**Global Terrorism Analysis**

**Pramathesh Ray, Shila Rajpurohit,**

**Swapnil Anand**

**Data science trainees,**

**AlmaBetter**

**Abstract:**

**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 period 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.**

**The Project focuses on exploring and analyzing the data to discover key findings pertaining to terrorist activities.**

***Keywords: Terrorism, Exploratory Data Analysis, Insurgency***

**2. Introduction**

### Terrorism indicates a criminal and violent activity performed by an individual or group of individuals or an organization in order to strike terror among the general public and send messages to the public and governments, to fulfill a goal.

Terrorism is a major problem all over the world at the moment. Its direct effect is loss of human lives and injury. While the human cost is devastating, the economic impact may be larger than most realize. It affects a country’s economy directly by the destruction of economic properties. Terrorism indirectly affects the economy by creating market uncertainty, xenophobia, loss of tourism, and increased insurance claims.

In these scenarios, if one can predict the next time and place of attack, the world will benefit greatly. And to predict the future, one has to go through the past. So an analysis of past terrorist attacks holds significance in improving counter-terrorism measures and potentially preventing future attacks.

The Exploratory data analysis project involves analysis of the global terrorism data between 1970 to 2017, providing insights on the attacks conducted in different years and in different countries. We show the violent terrorist groups in the world, their attack types and their places of operations. We also analyze attacks in India over the years, in different states, the most active terrorist groups and their areas of operations.

## **3. Challenges Faced**

1. The dataset had too many variables (135 columns). It was difficult to view all data at once for preliminary data exploration. Also analysis on all columns was difficult.
2. Columns were not properly named e.g. *‘gname’, ‘propextent\_txt’*. To know the contents of a column was challenging and it was possible after a thorough exploration of the contents of a column.
3. The data had too many ‘Nan’ and ‘Unknown’ values. This lack of information forced us to not consider for analysis. Even though numerical data were filled, textual data were not possible to be filled and so the rows had to be dropped.
4. Inconsistent data in ‘Day’ and ‘Month’ columns made us to drop rows and in binary variables were converted to False.

## **3. Data Summary**

The dataset has one csv file named “Global Terrorism Data.csv”. The csv file has 1,81,691 rows and 135 columns. The rows represent a particular incident of a terrorist attack identified by the 1st column “eventid”. The project does not involve an analysis on all of the 135 column or variables. The following are the column names or variables which are used in the project along with its description.

'eventid' : Serial number of a particular incident, useful to keep count of incidents

'iyear' : year of the incident

'imonth' : month of the incident

'iday' : day of the incident

'country\_txt' : country in which the incident occurred

'region\_txt' : region in which the incident occurred

'provstate' : state or province of the country in which the incident occurred

'summary' : summary of the incident, when available

'alternative\_txt' : type of attack if it was not terrorist for certain

'attacktype1\_txt' : general method of attack used (i.e. assassination, hijacking, bombing/explosion, etc.)

'targtype1\_txt' : general type of target/victim (i.e. business, government, police, military, etc.)

'gname' : terrorist group responsible for the attack

'target1' : specific person, building, installation, etc. that was targeted

'nkill' : number of dead

'nwound' : number of wounded

'property' : was property damaged during the attack?

'propextent\_txt' : type of property damage (major, minor or catastrophic in terms of monetary value)

'latitude' : latitude of the incident

'longitude' : longitude of the incident

Some of the column names or variables not used in the project along with its description are:

'suicide' : did the incident involve a suicide attack?

'crit1' : was the attack aimed at attaining a political, economic, religious, or social goal?

'crit2' : was there intent to coerce or intimidate a larger audience than the victims?

'crit3' : was the incident outside legitimate warfare activities (i.e. target non-combattants)?

'doubtter' : was there doubt as to whether or not the incident is a terrorist attack

'extended' : has the incident lasted for more than 24 hours?

'multiple' : is the incident connected to other attacks?

'success' : did the terrorist attack achieve its goal (i.e. assassination, etc.)?

'guncertain1' : was the terrorist group confirmed?

'claimed' : was the incident claimed by a particular group?

'ishostkid' : were victims taken hostages or kidnapped?

'natlty1\_txt' : nationality of the target/victim

'weaptype1\_txt' : general type of weapon used in the incident (i.e. biological, chemical, firearms, etc.)

Basically, the variables can be classified into 4 categories:

1. **Spatio-temporal variables:** Values which contains information about space and time. E.g. Month, Year, latitude and longitude
2. **Categorical variables:** variable that can take on fixed number of possible values. E.g. Country, Region, States, Terrorist group
3. **Binary variables:** the one that takes only the value 0 or 1 to indicate the absence or presence. E.g. Is property damaged?
4. **Continuous variables:** they are typically obtained by measuring or counting numerically. E.g.: No. of Kills, No. of wounded.

## **4. Data Cleaning**

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. It is necessary to clean the data before performing analysis to have accurate outputs.

1. **Checking duplicate rows**: duplicate rows are checked and if found are removed from the dataset. We use duplicated() function on the dataframe for this. However, no duplicate rows were found so we proceed.
2. **Cleaning Spatio-temporal variables**: Some rows had month and day column values set to zero. These rows were dropped using the drop() function. After this step, no. of rows was reduced to 1,80,800.
3. **Cleaning continuous variables:** Number of deaths and number of wounded columns had many null values. We replaced those missing values with the median values of the column using fillna() method on the dataframe with NumPy median given as a parameter.
4. **Cleaning Binary variables:** Property damaged column should have values 0 and 1 representing False and True only but some rows have -9. So we converted them to 0 meaning False using replace() method on the column.
5. **Cleaning Categorical variables:** Indian state data has 2 values "Orissa" and "Odisha" which correspond to the same state. Also, it has "Andhra Pradesh" and "Andhra pradesh" as two separate values. So we bring these rows under one name. We create a function to replace the above names and using the apply() method, the function is called on all values of ‘states’

# **5. Tools used in EDA**

1. **Matplotlib Library**

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram, etc.

1. **Line Plot**

Line Plots display numerical values on one axis and categorical values on the other. They can typically be used in much the same way Bar Plots can be used, though, they're more commonly used to keep track of changes over time.

To plot a line plot in Matplotlib, we use the generic plot() function from the PyPlot instance of the Matplotlib module.

1. **Bar Plot**

A bar plot or bar chart is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent.

A bar chart describes the comparisons between the discrete categories. One of the axis of the plot represents the specific categories being compared, while the other axis represents the measured values corresponding to those categories.

To plot a bar plot in Matplotlib, we use the generic bar() function from the PyPlot instance of Matplotlib module.

1. **Pie chart**

A Pie Chart can only display one series of data. Pie charts show the size of items (called wedges) in one data series, proportional to the sum of the items. The data points in a pie chart are shown as a percentage of the whole pie. To plot a pie plot in Matplotlib, we use the generic pie() function from the PyPlot instance of Matplotlib module

1. **Pandas Library**

Pandas is an open-source library that is made mainly for working with relational or labeled data. DataFrame is the most important and widely used data structure and is a standard way to store data.

DataFrame has data aligned in rows and columns like a spreadsheet database. We import the CSV file “Global Terrorism Data.csv” into a dataframe using pandas method read\_csv() to begin the analysis.

* **Inspecting data in DataFrame**

Running the DataFrame using its name displays the entire table. We also use df.head(n)to get the first n rows or df.tail(n)to print the last n rows.

df.columns prints the columns of the DataFrame.

We can get a statistical summary (count, mean, standard deviation, min, max etc.) of the data using df.describe() function.

df.unique() returns unique values of column.

* **Groupby function**

This project extensively uses the pandas groupby method. Pandas groupby is used for grouping the data according to the categories and apply a function to the categories. It also helps to aggregate data efficiently. Pandas dataframe.groupby() function is used to split the data into groups based on some criteria.

The column name given as a parameter is used to determine the groups for the groupby. The function returns a groupby object that contains information about the groups.

An aggregated function returns a single aggregated value for each group. Once the groupby object is created, several aggregation operations can be performed on the grouped data to compute a summary statistic for each group.

The most basic aggregation method is counting. count() function counts the number of values in each group. Other aggregate functions include sum(), min(), max(), mean(), etc.

At times when we need to perform groupby on multiple columns of DataFrame, we do this by passing a list of column labels that we want to perform group by on. Then we use the agg() function which allows us to specify multiple aggregation functions at once like .agg(['max', 'min', 'count', 'median', 'mean']) on different groups.

* **Other Pandas methods**

Dataframe.sort\_values() is used to sort the dataframe according to a column which is given as a value to parameter ‘by’. Sorting in descending order is done by giving ‘False’ value to parameter ‘ascending’.

Pandas DataFrame apply() function is used to apply a function along an axis of the DataFrame.

It takes another function as parameter value.

* **Pandas plot**

Pandas also provides several different options for visualizing data with method plot() which is based on matplotlib’s pyplot. When we call .plot() on a DataFrame object, Matplotlib creates the plot under the hood

The ‘kind’ parameter accepts different string values and determines which kind of plot to create like "bar" for vertical bar charts, "line (default)" for line graphs , "pie" for pie charts, etc.

Other optional parameters for plot() can be added like

“color” for color of plot

“figsize” for size of figure

“fontsize” for size of font

“marker” for marker type

“ms” for marker size

We can format the plot by calling methods on the Pyplot object like

“xlabel”: to set labels on x-axis

“ylabel”: to set labels on x-axis

“legend”: to set legend

“grid”: to show gridlines on the plot

“axvline”: to show a vertical line at given value

“xticks”: to set xtick labels and their orientation

1. **Geopandas Library**

GeoPandas module makes working with geospatial data in python easier. GeoPandas is based on the pandas. It extends pandas data types to include geometry columns and perform spatial operations. It combines the capabilities of pandas and shapely, providing geospatial operations in pandas and a high-level interface to multiple geometries to shapely.

Geospatial data describe objects, events, or other features with respect to a location (coordinates) of the earth.

The main data structure in GeoPandas is the GeoDataFrame that extends the pandas DataFrame. So all the base DataFrame operations can be performed on the GeoDataFrame.

“naturalearth\_lowres” is a base map provided with geopandas which can be loaded.

df\_world = geopandas.read\_file(geopandas.datasets.get\_path( 'naturalearth\_lowres'))

df\_world is of type GeoDataFrame with continent, (country) name, and geometry (of country area) columns. The boundary.plot() function on df\_world plots the world map with boundary only as follows:



# **5. EDA - Exploratory Data Analysis**

## We have divided the EDA into 2 parts.

## 1. Analysis on World data

## 2. Analysis on India data

## **World data**

We begin with the analysis on whole dataset i.e. we analyze terrorist attacks over the whole world.

