Data Science And Bussiness Analytics Intern At TheSparksFoundation

GRIPJAN21

Author:- Rajat Kumar

Task 4:-Exploratory Data Analysis - Terrorism

Problem Statement:- Perform ‘Exploratory Data Analysis’ on dataset Global Terrorism.

My work:-As a security/defense analyst, try to find out the hot zone of terrorism.

Step1: Importing some libaries.

In [4]:

```
import numpy as np
import pandas as pd
from pandas import ExcelWriter
from pandas import ExcelFile
import seaborn as sns
import matplotlib.pyplot as plt
```

Step 2: Reading my dataset.

In [5]:

```
## we need to encode this dataset in to ISO
df= pd.read_csv('terrorism.csv', sep=',', encoding='ISO-8859-1')
```

```
C:\Users\Rajat Kumar\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3146: DtypeWarning: Columns (4,6,3
1,33,61,62,63,76,79,90,92,94,96,114,115,121) have mixed types.Specify dtype option on import or set low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

In [6]:

```
df.head()
```

Out[6]:

	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	country_txt	region	...	addnotes	scite1	scite2	scite3	dbsourc
0	197000000001	1970	7	2	NaN	0	NaN	58	Dominican Republic	2	...	NaN	NaN	NaN	NaN	PGI
1	197000000002	1970	0	0	NaN	0	NaN	130	Mexico	1	...	NaN	NaN	NaN	NaN	PGI
2	197001000001	1970	1	0	NaN	0	NaN	160	Philippines	5	...	NaN	NaN	NaN	NaN	PGI
3	197001000002	1970	1	0	NaN	0	NaN	78	Greece	8	...	NaN	NaN	NaN	NaN	PGI
4	197001000003	1970	1	0	NaN	0	NaN	101	Japan	4	...	NaN	NaN	NaN	NaN	PGI

5 rows × 135 columns

Step 3: Renameing some columns of my dataset.

In [7]:

```
df.rename(columns={'iyear':'Year','imonth':'Month','extended':'Extended','iday' :'Day','country_txt' :'Country','regi
                   'provstate': 'State','nwound':'Wounded','nkill':'Killed','attacktype1_txt':'AttackType','targett
```

In [8]:

```
df.head()
```

Out[8]:

	eventid	Year	Month	Day	approxdate	Extended	resolution	country	Country	region	...	addnotes	scite1	scite2	scite3	dbsource
0	197000000001	1970	7	2	NaN	0	NaN	58	Dominican Republic	2	...	NaN	NaN	NaN	NaN	PGIS
1	197000000002	1970	0	0	NaN	0	NaN	130	Mexico	1	...	NaN	NaN	NaN	NaN	PGIS
2	197001000001	1970	1	0	NaN	0	NaN	160	Philippines	5	...	NaN	NaN	NaN	NaN	PGIS
3	197001000002	1970	1	0	NaN	0	NaN	78	Greece	8	...	NaN	NaN	NaN	NaN	PGIS
4	197001000003	1970	1	0	NaN	0	NaN	101	Japan	4	...	NaN	NaN	NaN	NaN	PGIS

5 rows × 135 columns

In [9]:

```
df.shape ## This command will give total number of row and columns.
```

Out[9]:

```
(181691, 135)
```

Step 4: Deleting some columns.

In [10]:

```
df=df[['Year','Month','Extended','Day','Country','State','Region','City','Latitude','Longitude','AttackType','Killed
```

In [11]:

```
df.shape  ## this command will give total number of columns in our dataset.
```

Out[11]:

```
(181691, 16)
```

Step 5: Checking for null value.

In [12]:

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

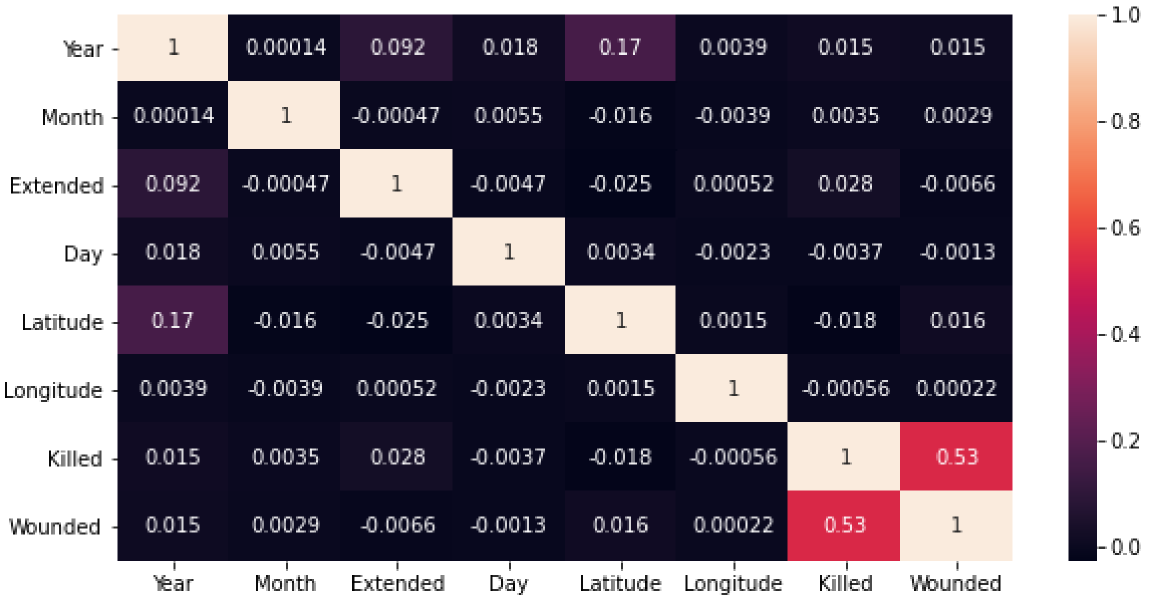
Out[12]:

```
Year                 0
Month                0
Extended             0
Day                  0
Country              0
State              421
Region               0
City               434
Latitude          4556
Longitude         4557
AttackType           0
Killed           10313
Wounded          16311
Group                0
Weapon_Type          0
Motive          131130
dtype: int64
```

Step 6: Finding correlation and visualizing it using heat map.

In [13]:

```
plt.figure(figsize=(10,5))
sns.heatmap(df.corr(),annot=True)
plt.show()
```



Step 7: Finding top 20 countries with most Terrorist Attacks.

In [14]:

```
print('Country with the most attacks:\n',df['Country'].value_counts().head(20))
```

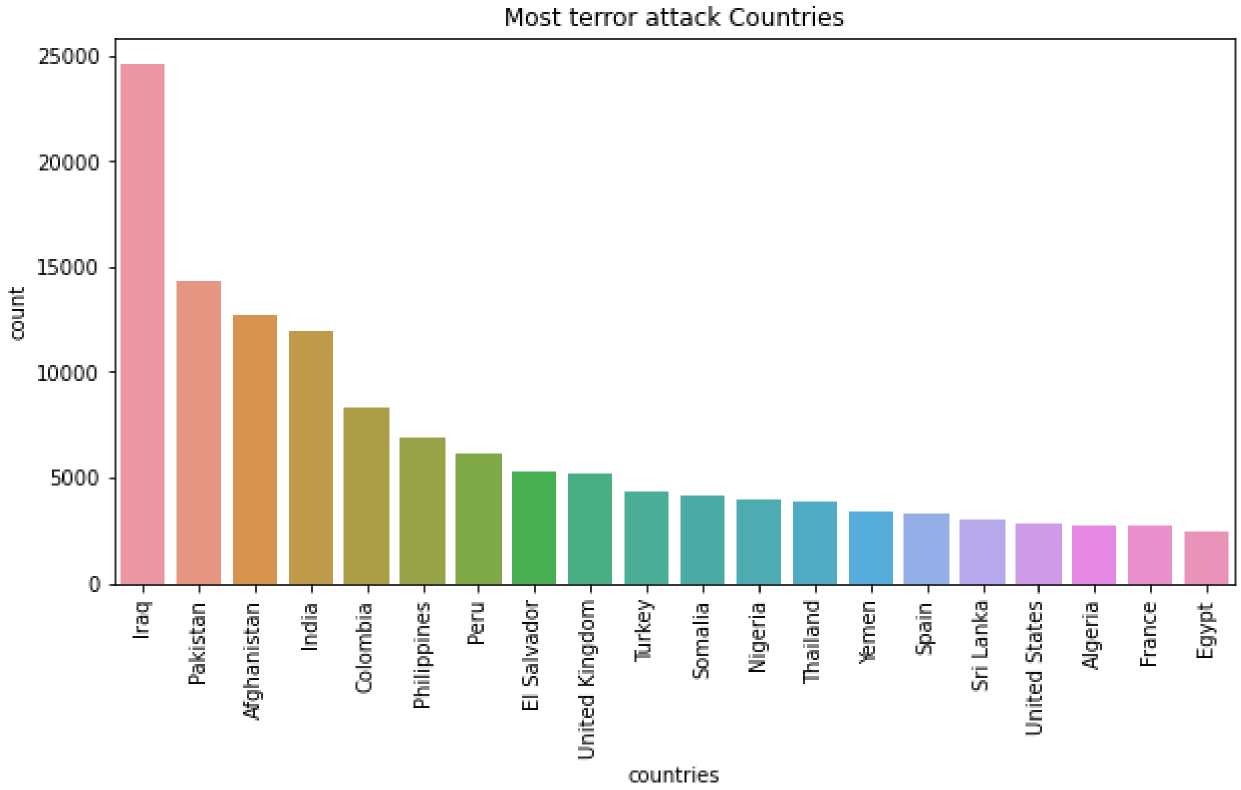
```
Country with the most attacks:
 Iraq              24636
Pakistan          14368
Afghanistan       12731
India             11960
Colombia           8306
Philippines        6908
Peru               6096
El Salvador        5320
United Kingdom     5235
Turkey             4292
Somalia            4142
Nigeria            3907
Thailand           3849
```

```
Yemen                3347
Spain                3249
Sri Lanka            3022
United States        2836
Algeria              2743
France               2693
Egypt                2479
Name: Country, dtype: int64
```

In [15]:

```
plt.figure(figsize=(10,5))
sns.barplot(df['Country'].value_counts()[:20].index,df['Country'].value_counts()[:20].values)
plt.title("Most terror attack Countries")
plt.xlabel("countries")
plt.ylabel('count')
plt.xticks(rotation=90)
plt.show()
```

```
C:\Users\Rajat Kumar\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variabl
es as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



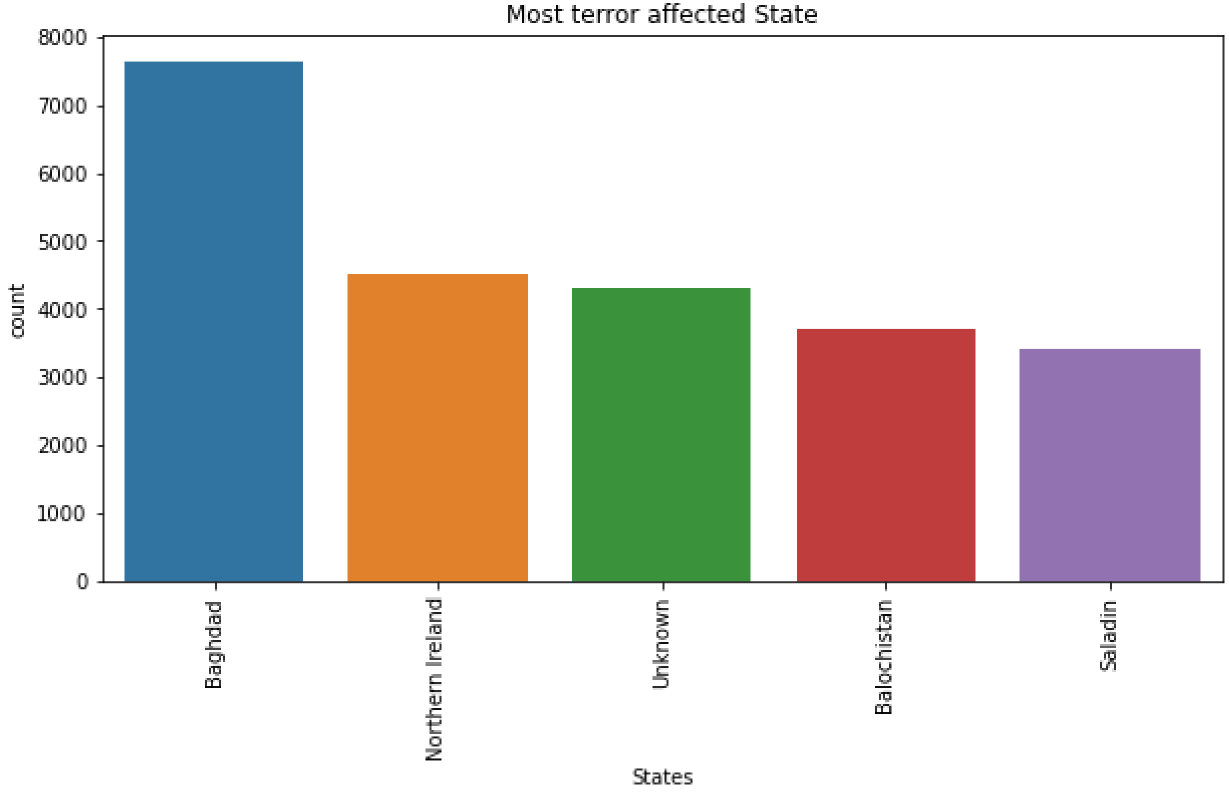
Step 8: Top five Staes with most terrorist Attacks

```
In [32]: print(' Top 5 State with most terror attacks are:\n',df['State'].value_counts().head(5))
```

```
 Top 5 State with most terror attacks are:
 Baghdad              7645
Northern Ireland    4498
Unknown             4290
Balochistan         3710
Saladin             3411
Name: State, dtype: int64
```

```
In [17]: plt.figure(figsize=(10,5))
         sns.barplot(df['State'].value_counts()[:5].index,df['State'].value_counts()[:5].values)
         plt.title('Most terror affected State')
         plt.xlabel("States")
         plt.ylabel('count')
         plt.xticks(rotation=90)
         plt.show()
```

```
C:\Users\Rajat Kumar\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variabl
es as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



Step 9:Top 10 region with most terrorist Attacks

```
In [18]: print('Region with most terror attacks:\n',df['Region'].value_counts().head(10))
```

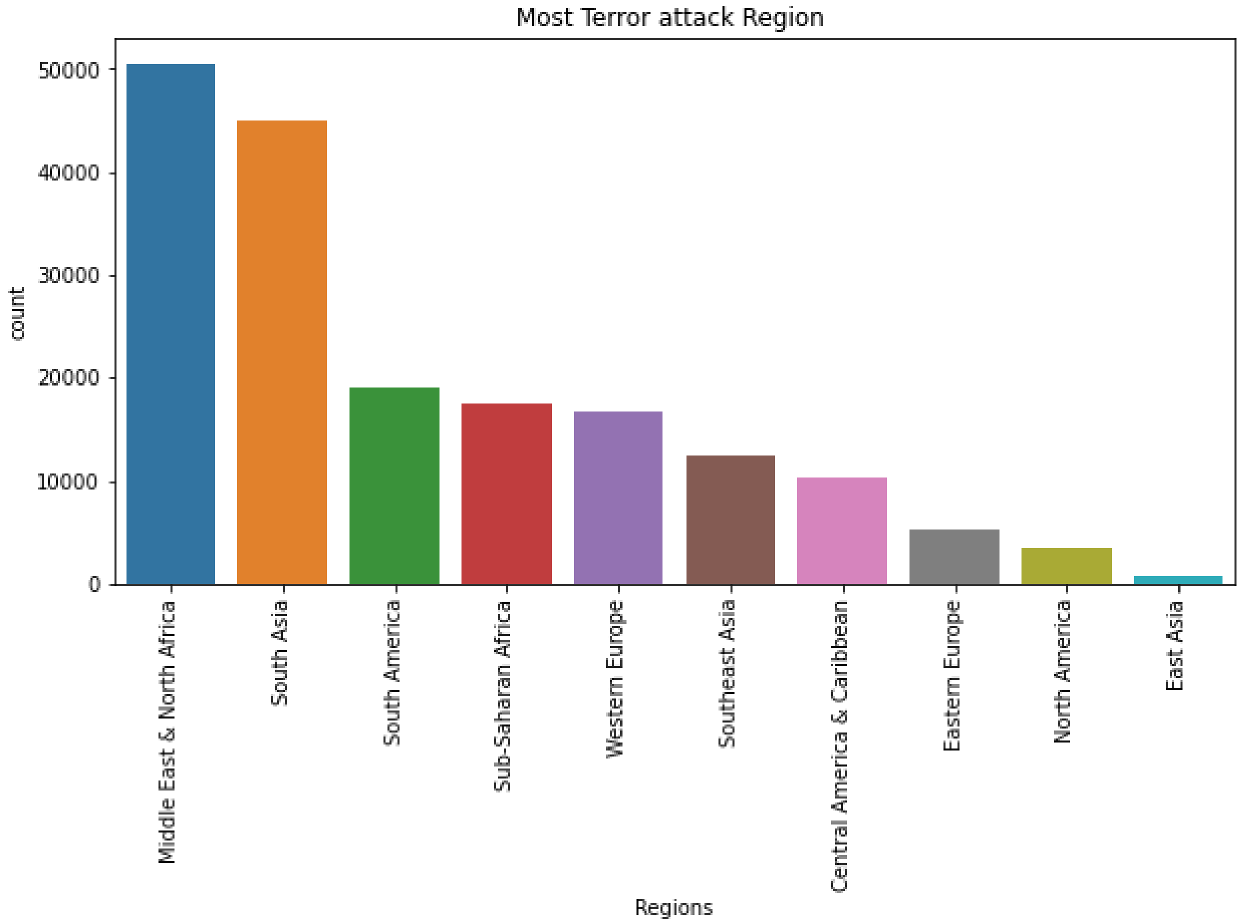
```
Region with most terror attacks:
 Middle East & North Africa     50474
```

```
South Asia                     44974
South America                  18978
Sub-Saharan Africa             17550
Western Europe                 16639
Southeast Asia                 12485
Central America & Caribbean    10344
Eastern Europe                  5144
North America                   3456
East Asia                        802
Name: Region, dtype: int64
```

In [19]:

```
plt.figure(figsize=(10,5))
sns.barplot(df['Region'].value_counts()[:10].index,df['Region'].value_counts()[:10].values)
plt.title("Most Terror attack Region ")
plt.xlabel("Regions")
plt.ylabel('count')
plt.xticks(rotation=90)
plt.show()
```

```
C:\Users\Rajat Kumar\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variabl
es as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



Step 10: Top 10 cities with most Terrorist Attacks

```
In [20]: print('Cities with most terror attacks :\n',df['City'].value_counts().head(10))
```

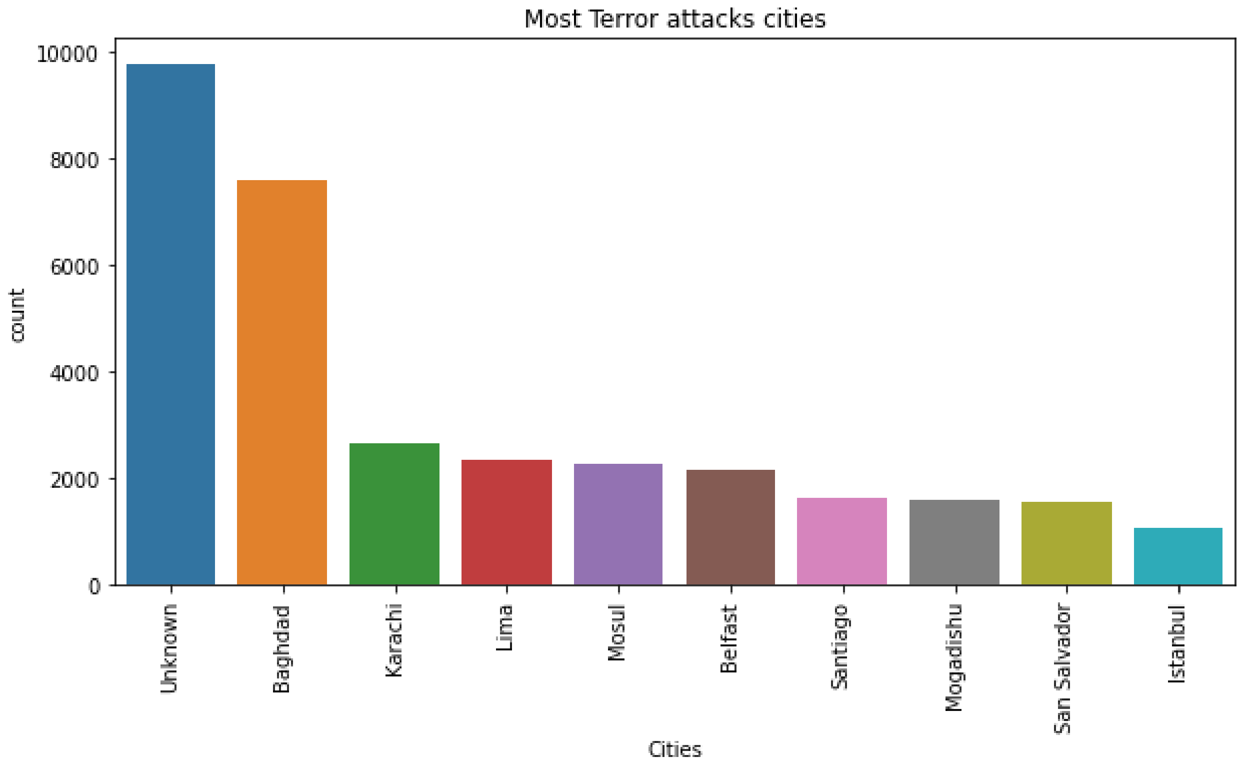
```
Cities with most terror attacks :
 Unknown         9775
Baghdad         7589
Karachi         2652
Lima            2359
Mosul           2265
Belfast         2171
```

```
Santiago        1621
Mogadishu       1581
San Salvador    1558
Istanbul        1048
Name: City, dtype: int64
```

In [21]:

```
plt.figure(figsize=(10,5))
sns.barplot(df['City'].value_counts()[:10].index,df['City'].value_counts()[:10].values)
plt.title("Most Terror attacks cities")
plt.xlabel("Cities")
plt.ylabel('count')
plt.xticks(rotation=90)
plt.show()
```

```
C:\Users\Rajat Kumar\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variabl
es as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



Step 11:Top 20 Years with most Terrorist Activity.

In [22]:

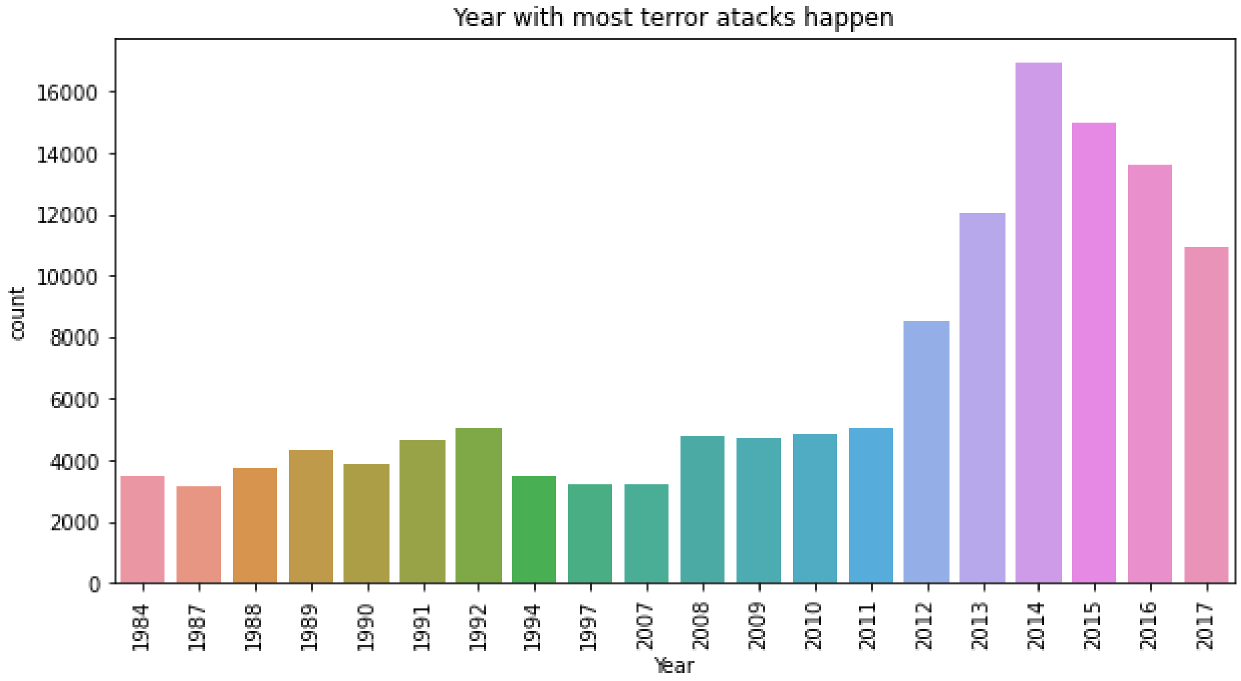
```
print('Year with the most terror attacks:\ ',df['Year'].value_counts().head(20))
```

```
Year with the most terror attacks:\  2014    16903
2015    14965
2016    13587
2013    12036
2017    10900
2012     8522
2011     5076
1992     5071
2010     4826
2008     4805
2009     4721
1991     4683
1989     4324
1990     3887
1988     3721
1984     3495
1994     3456
2007     3242
1997     3197
1987     3183
Name: Year, dtype: int64
```

In [23]:

```
plt.figure(figsize=(10,5))
sns.barplot(df['Year'].value_counts()[:20].index,df['Year'].value_counts()[:20].values)
plt.title("Year with most terror atacks happen")
plt.xlabel("Year")
plt.ylabel('count')
plt.xticks(rotation=90)
plt.show()
```

```
C:\Users\Rajat Kumar\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variabl
es as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



Step 12:Top 10 Most Frequent Attack Type

In [24]:

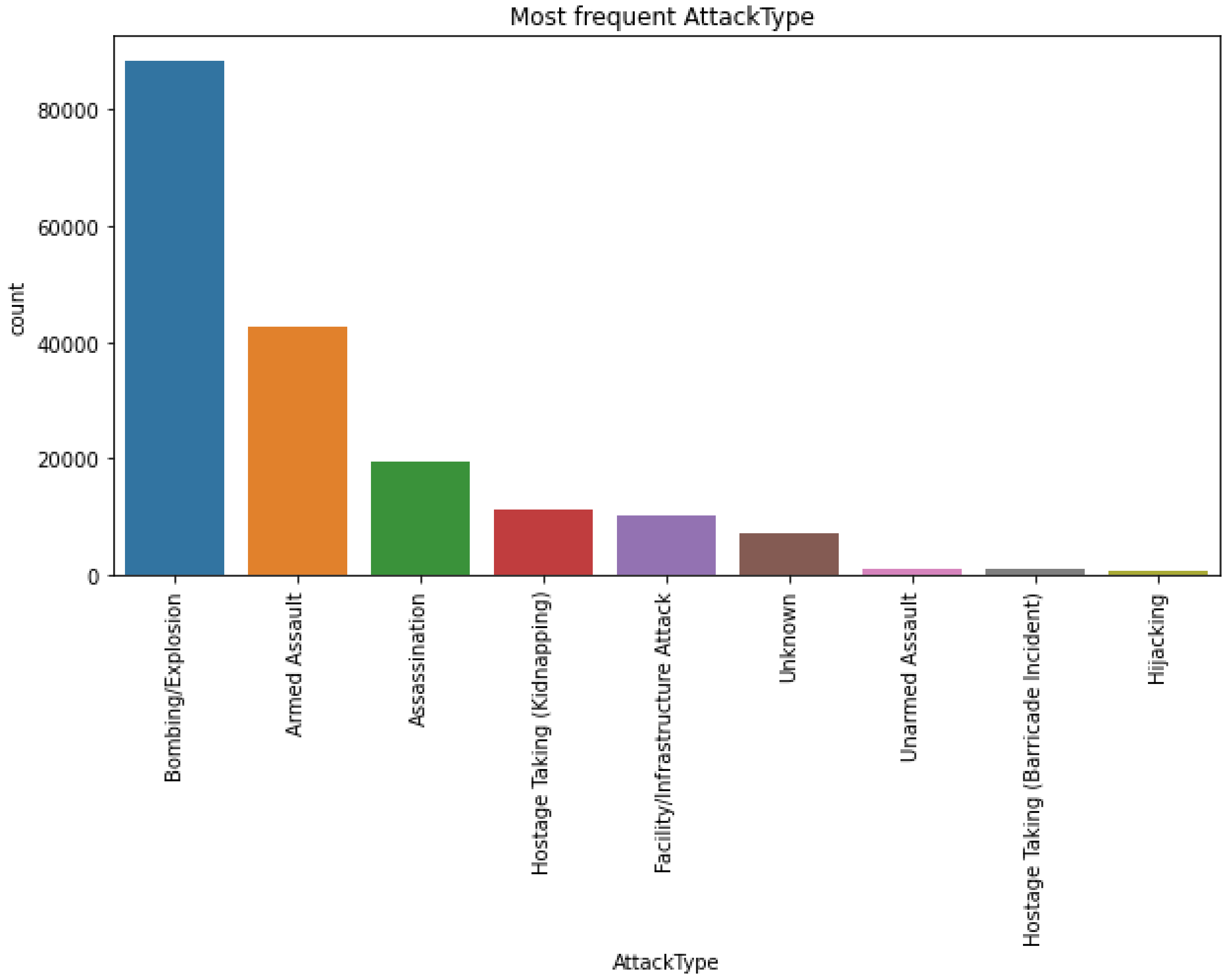
```
print("Most Frequent Attack Type:\n",df['AttackType'].value_counts().head(10))
```

```
Most Frequent Attack Type:
 Bombing/Explosion                      88255
Armed Assault                          42669
Assassination                          19312
Hostage Taking (Kidnapping)            11158
Facility/Infrastructure Attack         10356
Unknown                                 7276
Unarmed Assault                         1015
Hostage Taking (Barricade Incident)      991
Hijacking                                659
Name: AttackType, dtype: int64
```

In [25]:

```
plt.figure(figsize=(10,5))
sns.barplot(df['AttackType'].value_counts()[:10].index,df['AttackType'].value_counts()[:10].values)
plt.title("Most frequent AttackType ")
plt.xlabel("AttackType")
plt.ylabel('count')
plt.xticks(rotation=90)
plt.show()
```

```
C:\Users\Rajat Kumar\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variabl
es as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



Step 13:Top 5 Groups Involved in past Terrorist activity.

In [26]:

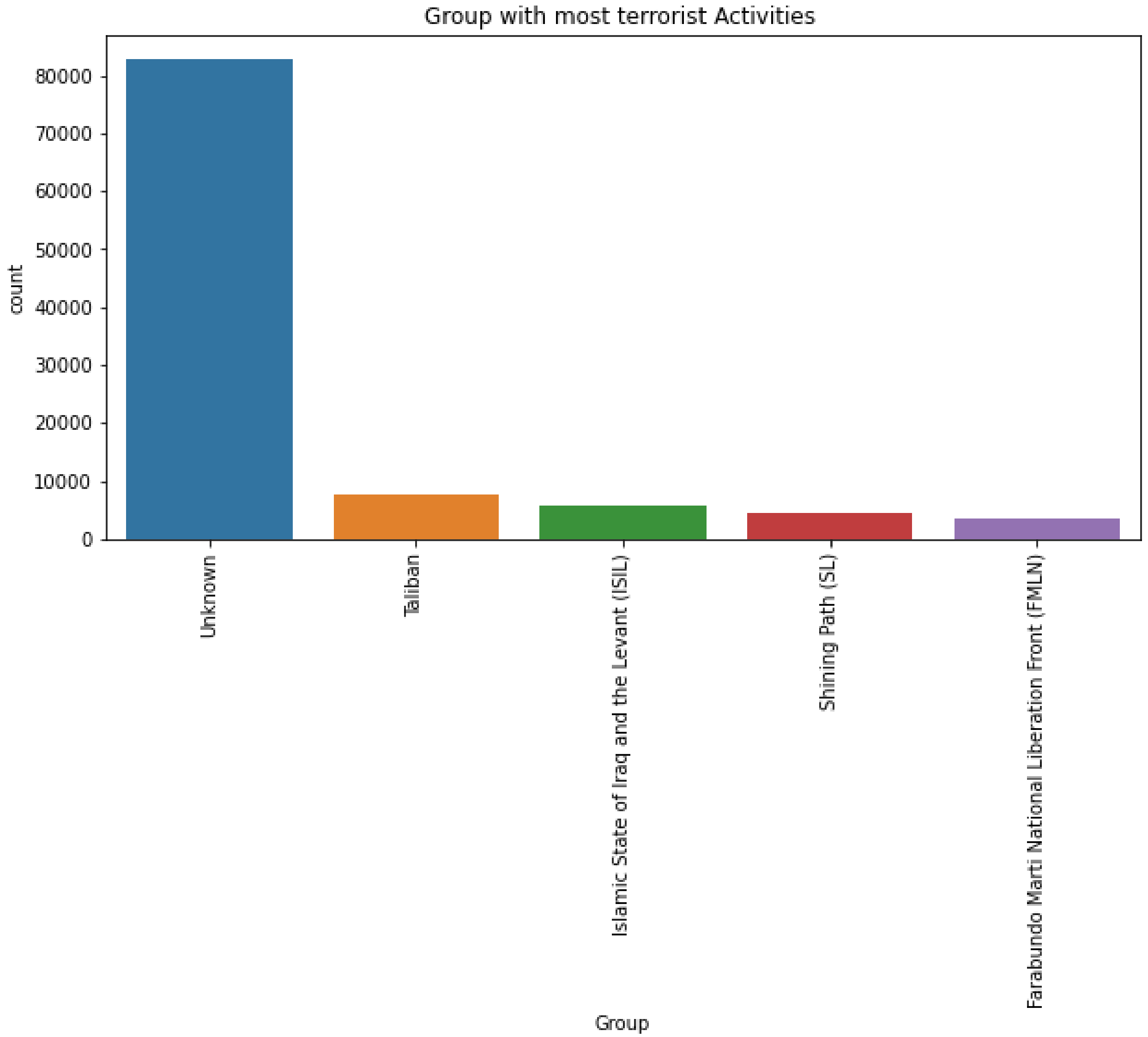
```
print("Group with the most attacks:\n",df['Group'].value_counts().head(5))
```

```
Group with the most attacks:
 Unknown                                             82782
Taliban                                              7478
Islamic State of Iraq and the Levant (ISIL)          5613
Shining Path (SL)                                    4555
Farabundo Marti National Liberation Front (FMLN)     3351
Name: Group, dtype: int64
```

In [27]:

```
plt.figure(figsize=(10,5))
sns.barplot(df['Group'].value_counts()[:5].index,df['Group'].value_counts()[:5].values)
plt.title("Group with most terrorist Activities ")
plt.xlabel("Group")
plt.ylabel('count')
plt.xticks(rotation=90)
plt.show()
```

```
C:\Users\Rajat Kumar\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variabl
es as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



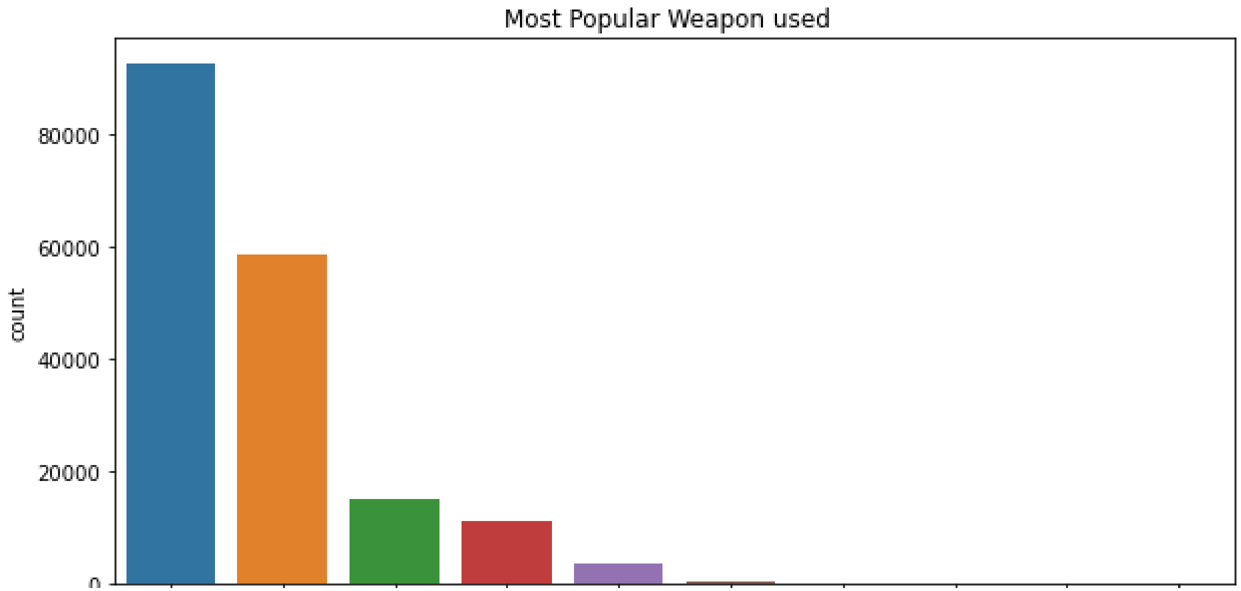
Step 14: Top 10 Weapon Type used in past terrorist Activity.

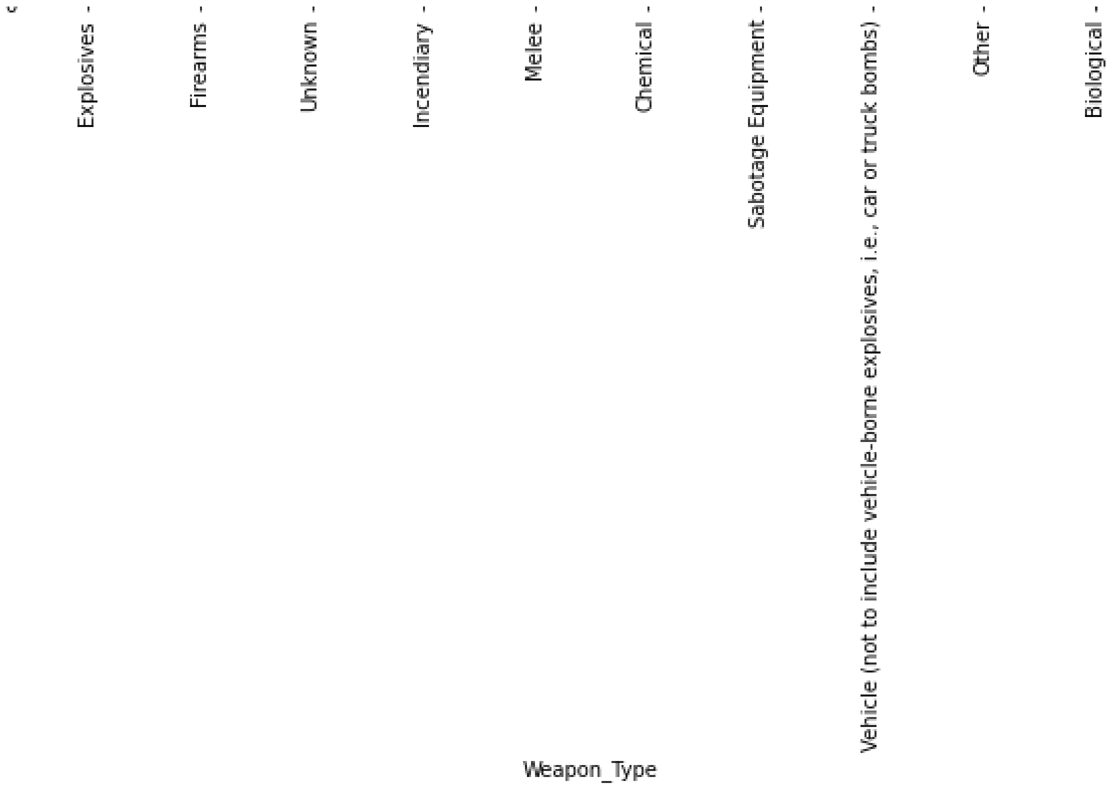
```
In [28]: print('Most popular weapon type used in terror activities are :\n',df['Weapon_Type'].value_counts().head(10))
```

```
Most popular weapon type used in terror activities are :
 Explosives                                                                   92426
Firearms                                                                      58524
Unknown                                                                       15157
Incendiary                                                                    11135
Melee                                                                          3655
Chemical                                                                        321
Sabotage Equipment                                                              141
Vehicle (not to include vehicle-borne explosives, i.e., car or truck bombs)     136
Other                                                                           114
Biological                                                                       35
Name: Weapon_Type, dtype: int64
```

In [29]:
```
plt.figure(figsize=(10,5))
sns.barplot(df['Weapon_Type'].value_counts()[:10].index,df['Weapon_Type'].value_counts()[:10].values)
plt.title("Most Popular Weapon used  ")
plt.xlabel("Weapon_Type")
plt.ylabel('count')
plt.xticks(rotation=90)
plt.show()
```

```
C:\Users\Rajat Kumar\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variabl
es as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arg
uments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

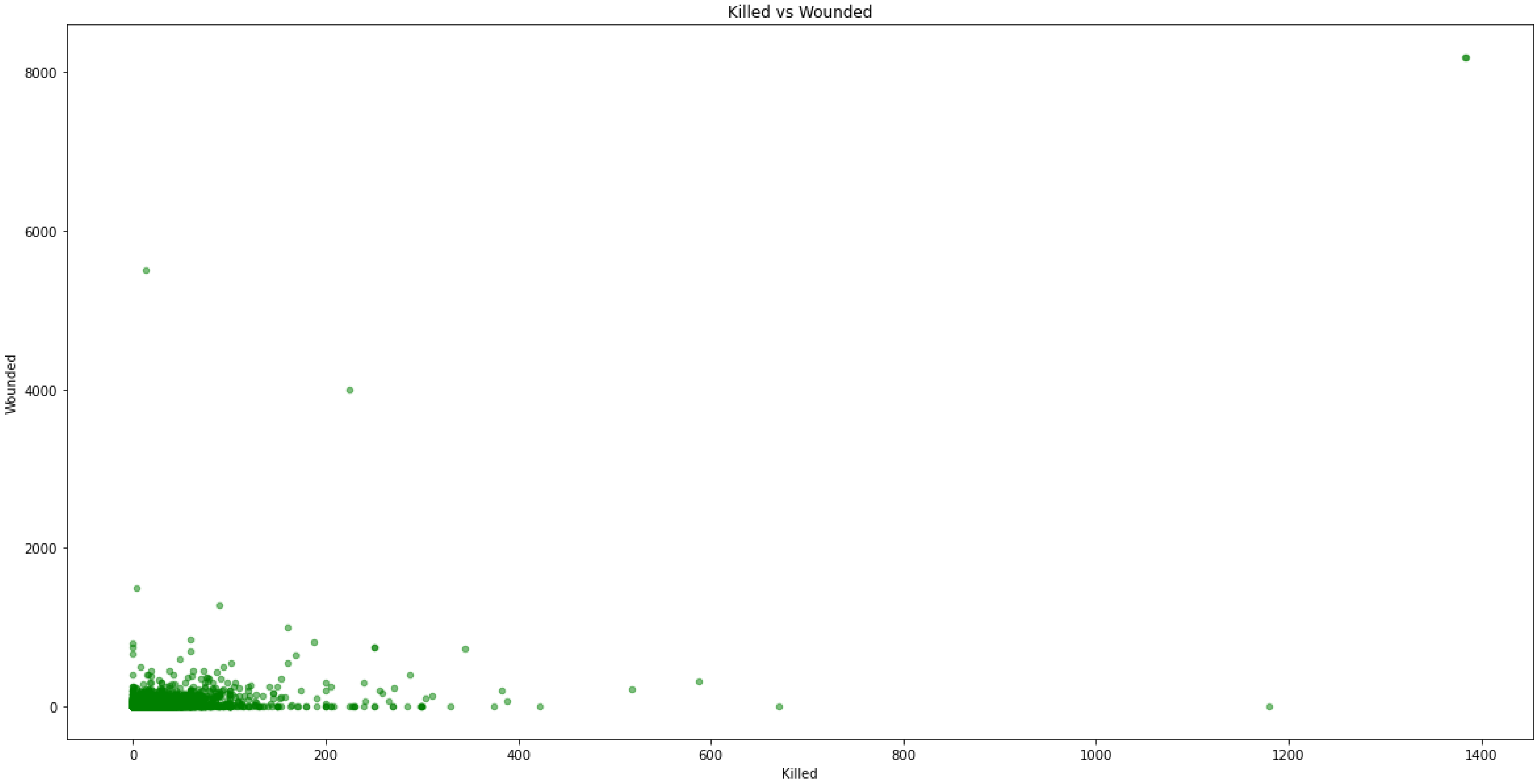




Step 15:Visualization Between Killed VS Wounded

In [30]:

```
df.plot(kind='scatter',x='Killed',y='Wounded',alpha=0.5,color="green",figsize=(20,10))
plt.xlabel('Killed')
plt.ylabel("Wounded")
plt.title("Killed vs Wounded")
plt.show()
```



Conclusion

Hot Zones and year:

1.Top five affected countries are Iraq, Pakistan,Afghanistan, India,Colombia.

2.Top two affected States are Baghdad,Northern Ireland.

3.Top Two most affected region are Middle East & North Africa, South Asia.

4.Most affected year in past is 2014 with total 16903 attacks.

Insights drawn:-

1. The most frequent attack type ia Bombing Explosion so we have to need aleartness in the crowed place.

2. The most popolar weapon use in attacks is Explosives.

Thank you

In [ ]: