Lets Grow More

LGM presents

VIRTUAL

INTERNSHIP PROGRAMME

Task-1 : Exploratory Data Analysis on Dataset - Terrorism(Intermediate Level Task)

As a security/ defense analyst,try to find out the hot zone of terrorism. You can choose any of the tool of your choice(Python/r/Tableau/PowerBI/Excel/SAP/SAS)

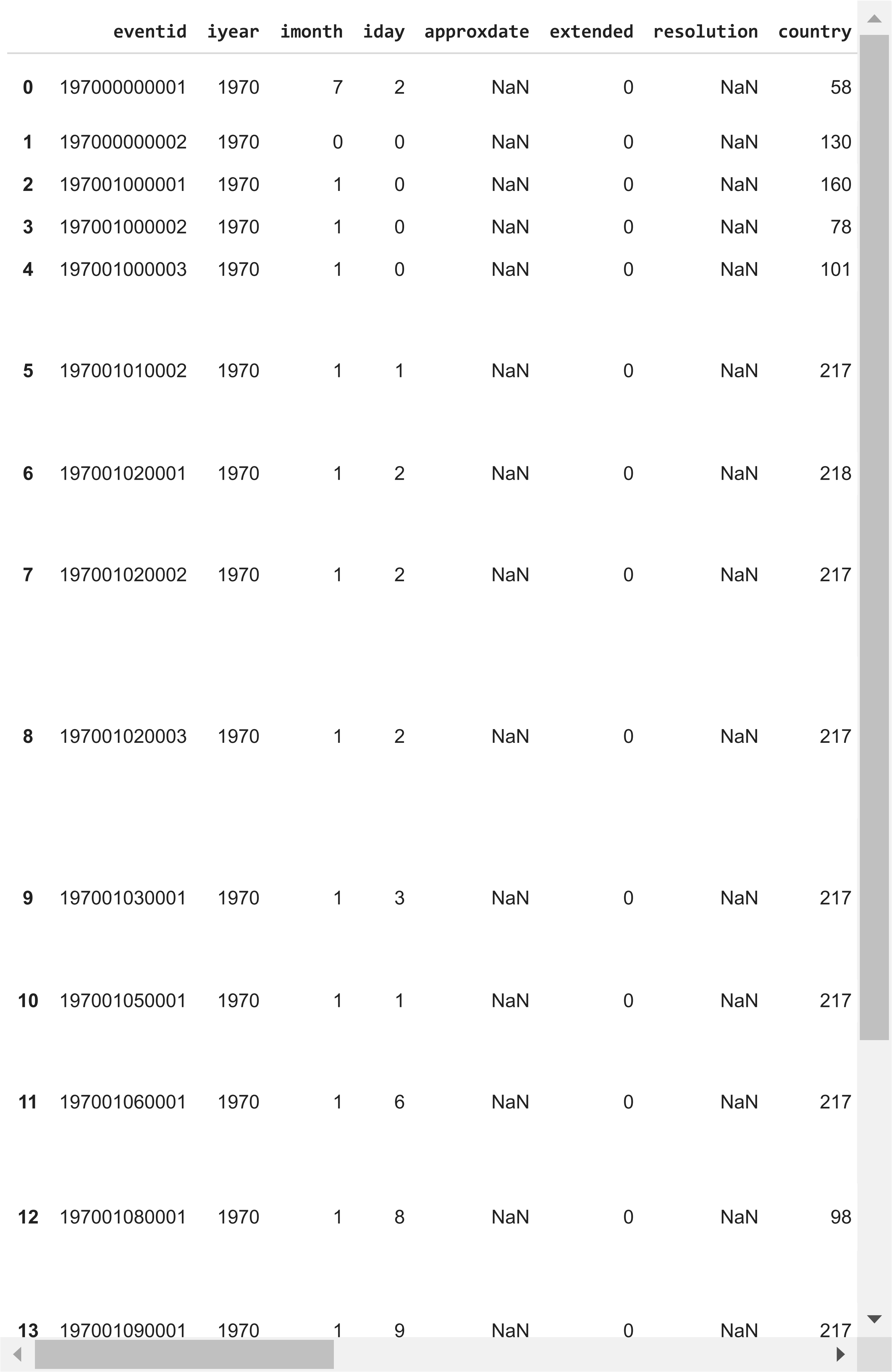
Dataset used link : https://bit.ly/2TK5Xn5

A look into dataset

```
# Import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
#to ignore the irrelevant warnings
import warnings
warnings.filterwarnings('ignore')
```

```
#reading the data
terrorism = pd.read_csv('/content/globalterrorismdb_0718dist.csv',encoding='ISO-8859-1')
```

```
terrorism.head(15)
```



	eventid	iyear	imonth	iday	approxdate	extended	resolution	country
0	197000000001	1970	7	2	NaN	0	NaN	58
1	197000000002	1970	0	0	NaN	0	NaN	130
2	197001000001	1970	1	0	NaN	0	NaN	160
3	197001000002	1970	1	0	NaN	0	NaN	78
4	197001000003	1970	1	0	NaN	0	NaN	101
5	197001010002	1970	1	1	NaN	0	NaN	217
6	197001020001	1970	1	2	NaN	0	NaN	218
7	197001020002	1970	1	2	NaN	0	NaN	217
8	197001020003	1970	1	2	NaN	0	NaN	217
9	197001030001	1970	1	3	NaN	0	NaN	217
10	197001050001	1970	1	1	NaN	0	NaN	217
11	197001060001	1970	1	6	NaN	0	NaN	217
12	197001080001	1970	1	8	NaN	0	NaN	98
13	197001090001	1970	1	9	NaN	0	NaN	217

```
terrorism.columns
```

```
Index(['eventid', 'iyear', 'imonth', 'iday', 'approxdate', 'extended',
       'resolution', 'country', 'country_txt', 'region',
       ...
       'addnotes', 'scite1', 'scite2', 'scite3', 'dbsource', 'INT_LOG',
       'INT_IDEO', 'INT_MISC', 'INT_ANY', 'related'],
      dtype='object', length=135)
```

