

```
In [1]: pip install plotly
```

Requirement already satisfied: plotly in c:\users\user\appdata\local\programs\python\python310\lib\site-packages (5.13.1)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\user\appdata\local\programs\python\python310\lib\site-packages (from plotly) (8.2.2)
Note: you may need to restart the kernel to use updated packages.

```
In [2]: import math  
import warnings  
import numpy as np  
import pandas as pd  
import seaborn as sns  
import matplotlib.pyplot as plt  
warnings.filterwarnings('ignore')
```

```
In [3]: data = pd.read_csv(r'D:\terrorism.csv',encoding='ISO-8859-1')  
data.head()
```

```
Out[3]:
```

	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	country_txt	region
0	1.970000e+11	1970	7	2	NaN	0	NaN	58	Dominican Republic	2
1	1.970000e+11	1970	0	0	NaN	0	NaN	130	Mexico	1
2	1.970010e+11	1970	1	0	NaN	0	NaN	160	Philippines	5
3	1.970010e+11	1970	1	0	NaN	0	NaN	78	Greece	8
4	1.970010e+11	1970	1	0	NaN	0	NaN	101	Japan	4

5 rows × 135 columns

```
In [4]: data.isnull().sum()
```

```
Out[4]:
```

eventid	0
iyear	0
imonth	0
iday	0
approxdate	172452
...	
INT_LOG	0
INT_IDEO	0
INT_MISC	0
INT_ANY	0
related	156653
Length:	135, dtype: int64

```
In [5]: data.rename(columns={'iyear':'Year','imonth':'Month','iday':'Day','country_txt':'Country','region_txt':'Region','attacktype1_txt':'AttackType','target1':'Target','nwound':'Wounded','summary':'Summary','gname':'Group','targtype1_txt':'Weaptype1_txt','weaptype1_txt':'Weapon_type','motive':'Motive'},inplace=True)
```

```
In [6]: data.columns
```

```
Out[6]: Index(['eventid', 'Year', 'Month', 'Day', 'approxdate', 'extended',  
             'resolution', 'country', 'Country', 'region',
```

```
...
'addnotes', 'scite1', 'scite2', 'scite3', 'dbsource', 'INT_LOG',
'INT_IDEO', 'INT_MISC', 'INT_ANY', 'related'],
dtype='object', length=135)
```

```
In [7]: data=data[['Year', 'Month', 'Day', 'Country', 'state', 'Region', 'city', 'latitude', 'longitude', 'AttackType', 'Killed', 'Wounded', 'Target', 'Summary', 'Group', 'Target_type', 'Weapon_type', 'Motive']]
```

```
In [8]: data.isnull().sum()
```

```
Out[8]: Year          0
Month         0
Day           0
Country        0
state          421
Region          0
city            434
latitude        4556
longitude        4557
AttackType       0
Killed          10313
Wounded          16311
Target            636
Summary          66129
Group             0
Target_type       0
Weapon_type       0
Motive          131130
dtype: int64
```

```
In [9]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 181691 entries, 0 to 181690
Data columns (total 18 columns):
 #   Column      Non-Null Count  Dtype  
 ---  --          -----          ----  
 0   Year        181691 non-null   int64  
 1   Month       181691 non-null   int64  
 2   Day         181691 non-null   int64  
 3   Country     181691 non-null   object 
 4   state        181270 non-null   object 
 5   Region       181691 non-null   object 
 6   city         181257 non-null   object 
 7   latitude     177135 non-null   float64
 8   longitude    177134 non-null   float64
 9   AttackType   181691 non-null   object 
 10  Killed       171378 non-null   float64
 11  Wounded      165380 non-null   float64
 12  Target        181055 non-null   object 
 13  Summary       115562 non-null   object 
 14  Group         181691 non-null   object 
 15  Target_type   181691 non-null   object 
 16  Weapon_type   181691 non-null   object 
 17  Motive        50561 non-null   object 
dtypes: float64(4), int64(3), object(11)
memory usage: 25.0+ MB
```

```
In [10]: print('Country with the most attacks : ',data['Country'].value_counts().idxmax())
print('City with the most attacks : ',data['city'].value_counts().index[1])
print('Region with the most attacks : ',data['Region'].value_counts().idxmax())
print('Year with the most attacks : ',data['Year'].value_counts().idxmax())
print('Month with the most attacks : ',data['Month'].value_counts().idxmax())
print('Group with the most attacks : ',data['Group'].value_counts().index[1])
print('Most Attack Types : ',data['AttackType'].value_counts().idxmax())
```

