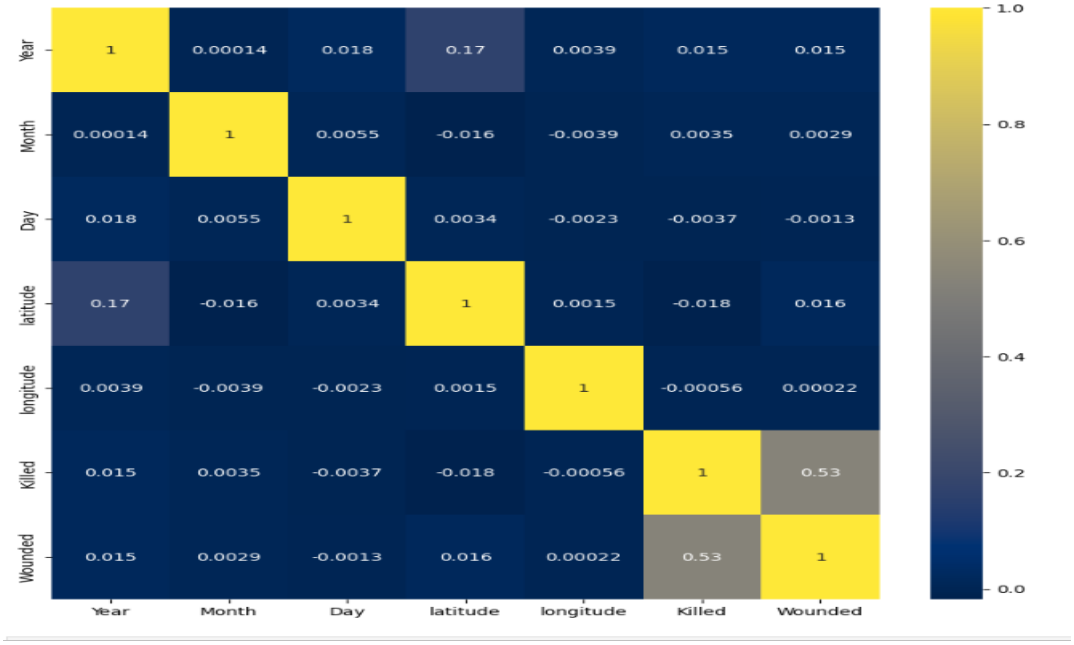
Bar plot of the number of attacks by country



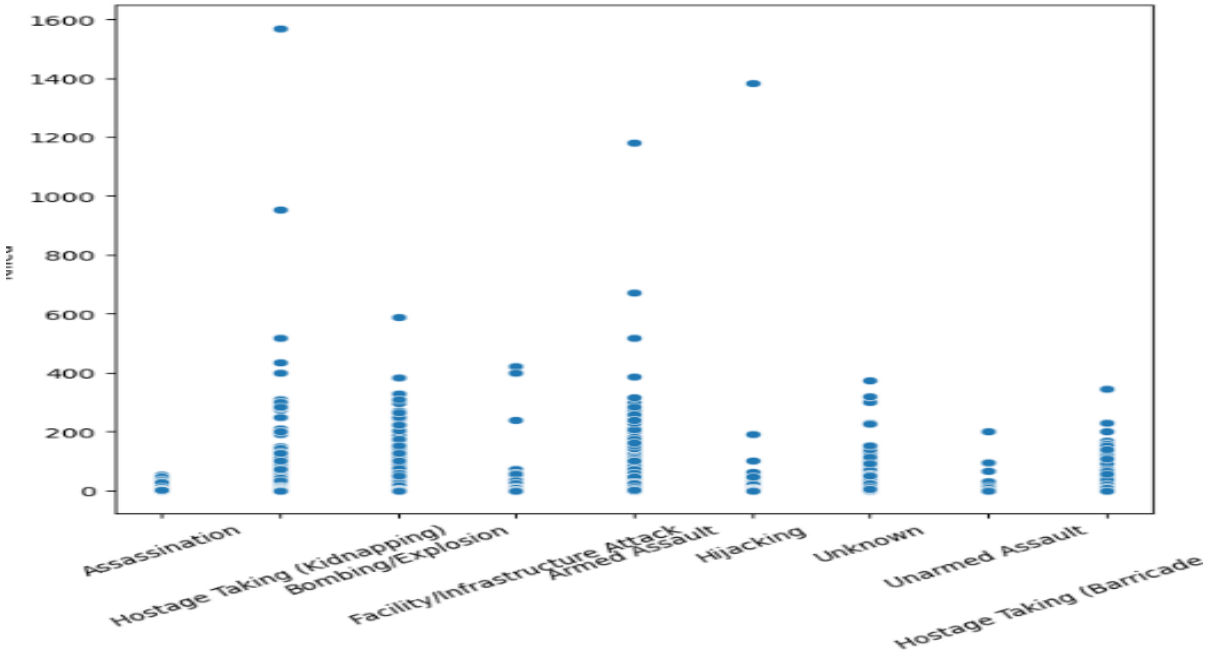
by region



Heatmap to visualize the correlation between deferent features



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



4. Performance Comparison with Dask:

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

```
# Pandas
start_time = time.time()
df = pd.read_csv('globalterrorismdb_0718dist.csv', encoding='ISO-8859-1')
load_time_pandas = time.time() - start_time
print(f"Pandas Load Time: {load_time_pandas:.2f} seconds")
```

```
Pandas Load Time: 3.22 seconds
```

```
# Dask
start_time = time.time()
ddf = dd.read_csv('globalterrorismdb_0718dist.csv' ,encoding='ISO-8859-1')
load_time_dask = time.time() - start_time

print(f"Dask Load Time: {load_time_dask:.2f} seconds")
```

```
Dask Load Time: 0.03 seconds
```