we have 135 columns , so we'll just select few columns which seems to be important

```
terrorism.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
```

```
#let's Generate descriptive statistics.
#Descriptive statistics include those that summarize the central tendency, dispersion and
#Analyzes both numeric and object series, as well as DataFrame column sets of mixed data t
terrorism.describe()
```

	eventid	iyear	imonth	iday	extended	
count	1.816910e+05	181691.000000	181691.000000	181691.000000	181691.000000	1816
mean	2.002705e+11	2002.638997	6.467277	15.505644	0.045346	1
std	1.325957e+09	13.259430	3.388303	8.814045	0.208063	1
min	1.970000e+11	1970.000000	0.000000	0.000000	0.000000	
25%	1.991021e+11	1991.000000	4.000000	8.000000	0.000000	
50%	2.009022e+11	2009.000000	6.000000	15.000000	0.000000	
75%	2.014081e+11	2014.000000	9.000000	23.000000	0.000000	1
max	2.017123e+11	2017.000000	12.000000	31.000000	1.000000	10

8 rows × 77 columns

```
terrorism=terrorism[['iyear','imonth','iday','country_txt','provstate','region_txt','city'
```

```
#let's check if we got some null values or not
terrorism.isnull().sum()
```

```
iyear                   0
imonth                  0
iday                    0
country_txt             0
provstate             421
region_txt              0
city                  434
latitude             4556
longitude            4557
attacktype1_txt         0
target1               636
nkill               10313
nwound              16311
summary             66129
gname                   0
targtype1_txt           0
weaptype1_txt           0
motive             131130
dtype: int64
```

```
#looks like we got some null values in provstate,city,latitude,longitude,target1,nkill,nwc
terrorism.dropna(axis=0, inplace=True)
```

```
terrorism.isnull().sum()
```

```
iyear              0
imonth             0
iday               0
country_txt        0
provstate          0
region_txt         0
city               0
latitude           0
longitude          0
attacktype1_txt    0
target1            0
nkill              0
nwound             0
summary            0
gname              0
targtype1_txt      0
weaptype1_txt      0
motive             0
dtype: int64
```

Visualisation of data

```
print("Country with the most attacks:",terrorism['country_txt'].value_counts().idxmax())
print("City with the most attacks:",terrorism['city'].value_counts().idxmax())
print("Region with the most attacks:",terrorism['region_txt'].value_counts().idxmax())
print("Year with the most attacks:",terrorism['iyear'].value_counts().idxmax())
print("Month with the most attacks:",terrorism['imonth'].value_counts().idxmax())
print("Group with the most attacks:",terrorism['gname'].value_counts().index[1])
print("Most Attack Types:",terrorism['attacktype1_txt'].value_counts().idxmax())
```

```
     Country with the most attacks: Iraq
     City with the most attacks: Baghdad
     Region with the most attacks: South Asia
     Year with the most attacks: 2011
     Month with the most attacks: 7
     Group with the most attacks: Taliban
     Most Attack Types: Bombing/Explosion
```

year wise terror attacks in increasing order

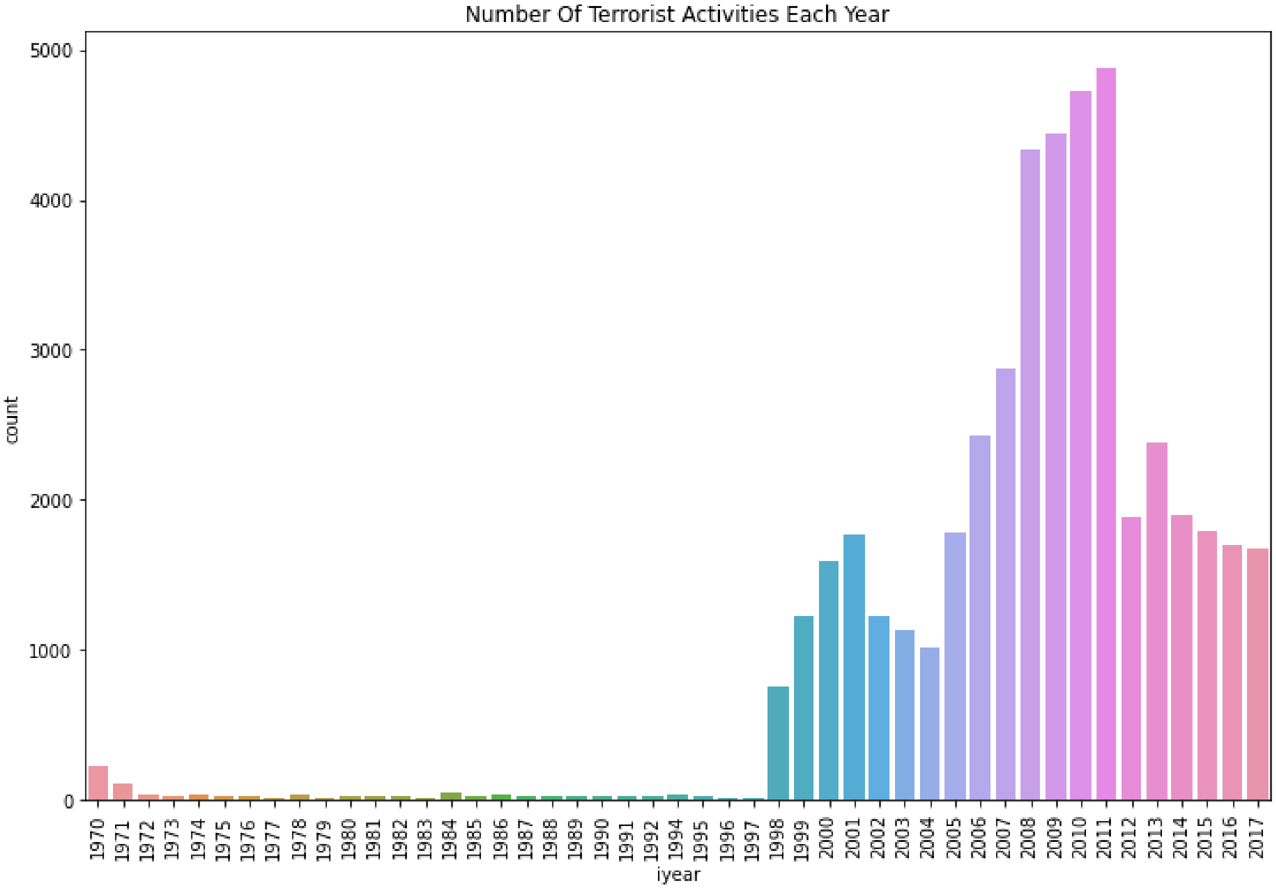
```
terrorism['iyear'].value_counts(dropna = False).sort_index()
```

```
     1970     222
     1971     111
     1972      40
     1973      25
     1974      40
     1975      26
     1976      30
     1977      18
     1978      41
     1979      13
     1980      23
     1981      22
     1982      27
     1983      20
     1984      46
     1985      26
     1986      37
     1987      27
     1988      28
     1989      29
     1990      27
     1991      28
     1992      28
     1994      34
     1995      25
     1996      13
     1997      20
     1998     759
     1999    1231
     2000    1597
     2001    1771
     2002    1223
     2003    1139
     2004    1012
     2005    1781
     2006    2424
     2007    2881
     2008    4341
     2009    4447
     2010    4722
```

```
2011    4876
2012    1890
2013    2387
2014    1898
2015    1788
2016    1693
2017    1670
Name: iyear, dtype: int64
```