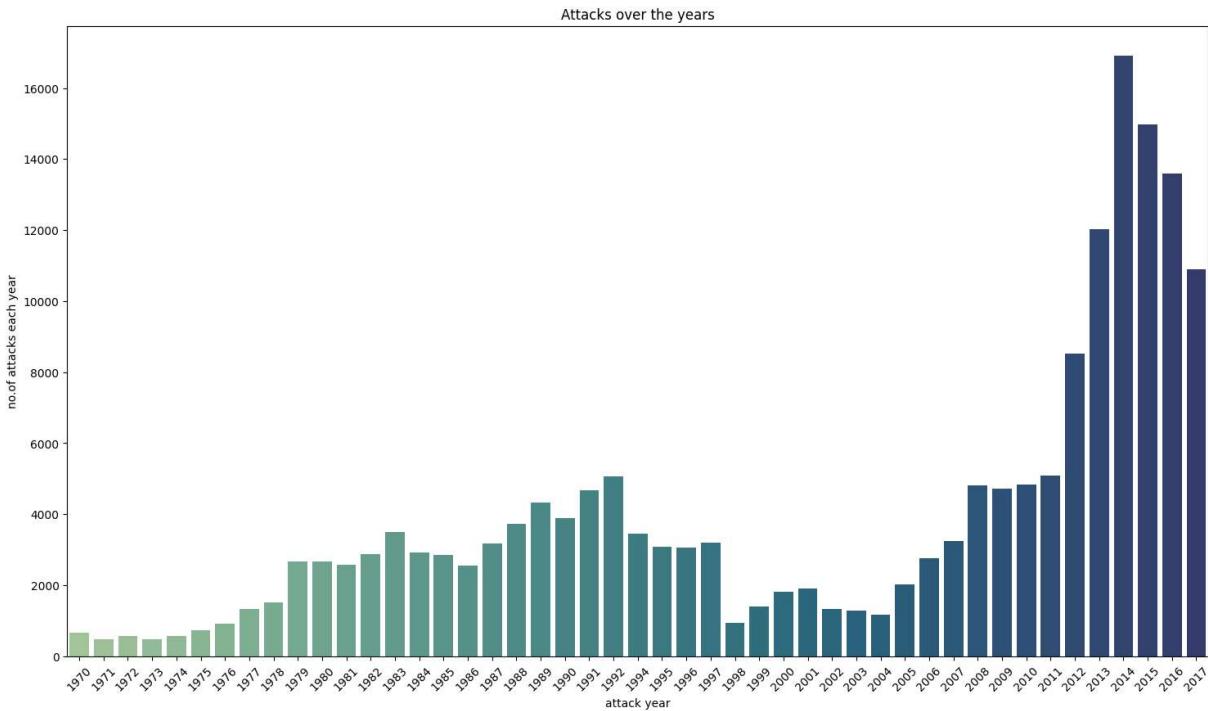
```
Country with the most attacks : Iraq
City with the most attacks : Baghdad
Region with the most attacks : Middle East & North Africa
Year with the most attacks : 2014
Month with the most attacks : 5
Group with the most attacks : Taliban
Most Attack Types : Bombing/Explosion
```

```
In [11]: data['Year'].value_counts(dropna = False).sort_index()
```

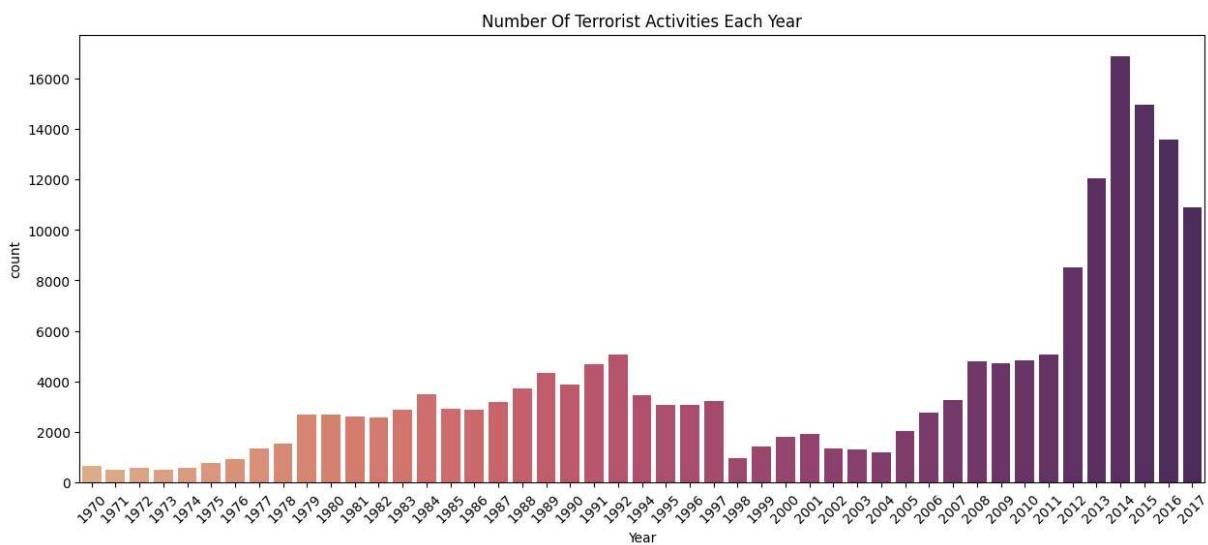
```
Out[11]: 1970      651
1971      471
1972      568
1973      473
1974      581
1975      740
1976     923
1977    1319
1978    1526
1979    2662
1980    2662
1981    2586
1982    2544
1983    2870
1984    3495
1985    2915
1986    2860
1987    3183
1988    3721
1989    4324
1990    3887
1991    4683
1992    5071
1994    3456
1995    3081
1996    3058
1997    3197
1998     934
1999    1395
2000    1814
2001    1906
2002    1333
2003    1278
2004    1166
2005    2017
2006    2758
2007    3242
2008    4805
2009    4721
2010    4826
2011    5076
2012    8522
2013   12036
2014   16903
2015   14965
2016   13587
2017   10900
Name: Year, dtype: int64
```

```
In [12]: x = data['Year'].unique()
y = data['Year'].value_counts(dropna = False).sort_index()
plt.figure(figsize = (18,10))
sns.barplot(x = x,
            y = y,
            palette = 'crest')
plt.xticks(rotation = 45)
plt.xlabel('attack year')
```

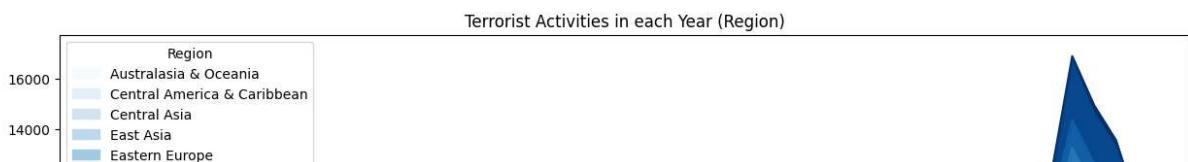
```
plt.ylabel('no.of attacks each year')
plt.title('Attacks over the years')
plt.show()
```

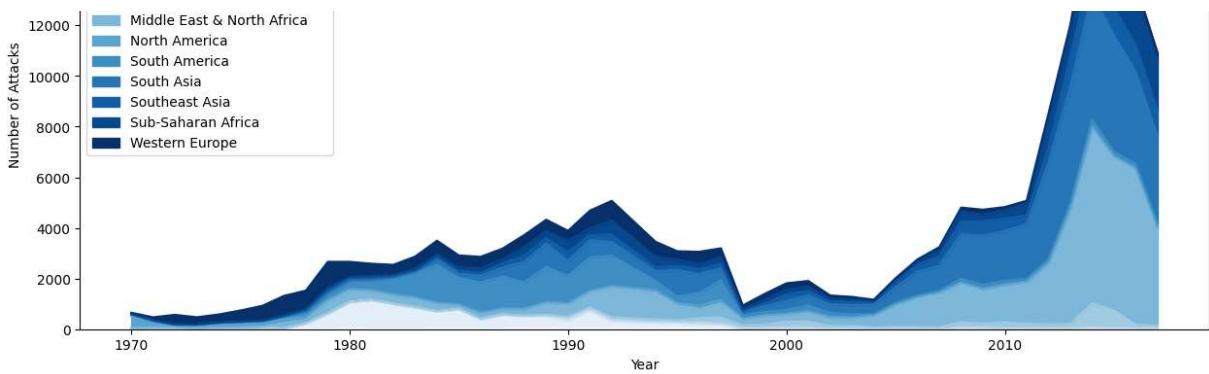


```
In [13]: plt.subplots(figsize=(15,6))
sns.countplot(x=data['Year'], palette='flare')
plt.xticks(rotation=45)
plt.title('Number Of Terrorist Activities Each Year')
plt.show()
```



```
In [14]: pd.crosstab(data.Year,data.Region).plot(kind='area',colormap='Blues',figsize=(15,6))
plt.title('Terrorist Activities in each Year (Region)')
plt.ylabel('Number of Attacks')
plt.show()
```





```
In [15]: data['Wounded'] = data['Wounded'].fillna(0).astype(int)
data['Killed'] = data['Killed'].fillna(0).astype(int)
data['casualties'] = data['Killed'] + data['Wounded']
```

```
In [16]: data1 = data.sort_values(by='casualties', ascending=False)[:40]
```

```
In [17]: heat = data1.pivot_table(index='Country', columns='Year', values='casualties')
heat.fillna(0, inplace=True)
```

```
In [18]: heat.head()
```

	Year	1982	1984	1992	1994	1995	1996	1997	1998	2001	2004	2005	2006	2007	20
Country															
Afghanistan	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Chad	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	116
Ethiopia	0.0	0.0	500.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
France	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
India	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1005.0	0.0	0.0

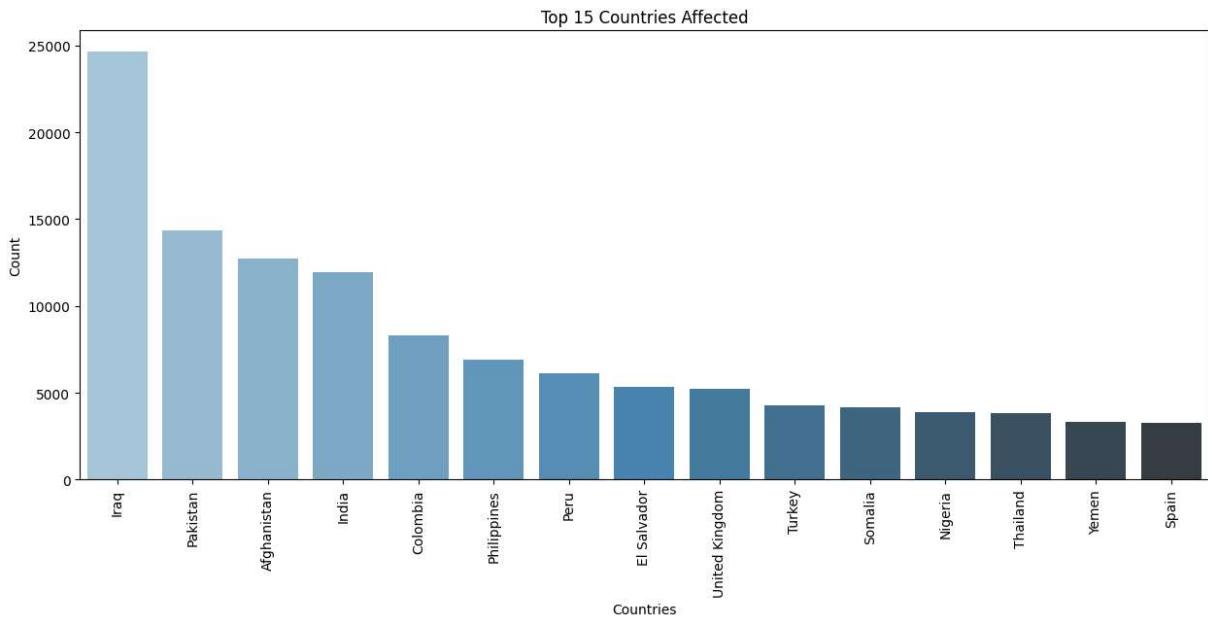
```
In [19]: import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
colorscale = [[0, '#edf8fb'], [.3, '#00BFFF'], [.6, '#8856a7'], [1, '#810f7c']]
heatmap = go.Heatmap(z=heat.values, x=heat.columns, y=heat.index, colorscale=colorscale)
d = [heatmap]
layout = go.Layout(
    title='Top 40 Worst Terror Attacks in History from 1982 to 2016',
    xaxis = dict(ticks='', nticks=20),
    yaxis = dict(ticks=''))
fig = go.Figure(data=d, layout=layout)
py.iplot(fig, filename='heatmap', show_link=False)
```

```
In [20]: data.Country.value_counts()[:15]
```