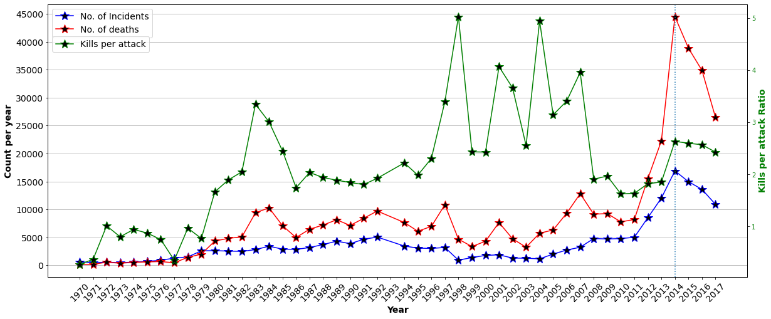
* **Timeline of terrorist attacks**

The dataset contains data from 1970 to 2017. In this section we see the number of attacks conducted worldwide every year along with the death count and kills per attack ratio.

Using Python groupby()method we grouped data according to year of incident, then we applied the aggregate function using agg() of count on *‘eventid’* column and sum on *‘nkill’* column to find number of attacks and number of deaths.

To plot these, we use line-plot as we are showing changes over a time. We also calculate another metric “Kills per attack ratio”. Since the range of this metric is different than other metrics, to show this we create a secondary y-axis on the same line-plot using twinx().

In the Axes Module of Matplotlib, there is a function named Axes.twinx() function which is used to create a twin Axes that are sharing the same x-axis. Other formatting functions of matplotlib are used to modify the plot.

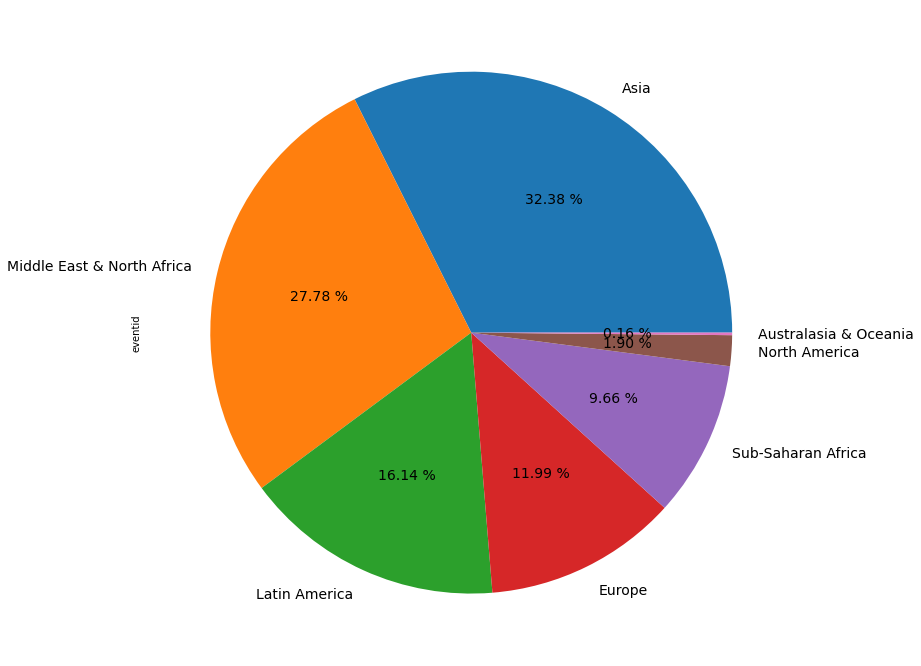


* **Countries and Regions with most attacks**

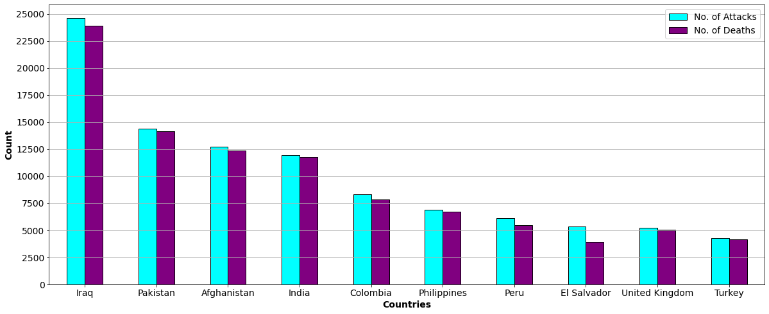
We would present an analysis on continent wise terrorism data. The dataset however contains continents divided like 'Western Europe' and 'Eastern Europe' and similarly with Asia. So we combine the regions to make our analysis simple.

We combined South America and Central America and the Caribbean into Latin America, East Asia, South Asia, Central Asia & Southeast Asia into Asia and Western Europe and Eastern Europe into Europe.

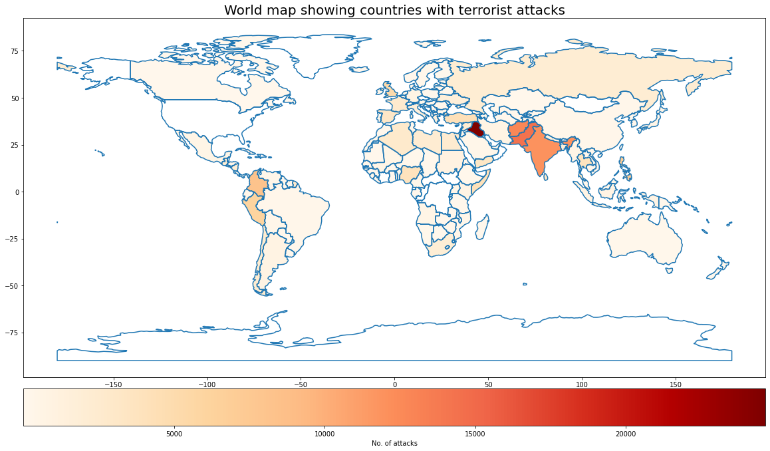
Then we find terrorist attacks based on regions. We do that by grouping the data according to regions using groupby(). We then apply the count() function on *‘eventid’* column to find the number of attacks and creating a pie chart to show the divisions and percentage of attacks done on a region.



Then we do country wise distribution of terrorist attacks using same approach of grouping by countries. Since we have too many countries, we plot the top 10 countries with most terrorist attacks on a bar plot instead of a pie chart.



We also use the geopandas module of python to create a world map showing countries affected by terrorism. Choropleth maps colour the regions/polygons in relation to a data variable. Using the country columns of the data set we merge the data with geodataframe to shade the countries with terrorist attacks on the world map.

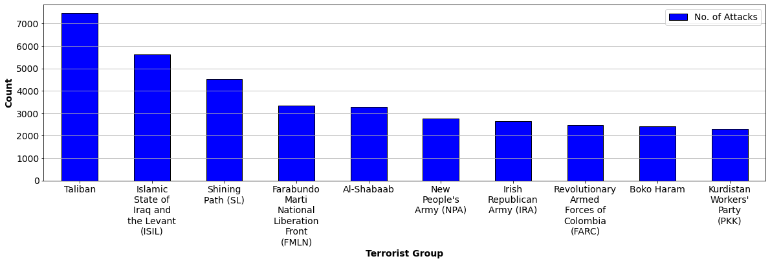


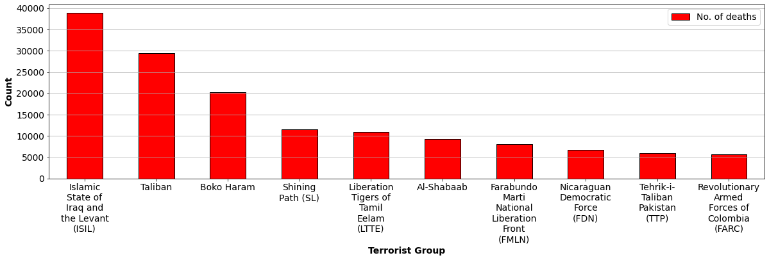
* **Terrorist organizations with most number of attacks**

In this section, we present the list of the terrorist organizations who have performed most no. of attacks and have inflicted maximum casualties.

We group the data set according to *‘gname’* column which stands for terrorist groups name. We then aggregate based on number of attacks and number of kills.

We create 2 different bar plots to show the top 10 terrorist organizations in terms of no. of attacks and death count. (A single bar plot can rank based on only one factor, but we want to see the rankings in terms of both attack count and death count)

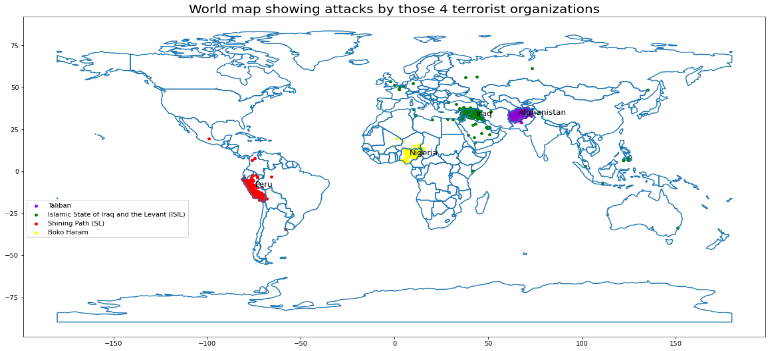




We also add another metric that is ‘kills per attack ratio’ which is obtained by dividing number of kills by number of attacks.

We obtain top 4 violent terrorist groups. We want to show in which time and country they were active. To obtain the timeline of the terrorist groups we perform group by operation on ‘Year’ and ‘Terrorist groups name’ columns. Then we plot the number of attacks per year for each terrorist group on line-plot.

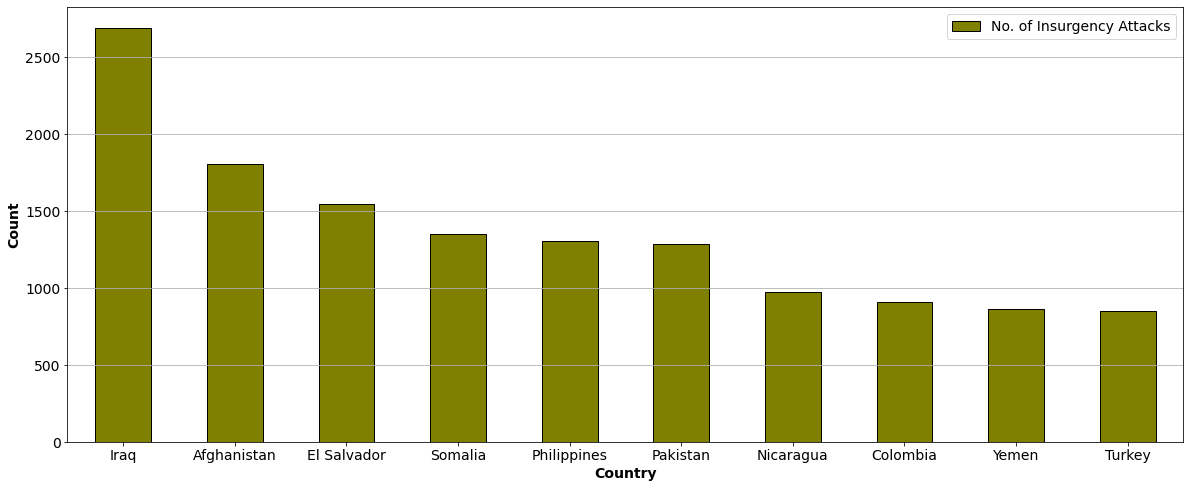
Using the geopandas module, we create world map and then we plot the areas of attacks by these terrorist groups with separate colours and use annotate() function to show the countries where they have majorly operated on.



* **Insurgencies**

In this section, we present the data on insurgencies. ‘*alternative\_txt’* column contains information on whether an attack is an insurgency. So we create a filter to create a separate dataframe for insurgencies.

Then we group the insurgency data according to countries and sort it to find the countries that have faced the most number of insurgencies and use bar plot to show it graphically.



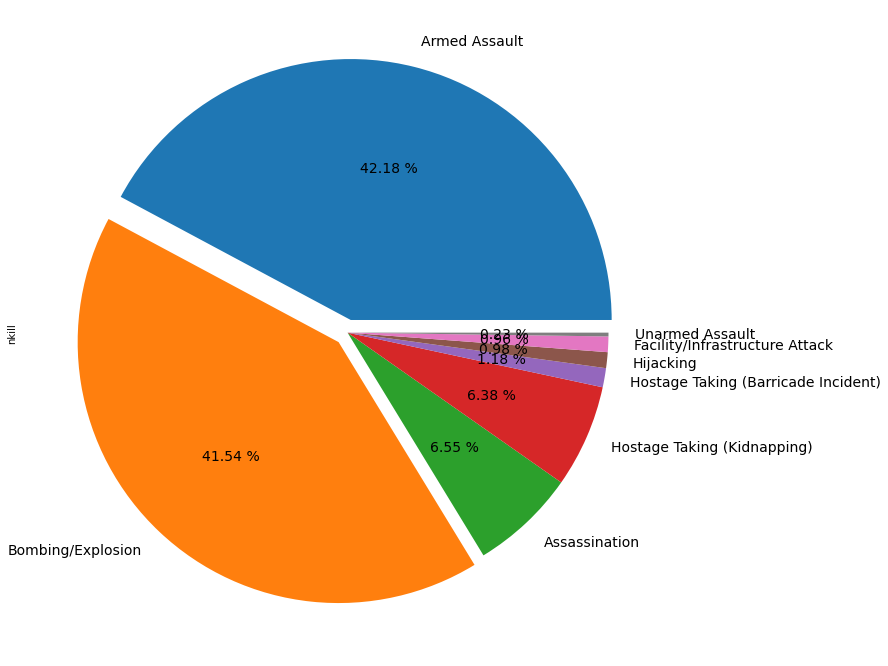
* **Most lethal attacks**

We print a list of top 3 attacks which has claimed most number of lives in a single attack. For this we sort the dataframe based on *‘nkill’* column and use head(3) method.

* **Deadliest type of attack**

In this section, we present the information on type of attack used by terrorists. *'attacktype1\_txt'* column has information on the type of attacks used by terrorist for a particular event like assassination, hijacking, bombing/explosion, etc.

We group data by *'attacktype1\_txt'* and use pie chart to show which attack has taken how much lives in terms of percentage of total deaths.



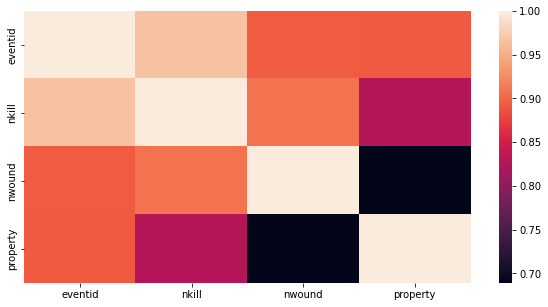
* **Correlation**

Our general conscience says more the number of attacks, more the number of deaths and wounded. So we check that mathematically. Here, we try to find correlation between "No. of Terrorist attacks", "No. of Deaths per attack", "No. of wounded per attack" and also "Count of Property damaged".

First we make a dataframe of only these columns which are grouped by each year.

Then we use corr() function which computes pairwise correlation of columns and returns the correlation matrix. By default it uses ‘Pearson’ correlation coefficient.

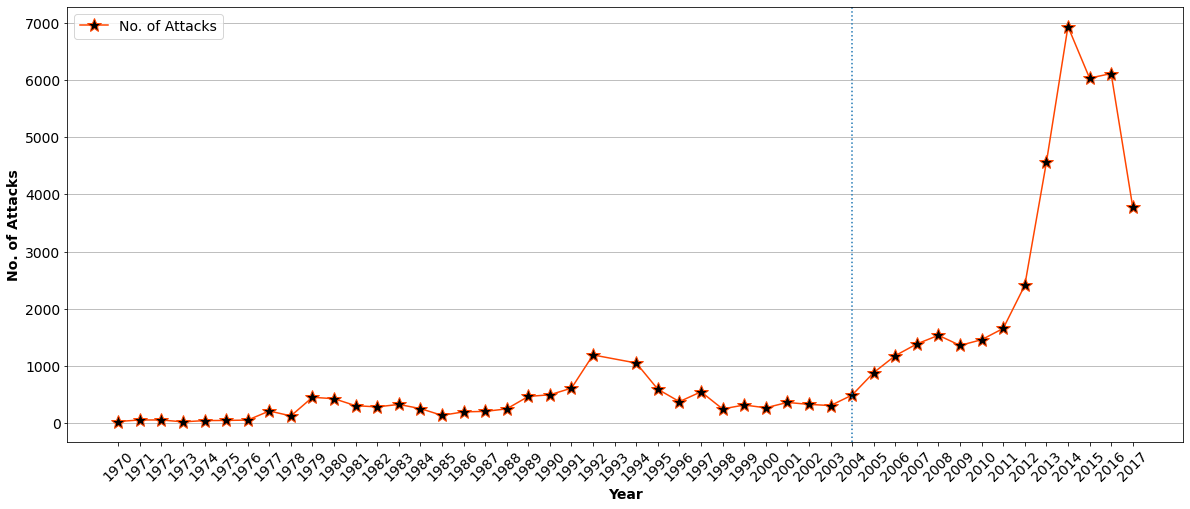
Also we plot the heatmap for the correlation matrix. Heatmap is defined as a graphical representation of data using colors to visualize the value of the matrix. For this, we use the Seaborn module. Heatmaps in Seaborn can be plotted by using the seaborn.heatmap() function.

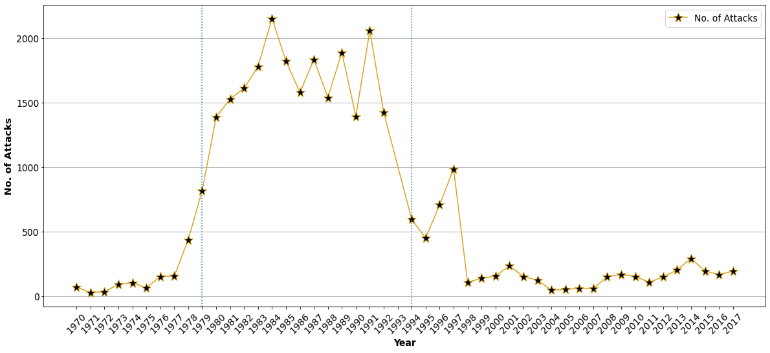


* **Comparing Terrorism in Middle East and Latin America**

We compare here the trend in terrorism over the years for the regions of Middle East and Latin America.

We proceed by creating a separate dataframe for the regions. For Latin America, we combine the regions of Central America & Caribbean with South America and country of Mexico. Then we group both data according to years and count the attacks to plot a line-plot for both the regions.

****

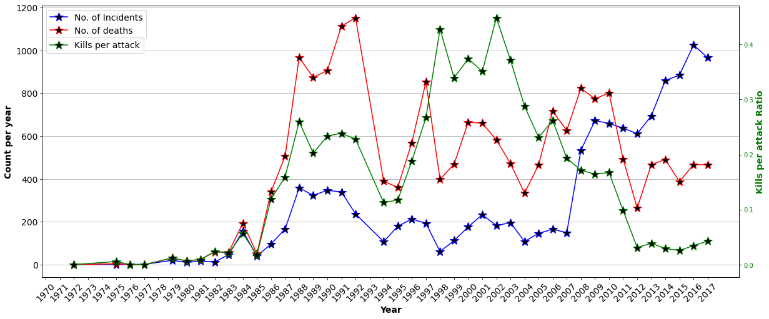
****

## **India data**

We then analyze the terrorism data for our country India. So we make a separate dataframe variable containing data for country "India" only..

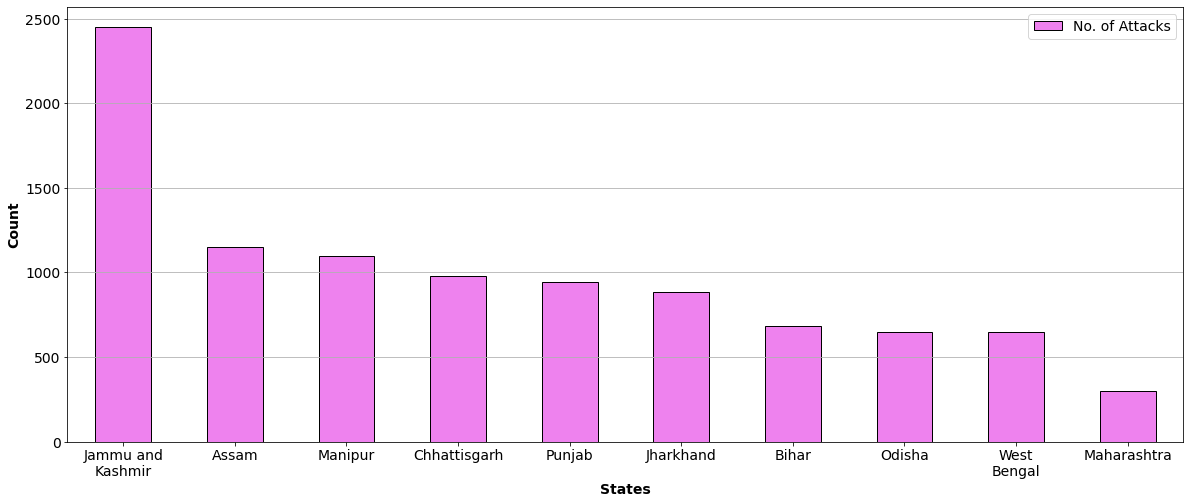
* **Timeline of Indian attacks**

To present terrorism timeline in India, we groupby the data according to years. Then using agg() we find No. of attacks and No. of Deaths per year and plot those on line-plot. We also find ‘Kills per attack ratio’ on plot it on same plot using twinx() function.

****

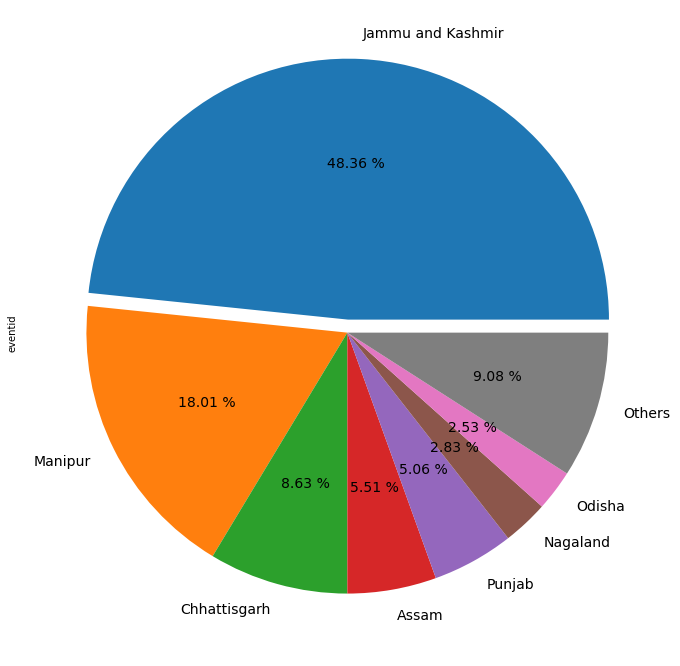
* **States with most number of attacks**

To find which state faced most terrorist attacks, we use groupby() function on *‘provstate’* column to categorize according to the states. Then using count() we count the number of attacks on each state and plot the top 10 states with most attacks on bar plot.

****

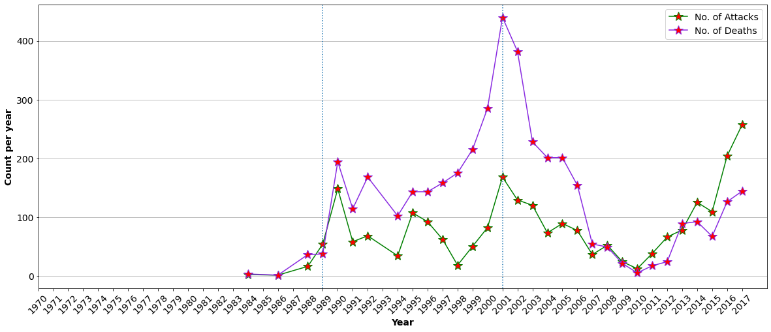
* **States with most insurgencies**

We create insurgency data by applying filter on .‘*alternative\_txt’* column. Then we group according to states, count no. of attacks and sort it in descending order to find the states with most insurgencies and plot it on pie plot.



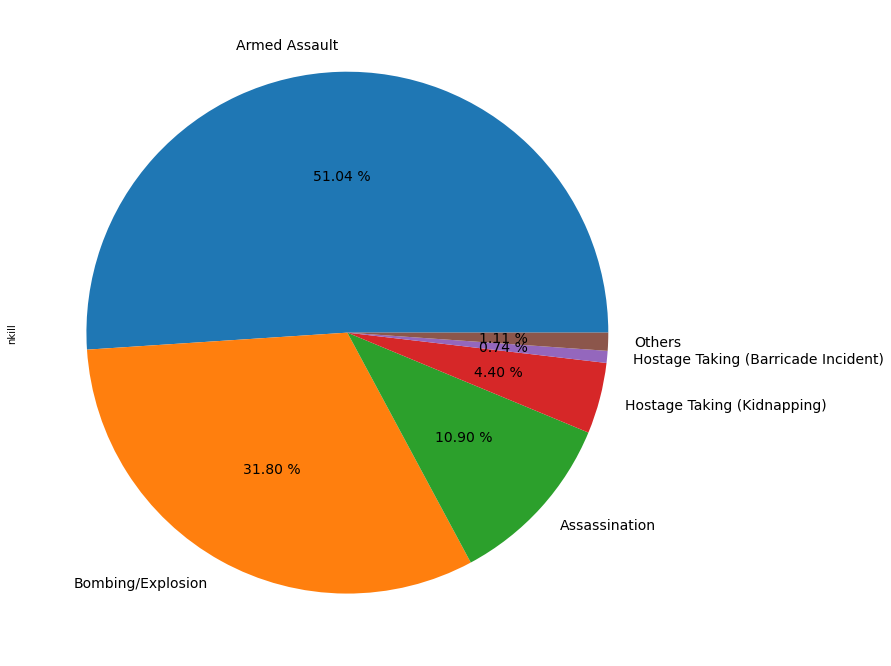
* **Jammu and Kashmir terrorism timeline**

We create dataframe by applying filter on *‘provstate’* and selecting ‘Jammu & Kashmir’. We then group according to year and count the no. of attacks to plot it on line-plot to show the terrorism in Jammu & Kashmir over the years.



* **Deadliest type of attacks in India**

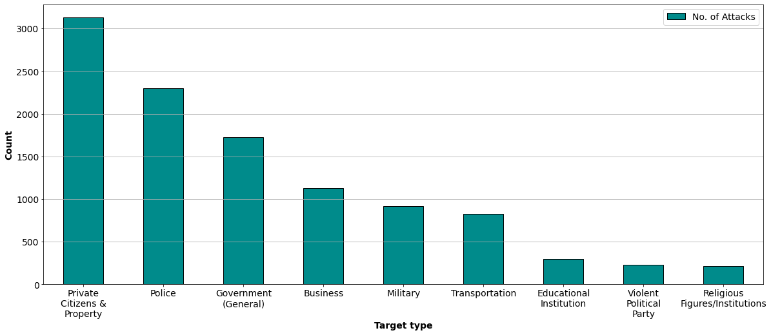
We group data by *'attacktype1\_txt'* and use count() on *‘nkill’* to find which attack has taken how much lives and use pie chart to show it in terms of percentage of total deaths..

****

* **Terrorists' Targets in India**

Here, we find out what were the top targets for the terrorists. *‘targtype1\_txt’* column has information on the type of targets that were attacked by the terrorist for a particular event like Private Citizens & Property, Educational Institution, Business, etc.

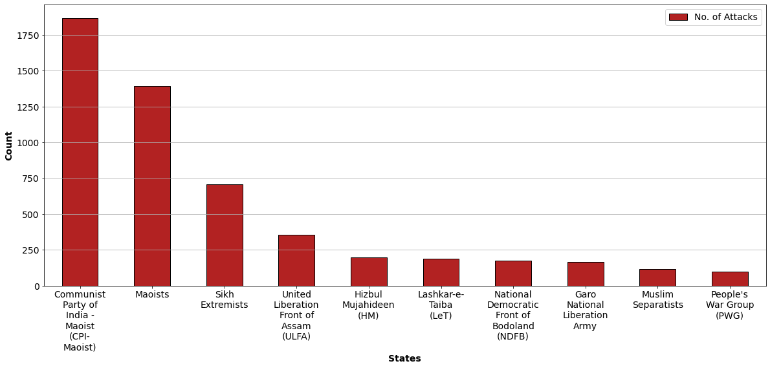
We group data by *'targettype1\_txt'* and count no. of attacks and use bar plot to show which was targeted most times.



* **Terrorist Organizations with most number of attacks**

Here, we present the list of the terrorist organizations who have performed most no. of attacks in India.

We group the data set according to *‘gname’* column which stands for terrorist groups name. We then aggregate based on number of attacks and plot it on bar plot.

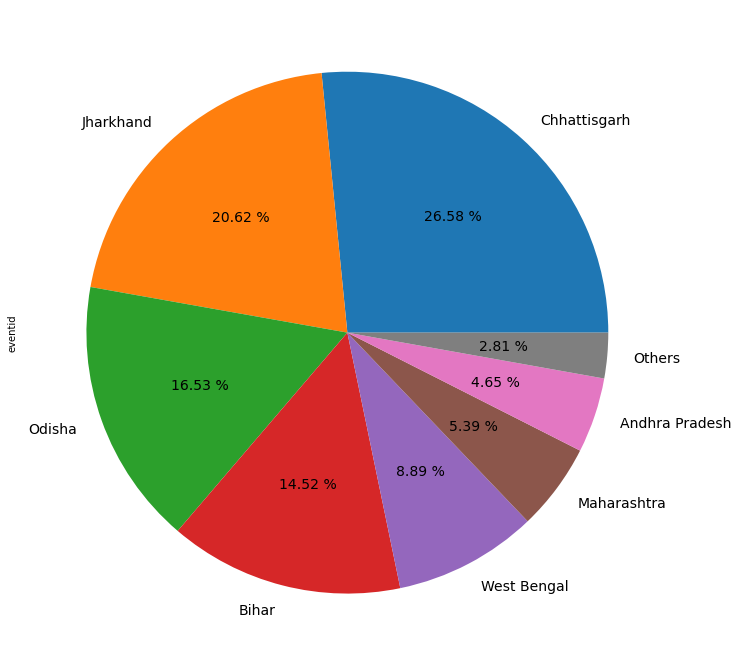
****

* **Indian states affected by Maoism**

Previous section showed that Maoist groups have conducted most no. of attacks in India. So here we find in which states the Maoists groups have operated mostly.

We filter the data based on *‘gname’,* those attacks which were conducted by groups from any of the following groups: *'Naxalites', 'Communist Party of India - Maoist (CPI-Maoist)', 'Maoists','Maoist Communist Center (MCC)', 'Communist Party of India', 'Communist Party of India-Marxist', 'Communist Trade Union Workers', 'Maoist Farm Laborers Struggle Committee (MXSS)', 'Communist Party of India- Marxist-Leninist', 'Maoist Communist Party of Manipur'*

After grouping the filtered data according to states and counting the no. of attacks, we plot the states with most attacks by Maoist groups on Pie chart.

****

**CONCLUSIONS**

1. Before 2000s, global terrorist activities remained comparatively lower than period after 2000s which reached peak in 2014. Although since then, drop in terrorist activities is seen but still it is higher than any other time period. Kills per attacks ratio which were at high during late 90s and early 2000s has reduced after 2007. 1970s were the most peaceful period.
2. Regions of Asia and Middle East & North Africa saw most no. of terrorist activities. Australasia & Oceania is the most peaceful region. Iraq saw the most attacks as a country.
3. Terrorist organizations of Islamic State of Iraq and the Levant (ISIL) and Taliban have conducted most no. of terrorist activities. Boko Haram has the highest Kills per attack ratio.
4. Iraq tops the list of countries with most insurgency attacks.
5. Regions of Middle East & North Africa and Latin America had a contrasting trend in terrorism where Latin America had most of the terrorism in 1980s and early 1990s which subdued later, Middle East & North Africa saw the rise in terrorism mainly after 2000.
6. Terrorism in India increased after mid 1980s. Since 2007, attacks have increased drastically. However, India has managed to keep the death count low in recent times compared to period before 2007.
7. Jammu & Kashmir has seen the most no. of terrorist attacks and most no. of insurgency attacks among the Indian states, particularly after 1989.
8. Armed Assault and Bombing/Explosion are the types of attacks that have taken most lives globally and also in India.
9. Maoist groups have had the most no. of terrorist attacks in India. They have operated mostly in Eastern states of Chhattisgarh, Jharkhand, Odisha and Bihar.