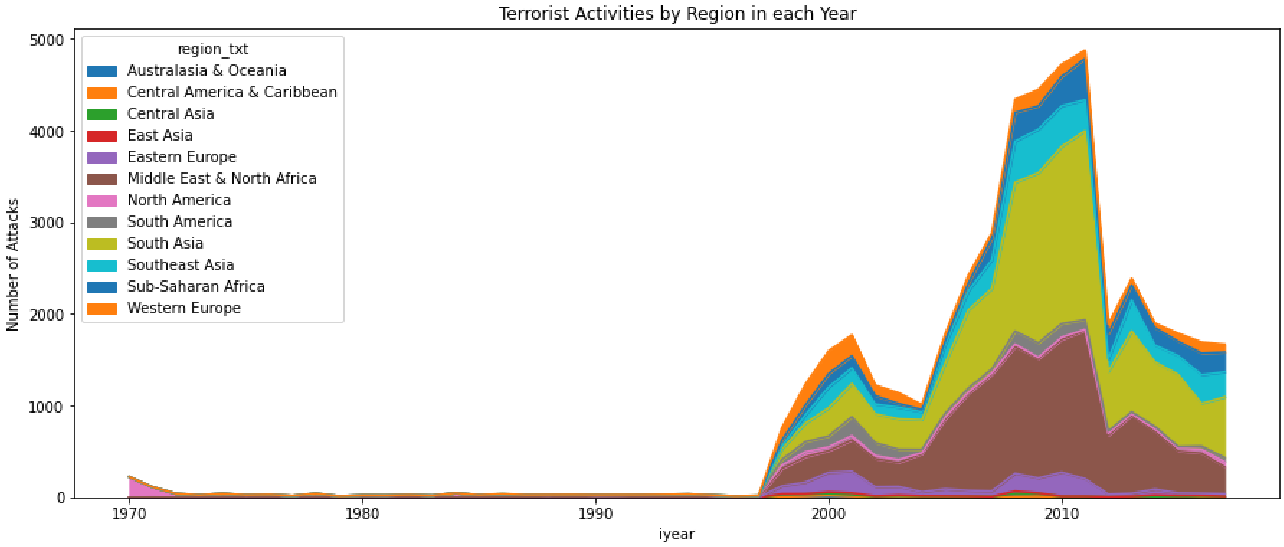
lets visualise yearly count of the attacks

```
plt.figure(figsize=(12,8))
sns.countplot(x="iyear", data=terrorism)
plt.xticks(rotation=90)
plt.title('Number Of Terrorist Activities Each Year')
plt.show()
```



we can see clearly that 2011 has most attacks compared to others

```
pd.crosstab(terrorism.iyear, terrorism.region_txt).plot(kind='area',figsize=(15,6))
plt.title('Terrorist Activities by Region in each Year')
plt.ylabel('Number of Attacks')
plt.show()
```



This is Terrorist Activities by Region in each Year

```
terrorism['nwound'] = terrorism['nwound'].fillna(0).astype(int)
terrorism['nkill'] = terrorism['nkill'].fillna(0).astype(int)
terrorism['casualities'] = terrorism['nkill'] + terrorism['nwound']
```

Values are sorted by the top 50 worst terror attacks as to keep the heatmap simple and easy to visualize

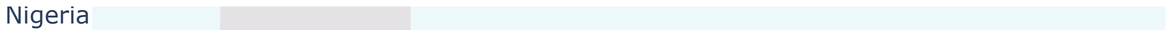
```
terror1 = terrorism.sort_values(by='casualities',ascending=False)[:50]
```

```
heat=terror1.pivot_table(index='country_txt',columns='iyear',values='casualities')
heat.fillna(0,inplace=True)
heat.head(20)
```

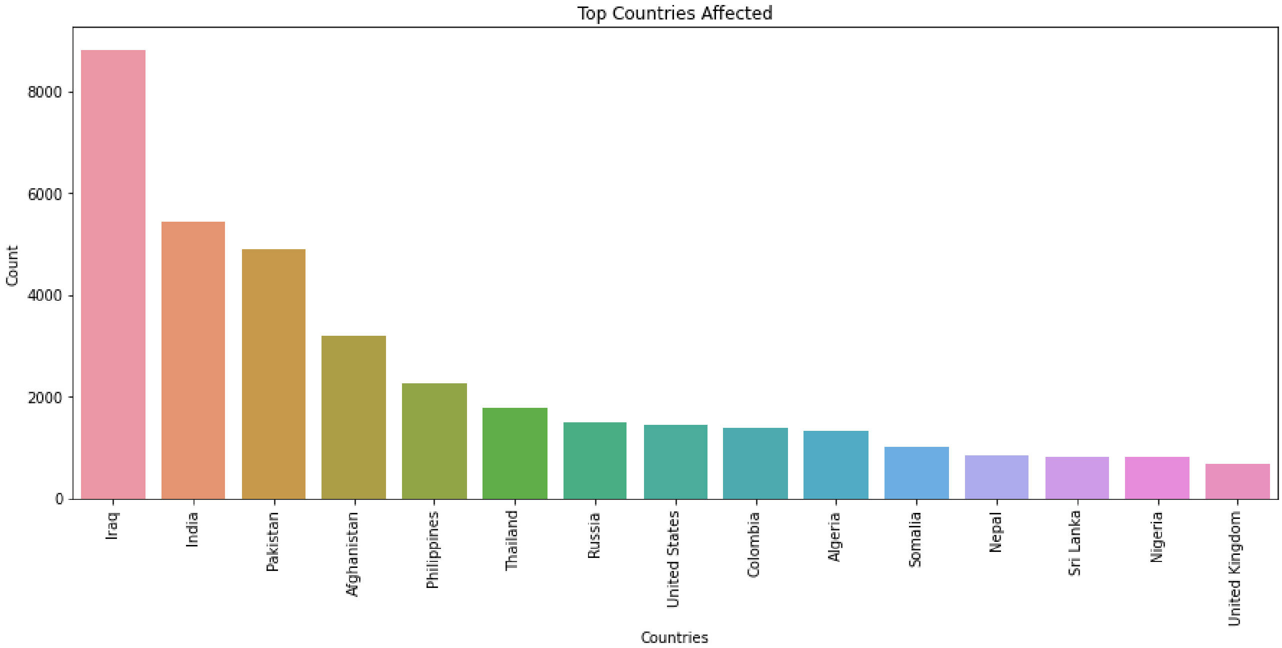
iyear	1998	1999	2001	2002	2003	2004	2005	2006	2007	20
country_txt										
Afghanistan	0.0	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.000000	0
Angola	0.0	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.000000	0
Bangladesh	0.0	0.0	0.0	317.0	0.0	0.00	0.0	0.0	0.000000	0
Chad	0.0	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.000000	1161
Egypt	0.0	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.000000	0
France	0.0	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.000000	0
India	0.0	0.0	0.0	0.0	0.0	0.00	0.0	1005.0	0.000000	0
Iran	0.0	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.000000	0
Iraq	0.0	0.0	0.0	0.0	300.0	343.00	702.0	387.0	556.571429	0
Kenya	4224.0	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.000000	0
Libya	0.0	0.0	0.0	0.0	0.0	0.00	0.0	0.0	0.000000	0
Nepal	0.0	0.0	0.0	0.0	0.0	734.00	0.0	0.0	0.000000	0

```
import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objects as go
colorscale = [[0, '#edf8fb'], [.3, '#A52A2A'],  [.6,'#76EE00'],  [1, '#008B8B']]
heatmap = go.Heatmap(z=heat.values, x=heat.columns, y=heat.index, colorscale=colorscale)
data = [heatmap]
layout = go.Layout(
    title='Top 50 Worst Terror Attacks in History from 1998 to 2016',
    xaxis = dict(ticks='', nticks=20),
    yaxis = dict(ticks='')
)
fig = go.Figure(data=data, layout=layout)
fig.show(renderer="colab")
```

Top Countries affected by terror attacks



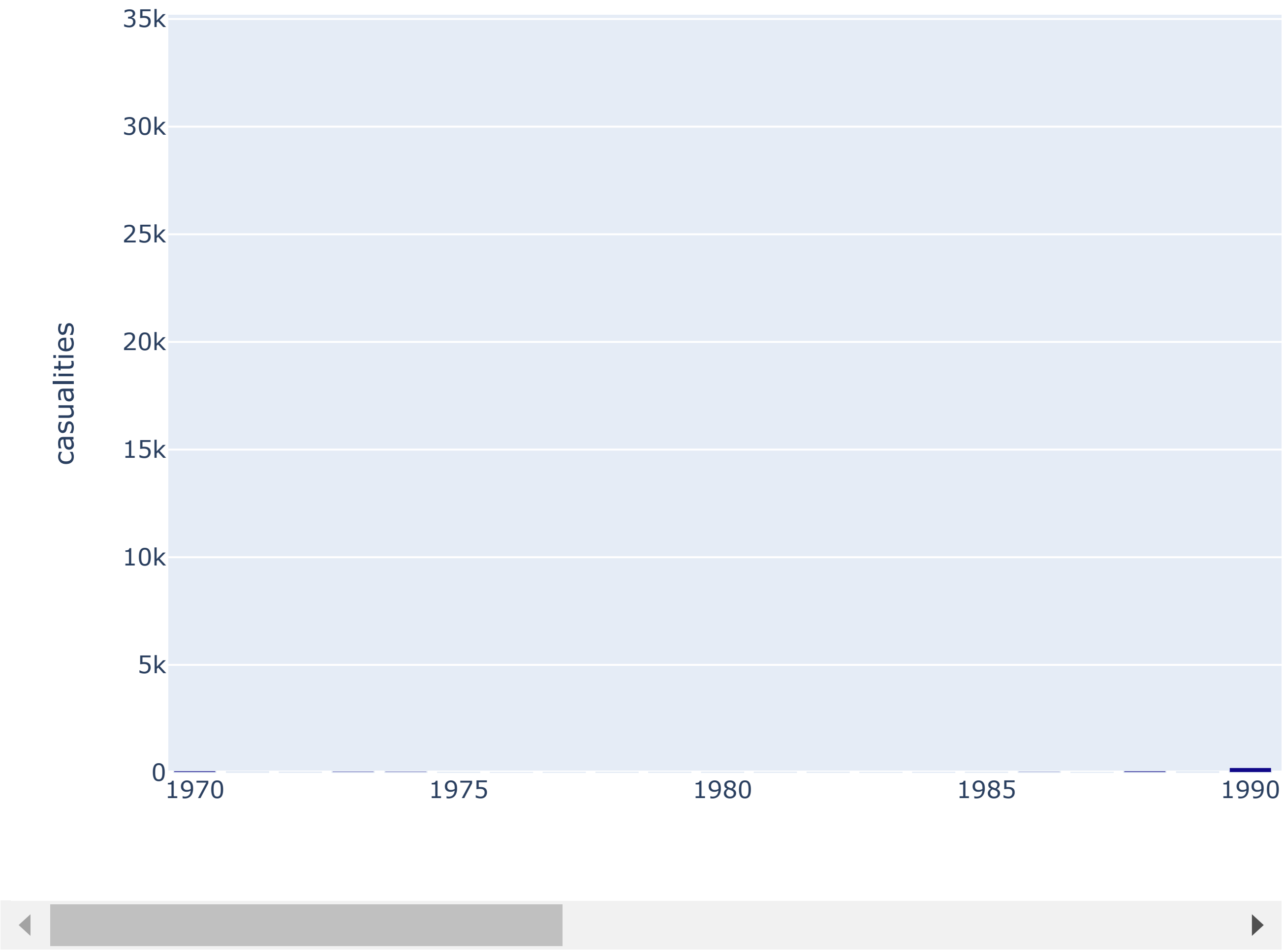
```
plt.subplots(figsize=(15,6))
sns.barplot(terrorism['country_txt'].value_counts()[:15].index,terrorism['country_txt'].va
plt.title('Top Countries Affected')
plt.xlabel('Countries')
plt.ylabel('Count')
plt.xticks(rotation= 90)
plt.show()
```



yearly casualities

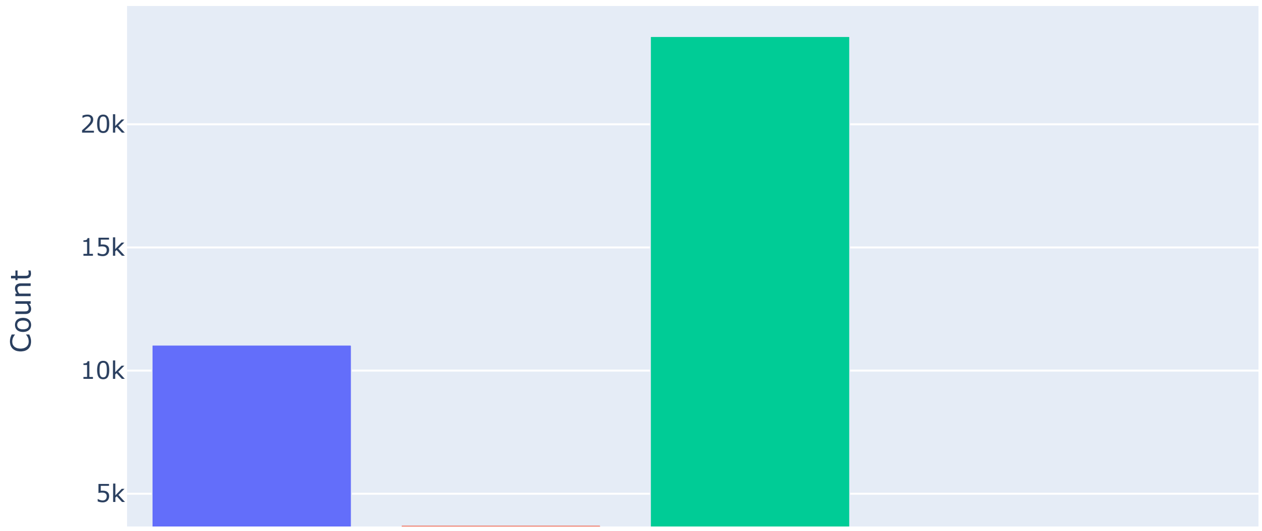
```
import plotly.express as px
year_cas = terrorism.groupby('iyear').casualities.sum().to_frame().reset_index()
```

```
year_cas.columns = ['iyear','casualities']
fig=px.bar(data_frame=year_cas,x = 'iyear',y = 'casualities',color='casualities')
fig.show(renderer="colab")
```



it is observed that in between 2005-2010 we have highest casualities

```
from collections import Counter
target = list(terrorism['attacktype1_txt'])
target_map = dict(Counter(target))
target_df = pd.DataFrame(target_map.items())
target_df.columns = ['attacktype1_txt','Count']
fig=px.bar(data_frame=target_df,x = 'attacktype1_txt',y = 'Count',color='attacktype1_txt')
fig.show(renderer="colab")
```



```
killData = terrorism.loc[:,'nkill']
print('Number of people killed by terror attack:', int(sum(killData.dropna())))
```

```
Number of people killed by terror attack: 112679
```

```
# Let's look at what types of attacks these deaths were made of.
attackData = terrorism.loc[:,'attacktype1_txt']
# attackData
typeKillData = pd.concat([attackData, killData], axis=1)
typeKillData.head()
```

	attacktype1_txt	nkill
5	Armed Assault	0
8	Facility/Infrastructure Attack	0
9	Facility/Infrastructure Attack	0
11	Facility/Infrastructure Attack	0
14	Facility/Infrastructure Attack	0

```
typeKillFormatData = typeKillData.pivot_table(columns='attacktype1_txt', values='nkill', a
typeKillFormatData
```

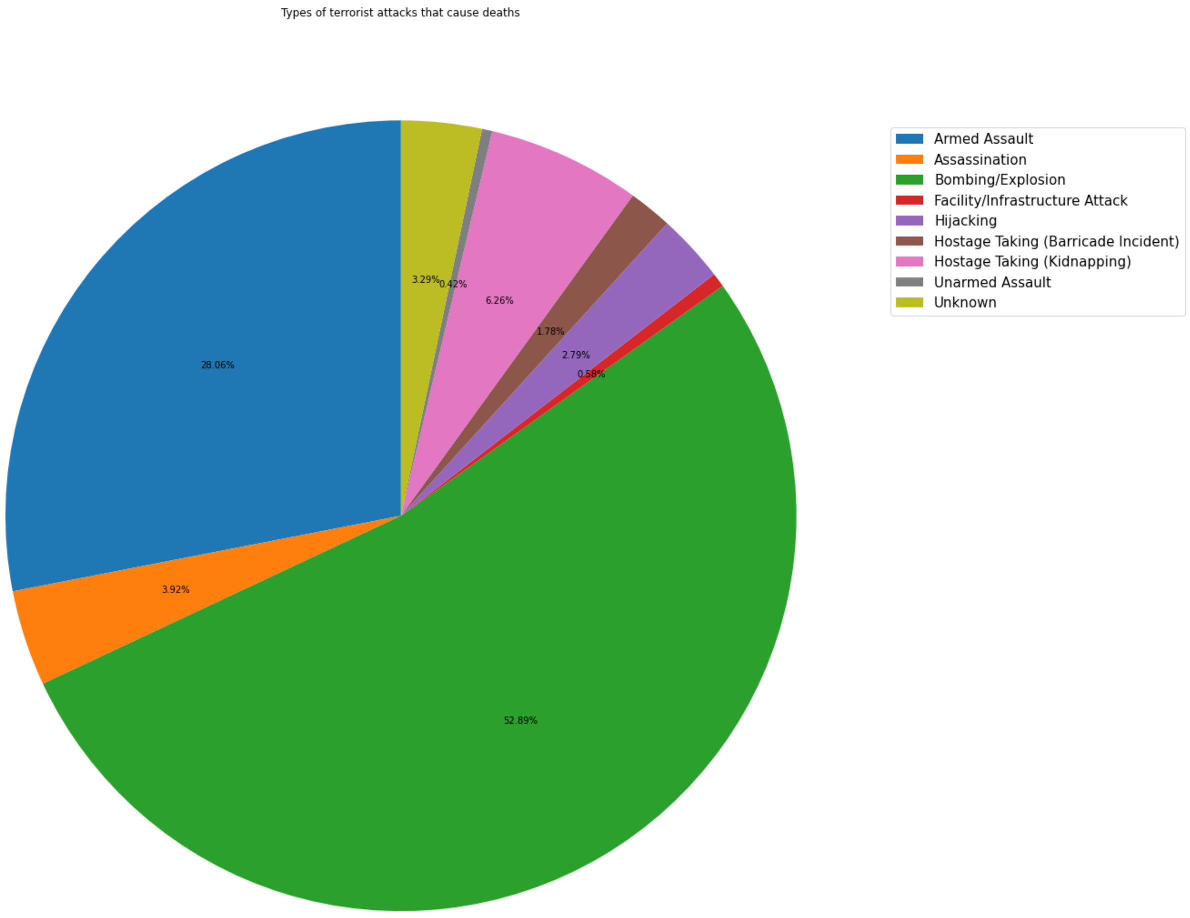
attacktype1_txt	Armed Assault	Assassination	Bombing/Explosion	Facility/Infrastructure Attack
nkill	31619	4419	59600	652

```
typeKillFormatData.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
Index: 1 entries, nkill to nkill
Data columns (total 9 columns):
 #   Column                             Non-Null Count  Dtype
---  ------                             --------------  -----
 0   Armed Assault                      1 non-null      int64
 1   Assassination                      1 non-null      int64
 2   Bombing/Explosion                  1 non-null      int64
 3   Facility/Infrastructure Attack     1 non-null      int64
 4   Hijacking                          1 non-null      int64
 5   Hostage Taking (Barricade Incident) 1 non-null      int64
 6   Hostage Taking (Kidnapping)        1 non-null      int64
 7   Unarmed Assault                    1 non-null      int64
 8   Unknown                            1 non-null      int64
dtypes: int64(9)
memory usage: 188.0+ bytes
```

```
labels = typeKillFormatData.columns.tolist() # convert line to list
transpoze = typeKillFormatData.T # transpoze
values = transpoze.values.tolist()
fig, ax = plt.subplots(figsize=(20, 20), subplot_kw=dict(aspect="equal"))
plt.pie(values, startangle=90, autopct='%.2f%%')
plt.title('Types of terrorist attacks that cause deaths')
plt.legend(labels, loc='upper right', bbox_to_anchor = (1.3, 0.9), fontsize=15) # location
plt.show()
```



Armed assault and bombing/explosion are seen to be the cause of 80.95% of the deaths in these attacks. This rate is why these attacks are used so many times in terrorist actions. This is how dangerous weapons and explosives are to the world.

Number of people killed in terrorist attacks by countries

```
#Number of Killed in Terrorist Attacks by Countries
countryData = terrorism.loc[:,'country_txt']
# countyData
countryKillData = pd.concat([countryData, killData], axis=1)


countryKillFormatData = countryKillData.pivot_table(columns='country_txt', values='nkill',
countryKillFormatData
```

```
country_txt  Afghanistan  Albania  Algeria  Angola  Argentina  Armenia  Australia  A
```

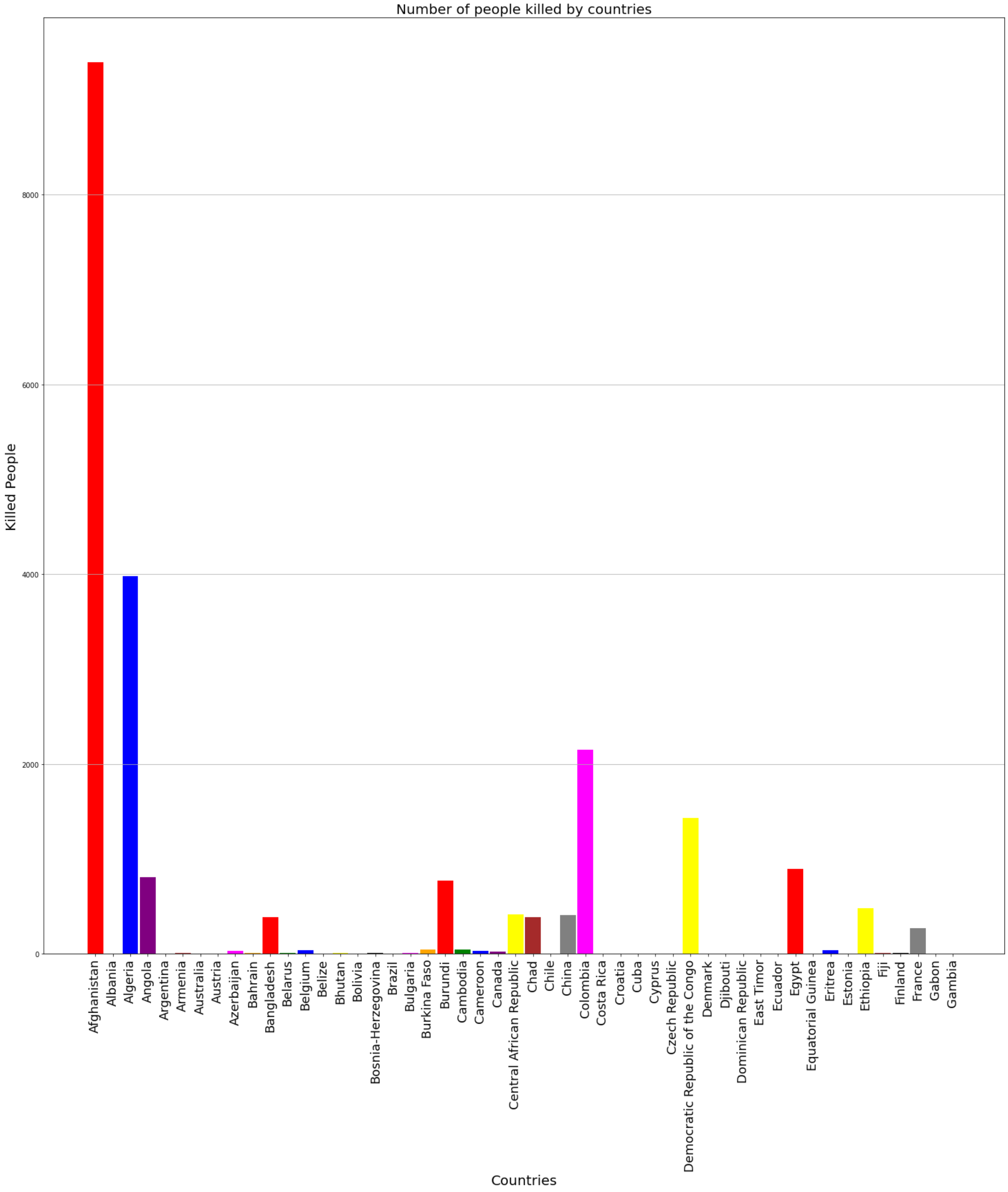
```
fig_size = plt.rcParams["figure.figsize"]
fig_size[0]=25
fig_size[1]=25
plt.rcParams["figure.figsize"] = fig_size
```

```
count_countries=countryKillFormatData.columns.tolist()
len(count_countries)
```

```
164
```

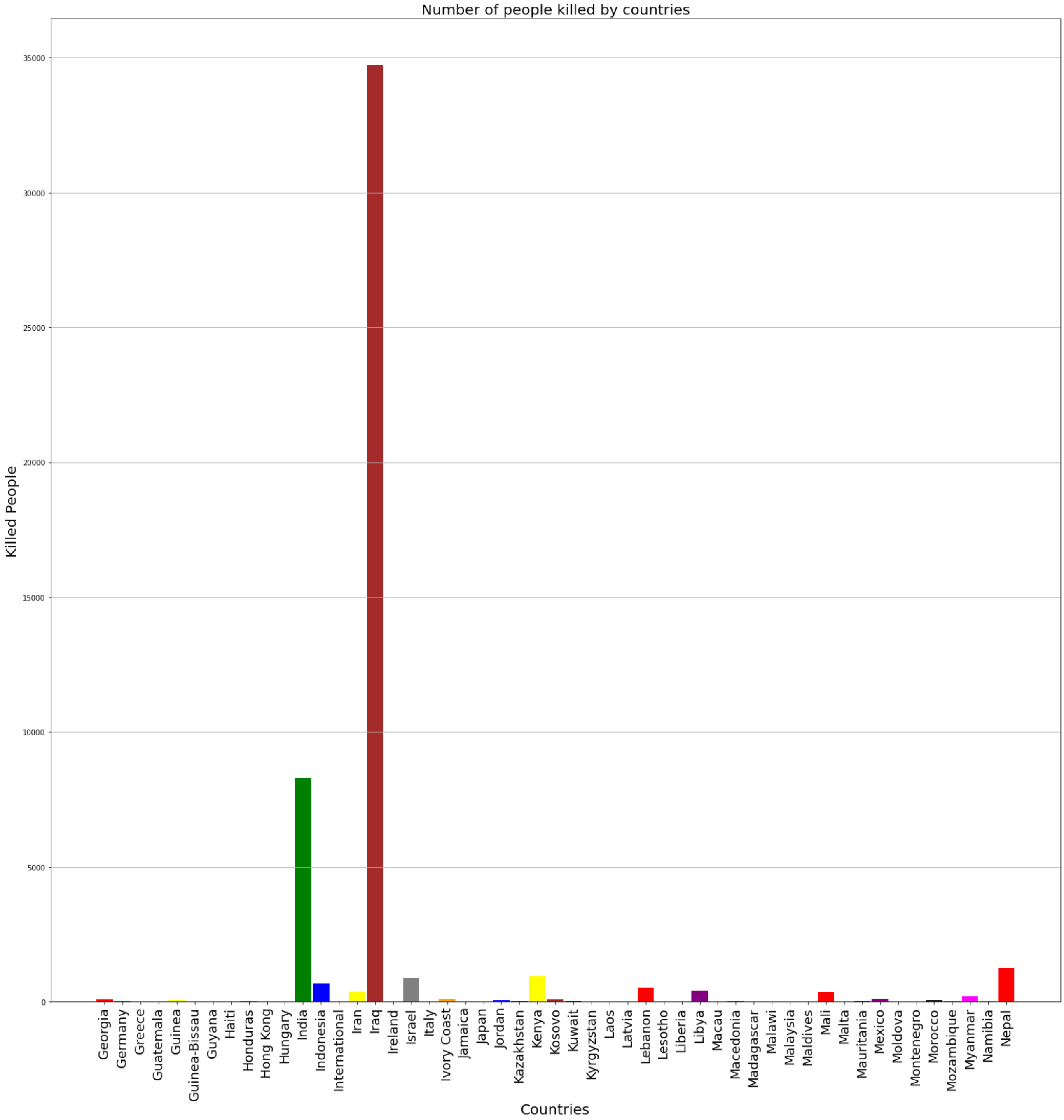
total 164 countries are to be plotted here

```
labels = countryKillFormatData.columns.tolist()
labels = labels[:50] #50 bar is easier to view at once
index = np.arange(len(labels))
transpoze = countryKillFormatData.T
values = transpoze.values.tolist()
values = values[:50]
values = [int(i[0]) for i in values] # convert float to int
colors = ['red', 'green', 'blue', 'purple', 'yellow', 'brown', 'black', 'gray', 'magenta',
fig, ax = plt.subplots(1, 1)
ax.yaxis.grid(True)
fig_size = plt.rcParams["figure.figsize"]
fig_size[0]=25
fig_size[1]=25
plt.rcParams["figure.figsize"] = fig_size
plt.bar(index, values, color = colors, width = 0.9)
plt.ylabel('Killed People', fontsize=20)
plt.xlabel('Countries', fontsize = 20)
plt.xticks(index, labels, fontsize=18, rotation=90)
plt.title('Number of people killed by countries', fontsize = 20)
# print(fig_size)
plt.show()
```


Number of people killed by countries
Killed People
0
2000
4000
6000
8000
Countries
Afghanistan
Albania
Algeria
Angola
Argentina
Armenia
Australia
Austria
Azerbaijan
Bahrain
Bangladesh
Belarus
Belgium
Belize
Bhutan
Bolivia
Bosnia-Herzegovina
Brazil
Bulgaria
Burkina Faso
Burundi
Cambodia
Cameroon
Canada
Central African Republic
Chad
Chile
China
Colombia
Costa Rica
Croatia
Cuba
Cyprus
Czech Republic
Democratic Republic of the Congo
Denmark
Djibouti
Dominican Republic
East Timor
Ecuador
Egypt
Equatorial Guinea
Eritrea
Estonia
Ethiopia
Fiji
Finland
France
Gabon
Gambia

Afghanistan,Algeria,Colombia has most of the attacks compared to others

```
labels = countryKillFormatData.columns.tolist()
labels = labels[50:101]
index = np.arange(len(labels))
transpoze = countryKillFormatData.T
values = transpoze.values.tolist()
values = values[50:101]
values = [int(i[0]) for i in values]
colors = ['red', 'green', 'blue', 'purple', 'yellow', 'brown', 'black', 'gray', 'magenta',
fig, ax = plt.subplots(1, 1)
ax.yaxis.grid(True)
fig_size = plt.rcParams["figure.figsize"]
fig_size[0]=20
fig_size[1]=20
plt.rcParams["figure.figsize"] = fig_size
plt.bar(index, values, color = colors, width = 0.9)
plt.ylabel('Killed People', fontsize=20)
plt.xlabel('Countries', fontsize = 20)
plt.xticks(index, labels, fontsize=18, rotation=90)
plt.title('Number of people killed by countries', fontsize = 20)
plt.show()
```


Number of people killed by countries
Killed People
0
5000
10000
15000
20000
25000
30000
35000
Countries
Georgia
Germany
Greece
Guatemala
Guinea
Guinea-Bissau
Guyana
Haiti
Honduras
Hong Kong
Hungary
India
Indonesia
International
Iran
Iraq
Ireland
Israel
Italy
Ivory Coast
Jamaica
Japan
Jordan
Kazakhstan
Kenya
Kosovo
Kuwait
Kyrgyzstan
Laos
Latvia
Lebanon
Lesotho
Liberia
Libya
Macau
Macedonia
Madagascar
Malawi
Malaysia
Maldives
Mali
Malta
Mauritania
Mexico
Moldova
Montenegro
Morocco
Mozambique
Myanmar
Namibia
Nepal

Iraq,India has most of the attacks compared to others

```
labels = countryKillFormatData.columns.tolist()
labels = labels[152:165]
index = np.arange(len(labels))
transpoze = countryKillFormatData.T
values = transpoze.values.tolist()
values = values[152:206]
values = [int(i[0]) for i in values]
colors = ['red', 'green', 'blue', 'purple', 'yellow', 'brown', 'black', 'gray', 'magenta',
fig, ax = plt.subplots(1, 1)
ax.yaxis.grid(True)
fig_size = plt.rcParams["figure.figsize"]
fig_size[0]=25
fig_size[1]=25
plt.rcParams["figure.figsize"] = fig_size
plt.bar(index, values, color = colors, width = 0.9)
plt.ylabel('Killed People', fontsize=20)
plt.xlabel('Countries', fontsize = 20)
plt.xticks(index, labels, fontsize=18, rotation=90)
plt.title('Number of people killed by countries', fontsize = 20)
plt.show()
```