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
Name: Country, dtype: int64
```

```
In [21]: plt.subplots(figsize=(15,6))
sns.barplot(x=data['Country'].value_counts()[:15].index,y=data['Country'].value_counts()[:15].
plt.title('Top 15 Countries Affected')
plt.xlabel('Countries')
plt.ylabel('Count')
plt.xticks(rotation= 90)
plt.show()
```



```
In [22]: data.Group.value_counts()[1:15]
```

```
Taliban          7478
Islamic State of Iraq and the Levant (ISIL) 5613
Shining Path (SL) 4555
Farabundo Marti National Liberation Front (FMLN) 3351
Al-Shabaab       3288
New People's Army (NPA) 2772
Irish Republican Army (IRA) 2671
Revolutionary Armed Forces of Colombia (FARC) 2487
Boko Haram        2418
Kurdistan Workers' Party (PKK) 2310
Basque Fatherland and Freedom (ETA) 2024
Communist Party of India - Maoist (CPI-Maoist) 1878
Maoists            1630
Liberation Tigers of Tamil Eelam (LTTE) 1606
Name: Group, dtype: int64
```

```
In [23]: test = data[data.Group.isin(['Shining Path (SL)', 'Taliban', 'Islamic State of Iraq and the Leva
```

```
Tn [24]: . . . . .
```

```
test.Country.unique()
```

```
Out[24]: array(['Peru', 'Bolivia', 'Colombia', 'Argentina', 'Brazil', 'Mexico',  
   'Afghanistan', 'Pakistan', 'Syria', 'Iraq', 'Turkey', 'Tunisia',  
   'Lebanon', 'Turkmenistan', 'Israel', 'Belgium', 'Egypt', 'Libya',  
   'Saudi Arabia', 'West Bank and Gaza Strip', 'France', 'Bahrain',  
   'Jordan', 'Somalia', 'Germany', 'Yemen', 'Philippines', 'Malaysia',  
   'Indonesia', 'Russia', 'Georgia', 'United Kingdom', 'Iran',  
   'Australia'], dtype=object)
```

```
In [25]:
```

```
terror_group = data.dropna(subset=['latitude','longitude'])  
terror_group = terror_group.drop_duplicates(subset=['Country','Group'])  
terrorist_groups = data.Group.value_counts()[1:8].index.tolist()  
terror_group = terror_group.loc[terror_group.Group.isin(terrorist_groups)]  
print(terror_group.Group.unique())
```

```
["New People's Army (NPA)" 'Irish Republican Army (IRA)'  
'Shining Path (SL)' 'Farabundo Marti National Liberation Front (FMLN)'  
'Taliban' 'Al-Shabaab' 'Islamic State of Iraq and the Levant (ISIL)']
```

```
In [26]:
```

```
data.head()
```

```
Out[26]:
```

	Year	Month	Day	Country	state	Region	city	latitude	longitude	AttackT
0	1970	7	2	Dominican Republic	NaN	Central America & Caribbean	Santo Domingo	18.456792	-69.951164	Assassina
1	1970	0	0	Mexico	Federal	North America	Mexico city	19.371887	-99.086624	Hostage Ta (Kidnapp
2	1970	1	0	Philippines	Tarlac	Southeast Asia	Unknown	15.478598	120.599741	Assassina
3	1970	1	0	Greece	Attica	Western Europe	Athens	37.997490	23.762728	Bombing/Expl
4	1970	1	0	Japan	Fukouka	East Asia	Fukouka	33.580412	130.396361	Facility/Infrastruc At

```
In [27]:
```

```
killData = data.loc[:, 'Killed']  
print('Number of people killed by terror attacks : ', int(sum(killData.dropna())))
```

```
Number of people killed by terror attacks : 411868
```

```
In [28]:
```

```
attackData = data.loc[:, 'AttackType']  
typeKillData = pd.concat([attackData, killData], axis=1)
```

```
In [29]:
```

```
typeKillData.head()
```

```
Out[29]:
```

	AttackType	Killed
0	Assassination	1
1	Hostage Taking (Kidnapping)	0
2	Assassination	1
3	Bombing/Explosion	0

4 Facility/Infrastructure Attack 0

```
In [30]: typeKillFormatData = typeKillData.pivot_table(columns='AttackType', values='Killed', aggfunc='sum')
```

Out[30]:

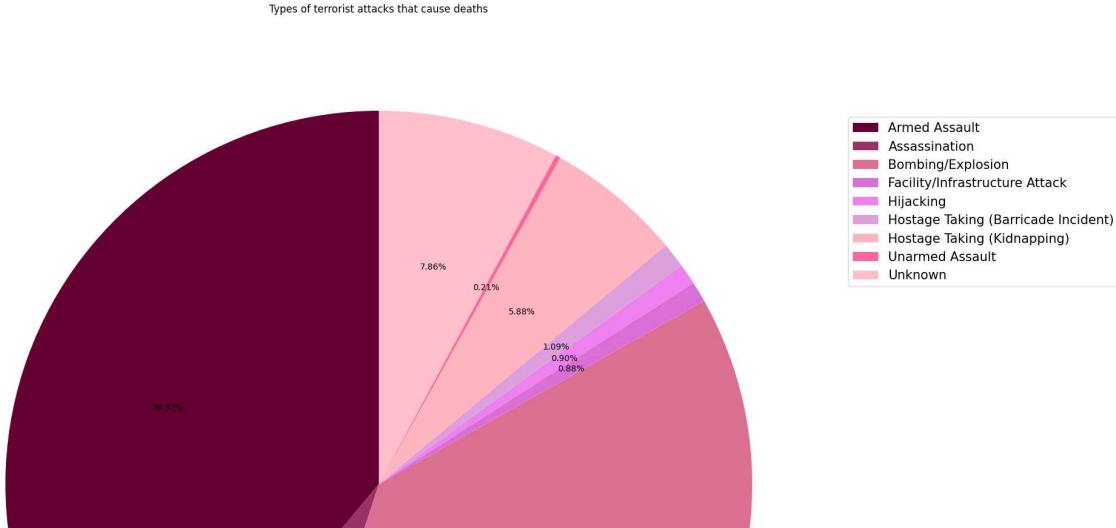
AttackType	Armed Assault	Assassination	Bombing/Explosion	Facility/Infrastructure Attack	Hijacking	Hostage Taking (Barricade Incident)	(Killed)
Killed	160297	24920	157321	3642	3718	4478	

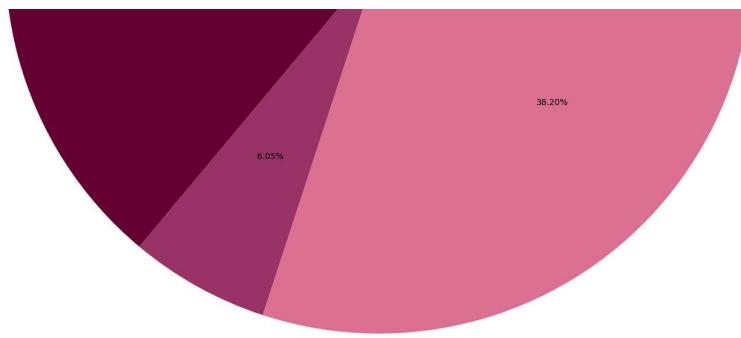
```
In [31]: typeKillFormatData.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1 entries, Killed to Killed
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Armed Assault    1 non-null      int32  
 1   Assassination    1 non-null      int32  
 2   Bombing/Explosion 1 non-null      int32  
 3   Facility/Infrastructure Attack 1 non-null      int32  
 4   Hijacking         1 non-null      int32  
 5   Hostage Taking (Barricade Incident) 1 non-null      int32  
 6   Hostage Taking (Kidnapping)        1 non-null      int32  
 7   Unarmed Assault   1 non-null      int32  
 8   Unknown           1 non-null      int32  
dtypes: int32(9)
memory usage: 152.0+ bytes
```

In [32]:

```
labels = typeKillFormatData.columns.tolist()
transpoze = typeKillFormatData.T
values = transpoze.values.tolist()
values = values[:50]
values = [int(i[0]) for i in values]
c = ['#660033','#993366','PaleVioletRed','#DA70D6','Violet','Plum','LightPink','#ff6699','Pink'
fig, ax = plt.subplots(figsize=(20, 20), subplot_kw=dict(aspect="equal"))
plt.pie(values, colors=c, 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)
plt.show()
```





```
In [33]: countryData = data.loc[:, 'Country']
countryKillData = pd.concat([countryData, killData], axis=1)
```

```
In [34]: countryKillFormatData = countryKillData.pivot_table(columns='Country', values='Killed', aggfunc
```

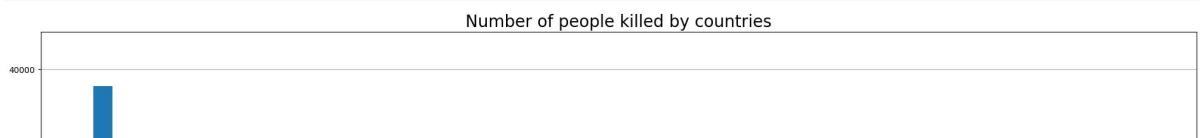
```
Out[34]:
```

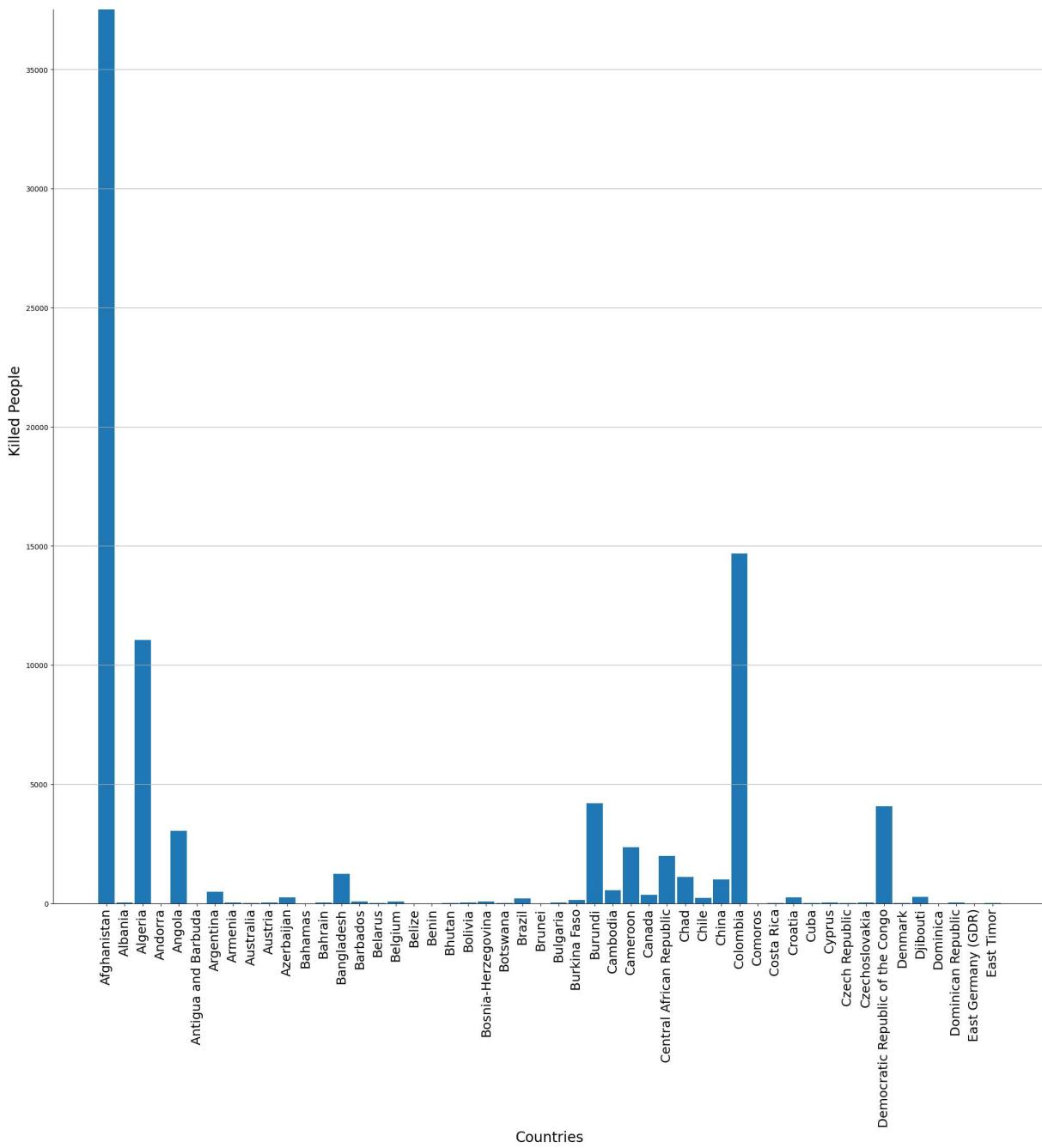
Country	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Au:
Killed	39384	42	11066	0	3043	0	490	37	23	

1 rows × 205 columns

```
In [35]: fig_size = plt.rcParams["figure.figsize"]
fig_size[0]=25
fig_size[1]=25
plt.rcParams["figure.figsize"] = fig_size
```

```
In [36]: labels = countryKillFormatData.columns.tolist()
labels = labels[:50]
index = np.arange(len(labels))
transpoze = countryKillFormatData.T
values = transpoze.values.tolist()
values = values[:50]
values = [int(i[0]) for i in values]
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, 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()
```



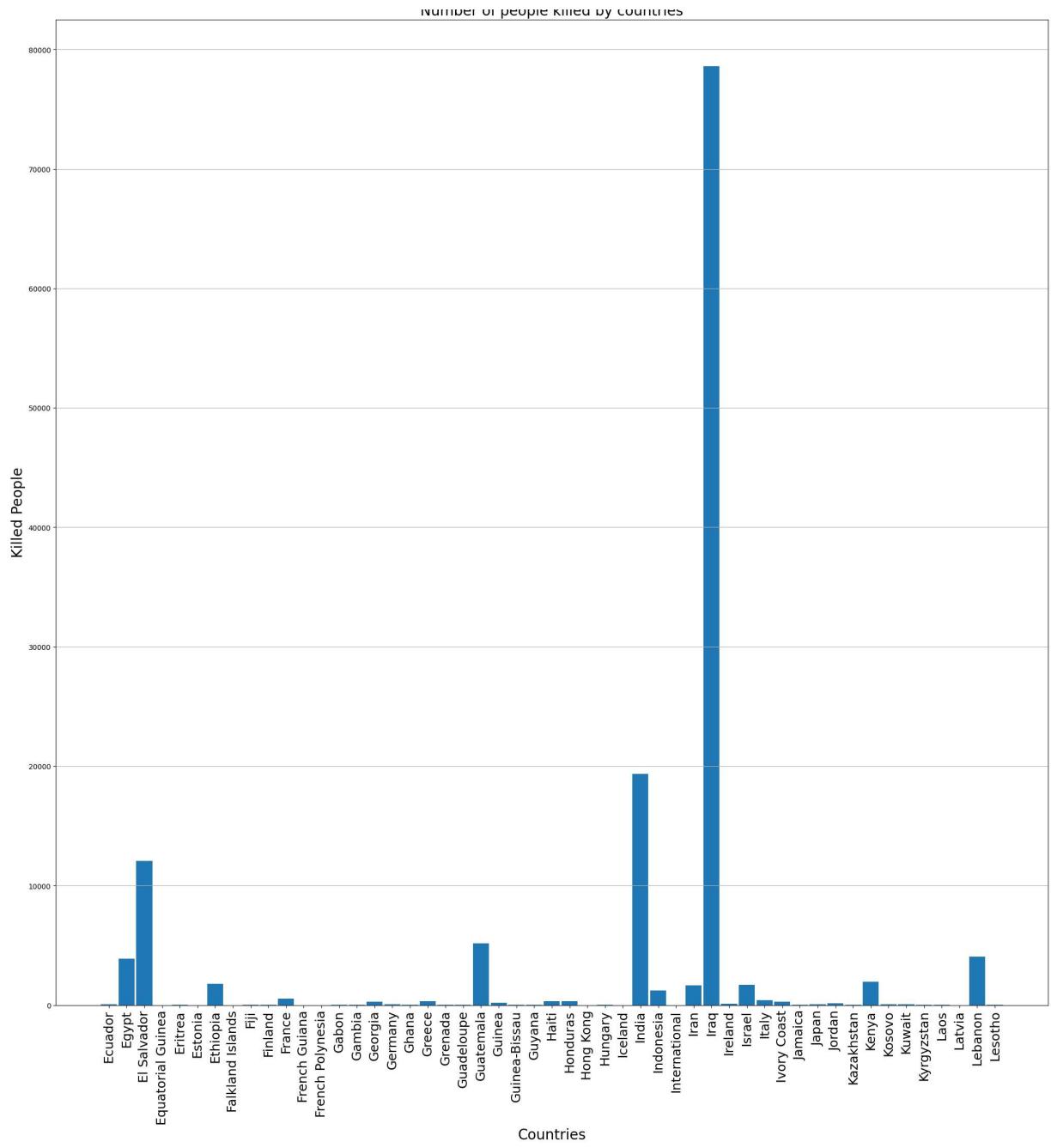


In [37]:

```

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]
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, 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()

```



In []: