In [1]:

```
import math
import warnings
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
import plotly.offline as py
import plotly.graph_objs as go
import matplotlib.pyplot as plt
warnings.filterwarnings('ignore')
```

In [5]:

```
terror = pd.read_csv('globalterrorismdb_0718dist.csv',encoding='ISO-8859-1')
```

```
terror.head(5)
```

In [6]:

```
terror.columns
```

Out[6]:

```
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)
```

In [7]:

```
terror.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)
```

In [8]:

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

In [9]:

```
terror.isnull().sum()
```

Out[9]:

```
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 [10]:

```
terror.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
```

Destructive Feature of data

In [16]:

```
print("Country with the most attacks: ",terror['Country'].value_counts().idxmax())
print("City with the most attacks: ",terror['city'].value_counts().index[1])
print("Region with the most attacks: ",terror['Region'].value_counts().idxmax())
print("Year with the most attacks: ",terror['Year'].value_counts().idxmax())
print("Month with the most attacks: ",terror['Month'].value_counts().idxmax())
print("Group with the most attacks: ",terror['Group'].value_counts().idxmax())
print("Most attack types: ",terror['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:  Unknown
Most attack types:  Bombing/Explosion
```

In [18]:

```
!pip install wordcloud
```

```
Requirement already satisfied: wordcloud in c:\users\atharva\anaconda3\lib\site-packages (1.9.2)
Requirement already satisfied: pillow in c:\users\atharva\anaconda3\lib\site-packages (from wordcloud)
(8.4.0)
Requirement already satisfied: numpy>=1.6.1 in c:\users\atharva\anaconda3\lib\site-packages (from wordcl
oud) (1.20.3)
Requirement already satisfied: matplotlib in c:\users\atharva\anaconda3\lib\site-packages (from wordclou
d) (3.4.3)
Requirement already satisfied: cycler>=0.10 in c:\users\atharva\anaconda3\lib\site-packages (from matplo
tlib->wordcloud) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\atharva\anaconda3\lib\site-packages (from m
atplotlib->wordcloud) (1.3.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\atharva\anaconda3\lib\site-packages (fro
m matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\atharva\anaconda3\lib\site-packages (from ma
tplotlib->wordcloud) (3.0.4)
Requirement already satisfied: six in c:\users\atharva\anaconda3\lib\site-packages (from cycler>=0.10->m
atplotlib->wordcloud) (1.16.0)
```

In [23]:

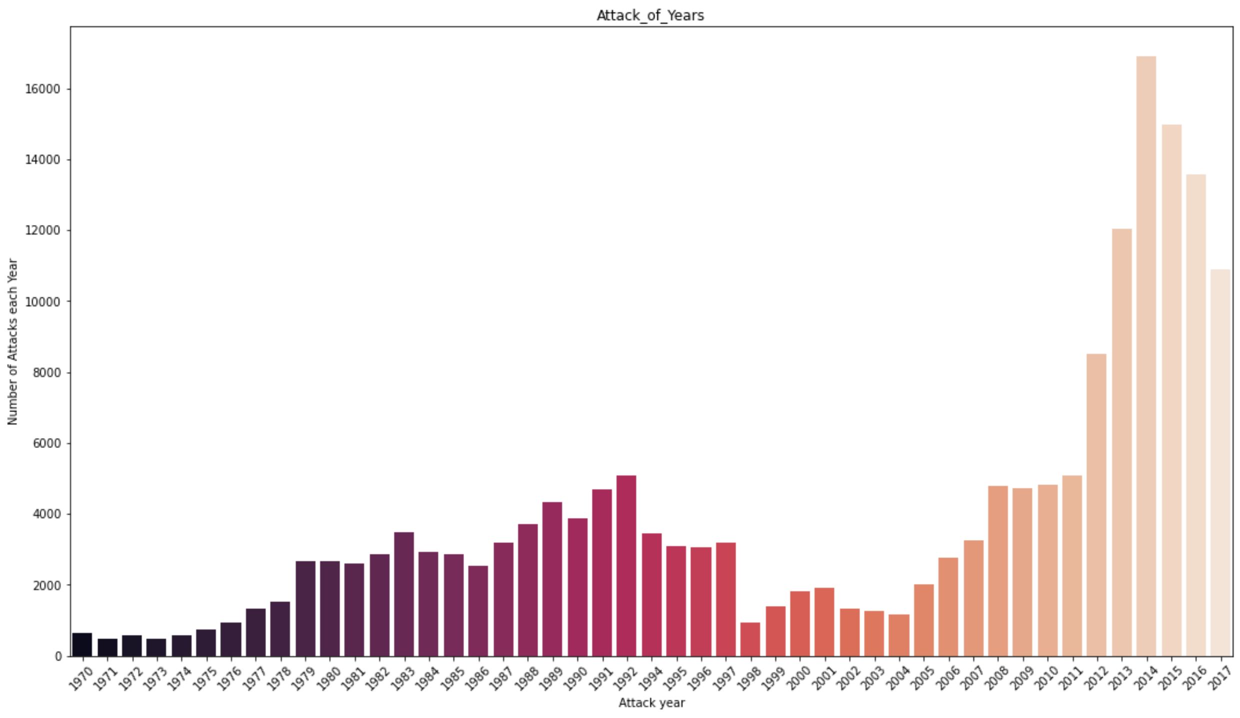
```
terror['Year'].value_counts(dropna = False).sort_index()
```

Out[23]:

```
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
```

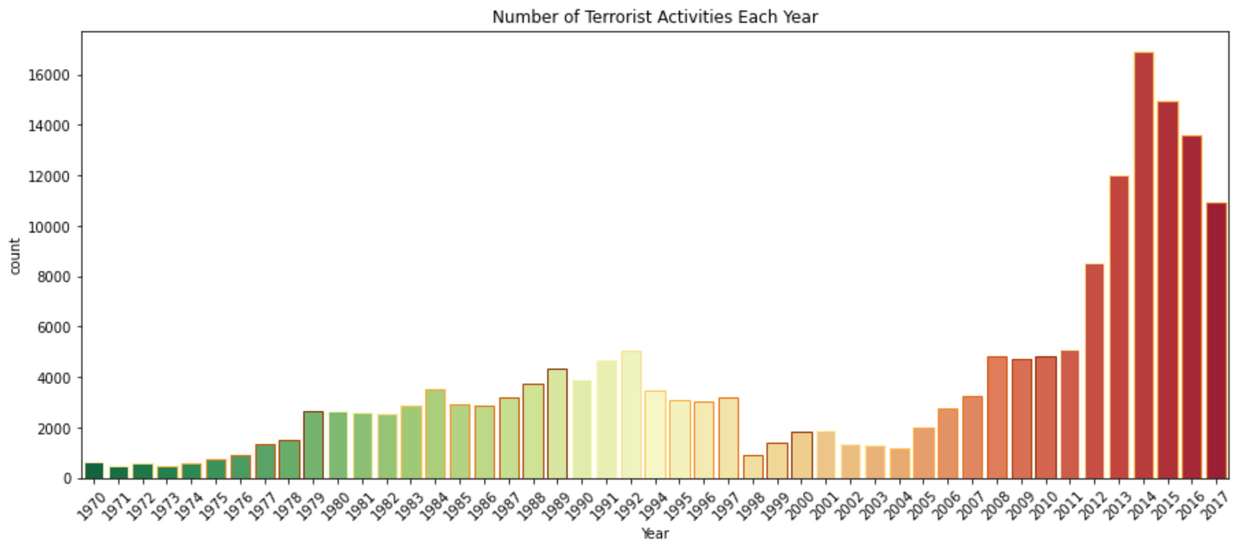
In [25]:

```python
# *********************** Data Visualization *************
# Number of Terrorist Activities each year
x_year = terror['Year'].unique()
y_count_years = terror['Year'].value_counts(dropna = False).sort_index()
plt.figure(figsize = (18,10))
sns.barplot(x = x_year,
           y = y_count_years,
           palette = 'rocket')
plt.xticks(rotation = 45)
plt.xlabel('Attack year')
plt.ylabel('Number of Attacks each Year')
plt.title('Attack_of_Years')
plt.show()
```



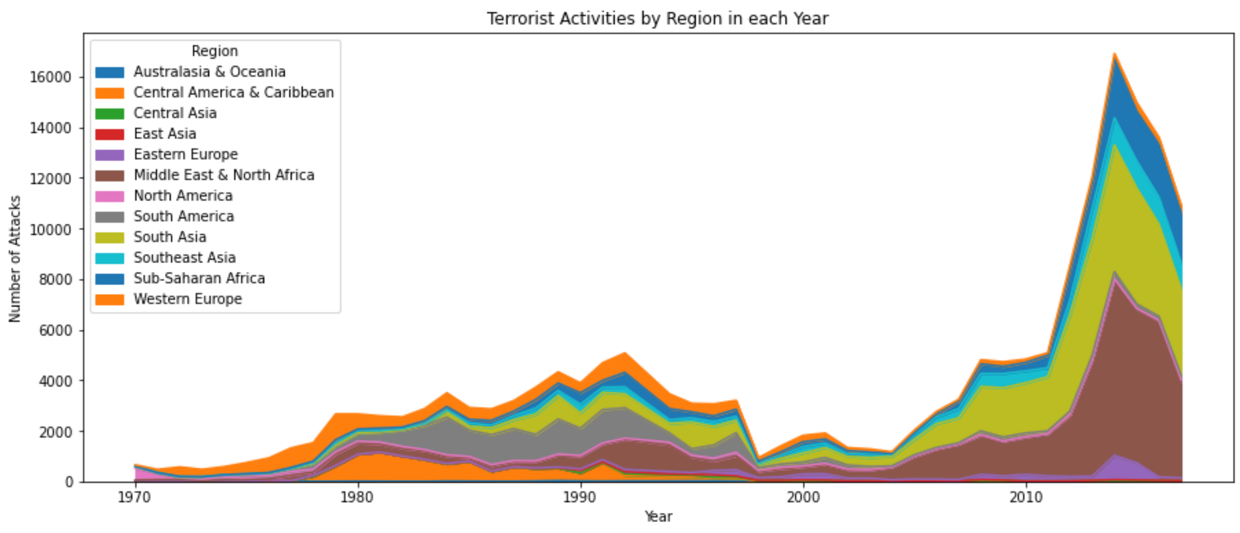
In [26]:

```python
plt.subplots(figsize = (15,6))
sns.countplot('Year',data = terror,palette = 'RdYlGn_r', edgecolor = sns.color_palette("YlOrBr", 10))
plt.xticks(rotation = 45)
plt.title('Number of Terrorist Activities Each Year')
plt.show()
```



In [27]:

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



In [31]:

```
terror['Wounded'] = terror['Wounded'].fillna(0).astype(int)
terror['Killed'] = terror['Killed'].fillna(0).astype(int)
terror['casualities'] = terror['Killed'] + terror['Wounded']
```

In [32]:

```
terror1 = terror.sort_values(by = 'casualities', ascending = False)[:40]
```

In [33]:

```
heat = terror1.pivot_table(index = 'Country', columns = 'Year', values = 'casualities')
heat.fillna(0, inplace = True)
```

In [34]:

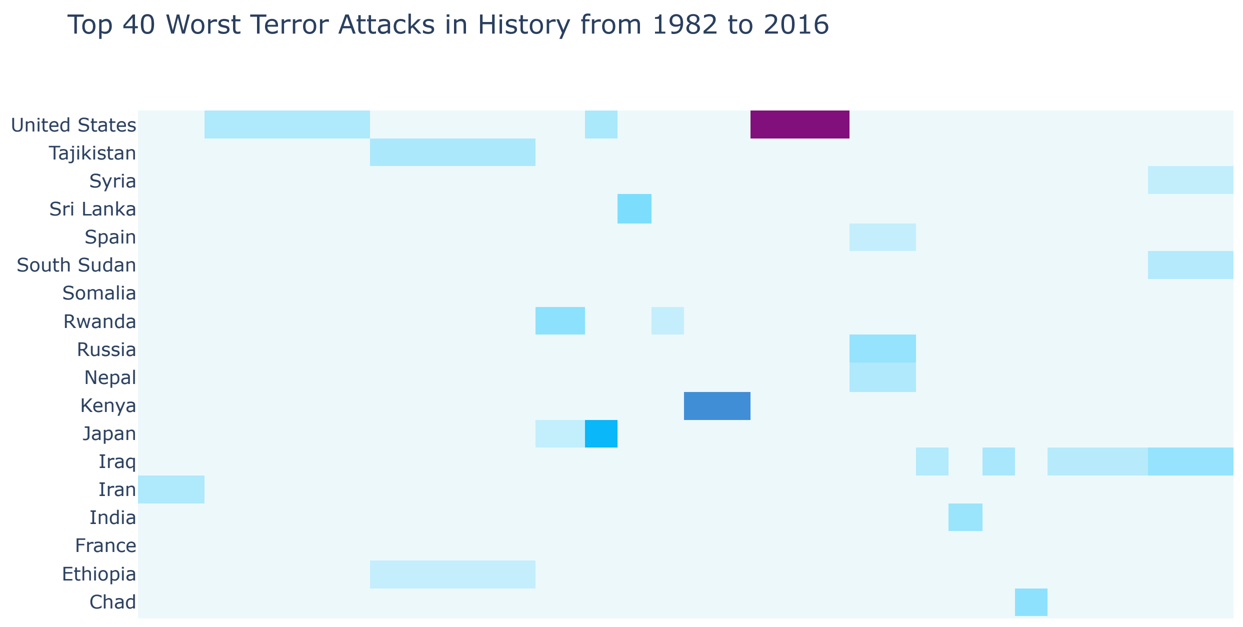
```
heat.head()
```

Out[34]:

Year	1982	1984	1992	1994	1995	1996	1997	1998	2001	2004	2005	2006	2007	2008	2009	2014	2015	2016
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	0.0	536.0	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	1161.0	0.0	0.0	0.0	0.0
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	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	0.0	0.0	520.0
India	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	0.0	0.0	0.0	0.0

In [38]:

```
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)
data = [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=data, layout=layout)
py.iplot(fig, filename='heatmap',show_link=False)
```



In [39]:

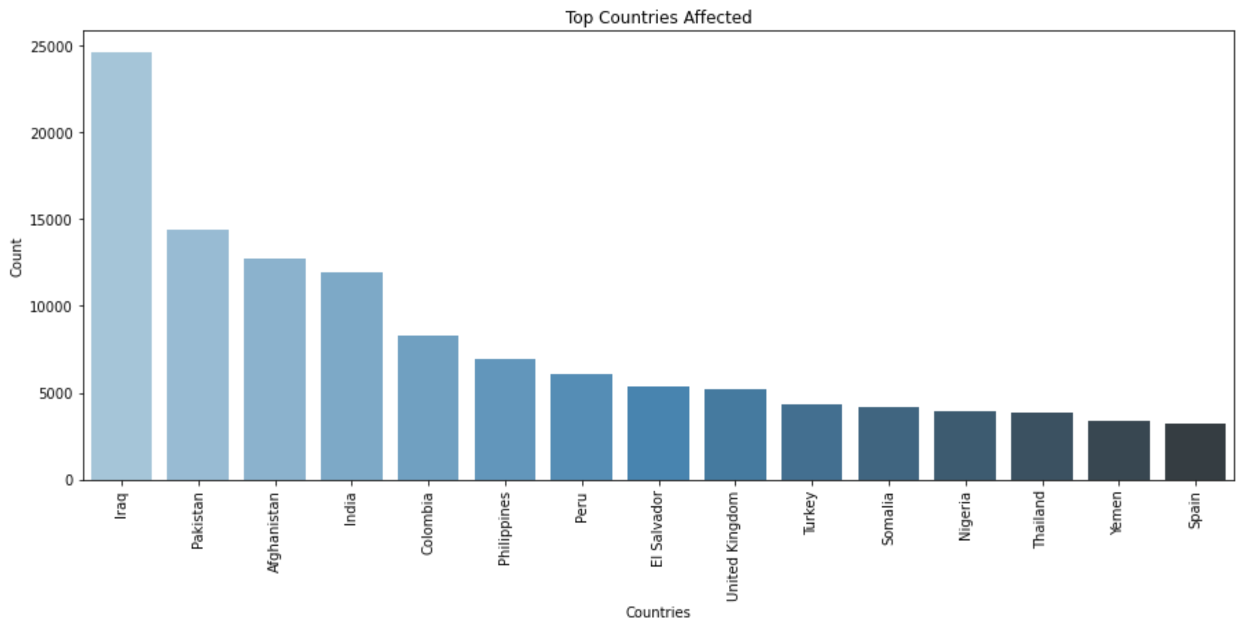
```
terror.Country.value_counts()[:15]
```

Out[39]:

```
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 [43]:

```
# Top Countries Affected by Terror Attacks
plt.subplots(figsize = (15,6))
sns.barplot(terror['Country'].value_counts()[:15].index, terror['Country'].value_counts()[:15].values, palette = 'Blue
plt.title('Top Countries Affected')
plt.xlabel('Countries')
plt.ylabel('Count')
plt.xticks(rotation = 90)
plt.show()
```



In [46]:

```
!pip install folium
```

```
Requirement already satisfied: folium in c:\users\atharva\anaconda3\lib\site-packages (0.14.0)
Requirement already satisfied: jinja2>=2.9 in c:\users\atharva\anaconda3\lib\site-packages (from folium)
(2.11.3)
Requirement already satisfied: numpy in c:\users\atharva\anaconda3\lib\site-packages (from folium) (1.2
0.3)
Requirement already satisfied: requests in c:\users\atharva\anaconda3\lib\site-packages (from folium)
(2.26.0)
Requirement already satisfied: branca>=0.6.0 in c:\users\atharva\anaconda3\lib\site-packages (from foliu
m) (0.6.0)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\atharva\anaconda3\lib\site-packages (from ji
nja2>=2.9->folium) (1.1.1)
Requirement already satisfied: idna<4,>=2.5 in c:\users\atharva\anaconda3\lib\site-packages (from reques
ts->folium) (3.2)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\atharva\anaconda3\lib\site-packages (from
requests->folium) (2022.12.7)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\atharva\anaconda3\lib\site-packages (fr
om requests->folium) (1.26.7)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\atharva\anaconda3\lib\site-packages
(from requests->folium) (2.0.4)
```

In [49]:

```
# Terror Attacks of a particular uear & their Locations
import folium
from folium.plugins import MarkerCluster
filterYear = terror['Year'] == 1970
```

In [53]:

```
filterData = terror[filterYear]
reqFilterData = filterData.loc[:,'city':'longitude']
reqFilterData = reqFilterData.dropna()
reqFilterDataList = reqFilterData.values.tolist()
```

In [55]:

```
map = folium.Map(location = [0, 30], tiles='CartoDB positron', zoom_start=2)
# clustered marker
markerCluster = folium.plugins.MarkerCluster().add_to(map)
for point in range(0, len(reqFilterDataList)):
    folium.Marker(location=[reqFilterDataList[point][1],reqFilterDataList[point][2]],
                  popup = reqFilterDataList[point][0]).add_to(markerCluster)
map
```

Out[55]:

In [56]:

```
terror.Group.value_counts()[1:15]
```

Out[56]:

```
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 [57]:

```
test = terror[terror.Group.isin(['Shining Path (SL)', 'Taliban State of Iraq and the Levant (ISIL)'])]
```

In [58]:

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

Out[58]:

```
array(['Peru', 'Bolivia', 'Colombia', 'Argentina', 'Brazil', 'Mexico'],
      dtype=object)
```

In [60]:

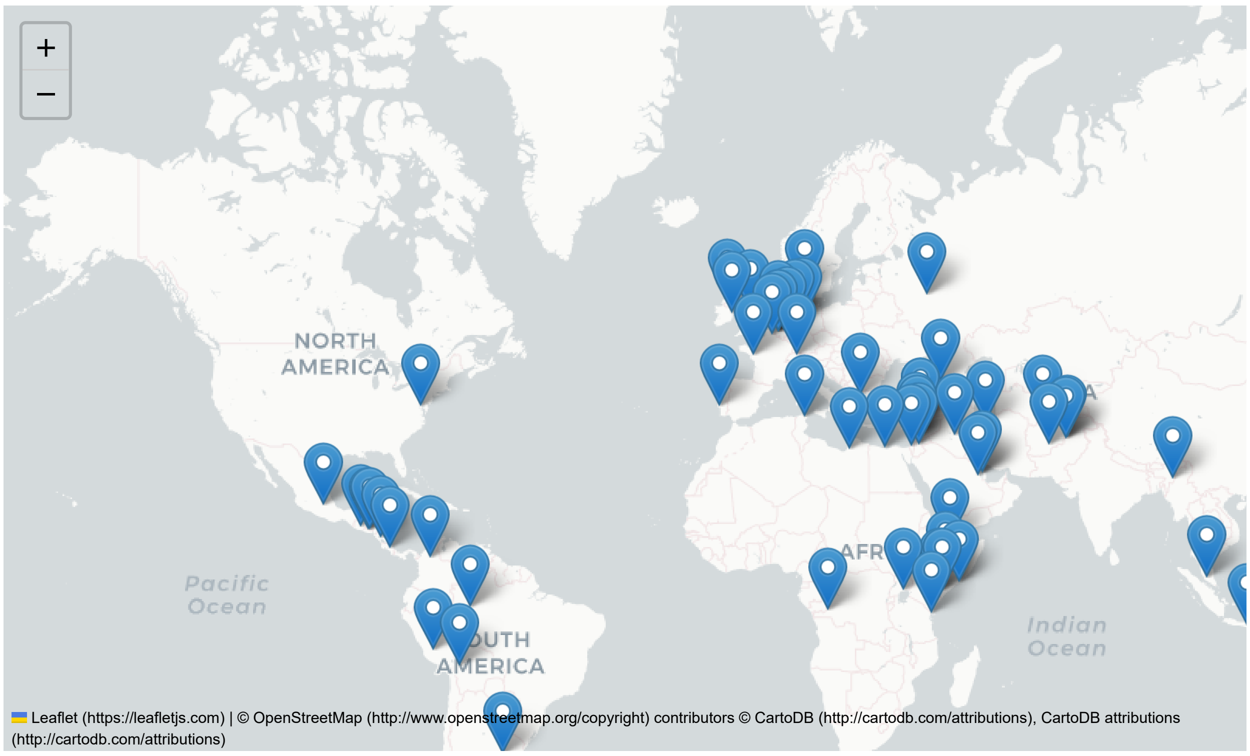
```
terror_df_group = terror.dropna(subset=['latitude','longitude'])
terror_df_group = terror_df_group.drop_duplicates(subset=['Country','Group'])
terrorist_groups = terror.Group.value_counts()[1:8].index.tolist()
terror_df_group = terror_df_group.loc[terror_df_group.Group.isin(terrorist_groups)]
print(terror_df_group.Group.unique())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

```
["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 [65]:

```
map = folium.Map(location=[20, 0], tiles="CartoDB positron", zoom_start=2)
markerCluster = folium.plugins.MarkerCluster().add_to(map)
for i in range(0,len(terror_df_group)):
    folium.Marker([terror_df_group.iloc[i]['latitude'],terror_df_group.iloc[i]['longitude']],
                  popup='Group:{}<br>Country:{}'.format(terror_df_group.iloc[i]['Group'],
                  terror_df_group.iloc[i]['Country'])).add_to(map)
map
```

Out[65]:



Leaflet (https://leafletjs.com) | © OpenStreetMap (http://www.openstreetmap.org/copyright) contributors © CartoDB (http://cartodb.com/attributions), CartoDB attributions (http://cartodb.com/attributions)

In [68]:

```
terror.head()
```

Out[68]:

	Year	Month	Day	Country	state	Region	city	latitude	longitude	AttackType	Killed	Wounded	Ta
0	1970	7	2	Dominican Republic	NaN	Central America & Caribbean	Santo Domingo	18.456792	-69.951164	Assassination	1	0	Guz
1	1970	0	0	Mexico	Federal	North America	Mexico city	19.371887	-99.086624	Hostage Taking (Kidnapping)	0	0	Na Ch dau
2	1970	1	0	Philippines	Tarlac	Southeast Asia	Unknown	15.478598	120.599741	Assassination	1	0	Empl
3	1970	1	0	Greece	Attica	Western Europe	Athens	37.997490	23.762728	Bombing/Explosion	0	0	Emb
4	1970	1	0	Japan	Fukouka	East Asia	Fukouka	33.580412	130.396361	Facility/Infrastructure Attack	0	0	Cons

In [70]:

```
# Total Number of people killed in terror attack
killData = terror.loc[:,'Killed']
print('Number of people killed by terror attack:', int(sum(killData.dropna())))
```

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

In [72]:

```
# what type of attack these deaths are made of
attackData = terror.loc[:, 'AttackType']
typeKillData = pd.concat([attackData, killData], axis = 1)
```

In [74]:

```
typeKillData.head()
```

Out[74]:

	AttackType	Killed
0	Assassination	1
1	Hostage Taking (Kidnapping)	0
2	Assassination	1
3	Bombing/Explosion	0
4	Facility/Infrastructure Attack	0

In [76]:

```
typeKillFormatData = typeKillData.pivot_table(columns = 'AttackType', values = 'Killed', aggfunc = 'sum')
typeKillFormatData
```

Out[76]:

AttackType	Armed Assault	Assassination	Bombing/Explosion	Facility/Infrastructure Attack	Hijacking	Hostage Taking (Barricade Incident)	Hostage Taking (Kidnapping)	Unarmed Assault	Unkn
Killed	160297	24920	157321	3642	3718	4478	24231	880	32

In [77]:

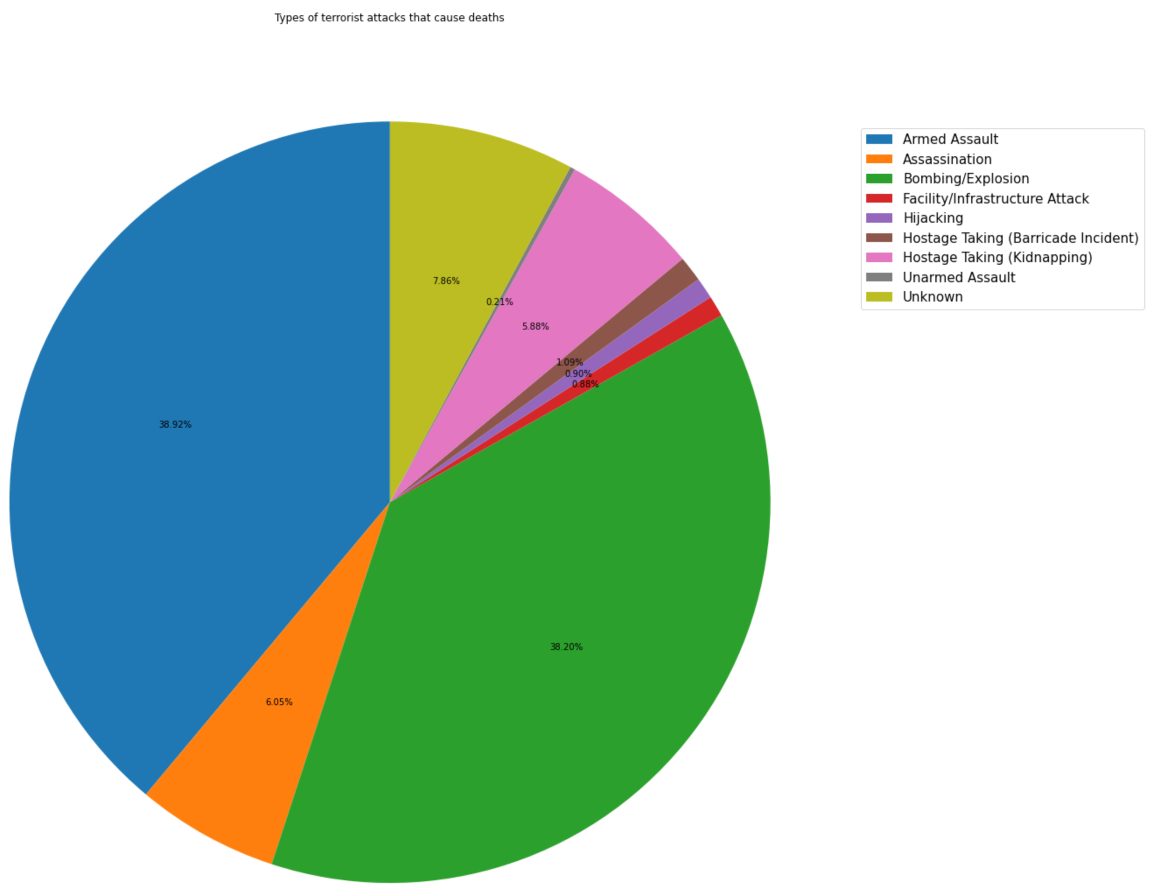
```
typeKillFormatData.info()
```

```
<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 [80]:

```
labels = typeKillFormatData.columns.tolist()  # convert line to list
transposed_values = typeKillFormatData.T.values.flatten()  # flatten the transposed values to create a 1D array

fig, ax = plt.subplots(figsize=(20, 20), subplot_kw=dict(aspect="equal"))
plt.pie(transposed_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 legend
plt.show()
```



In [82]:

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

In [83]:

```
countryKillFormatData = countryKillData.pivot_table(columns = 'Country', values = 'Killed', aggfunc = 'sum')
countryKillFormatData
```

Out[83]:

Country	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	...	Vietnam	Wallis and Futuna
Killed	39384	42	11066	0	3043	0	490	37	23	30	...	1	0

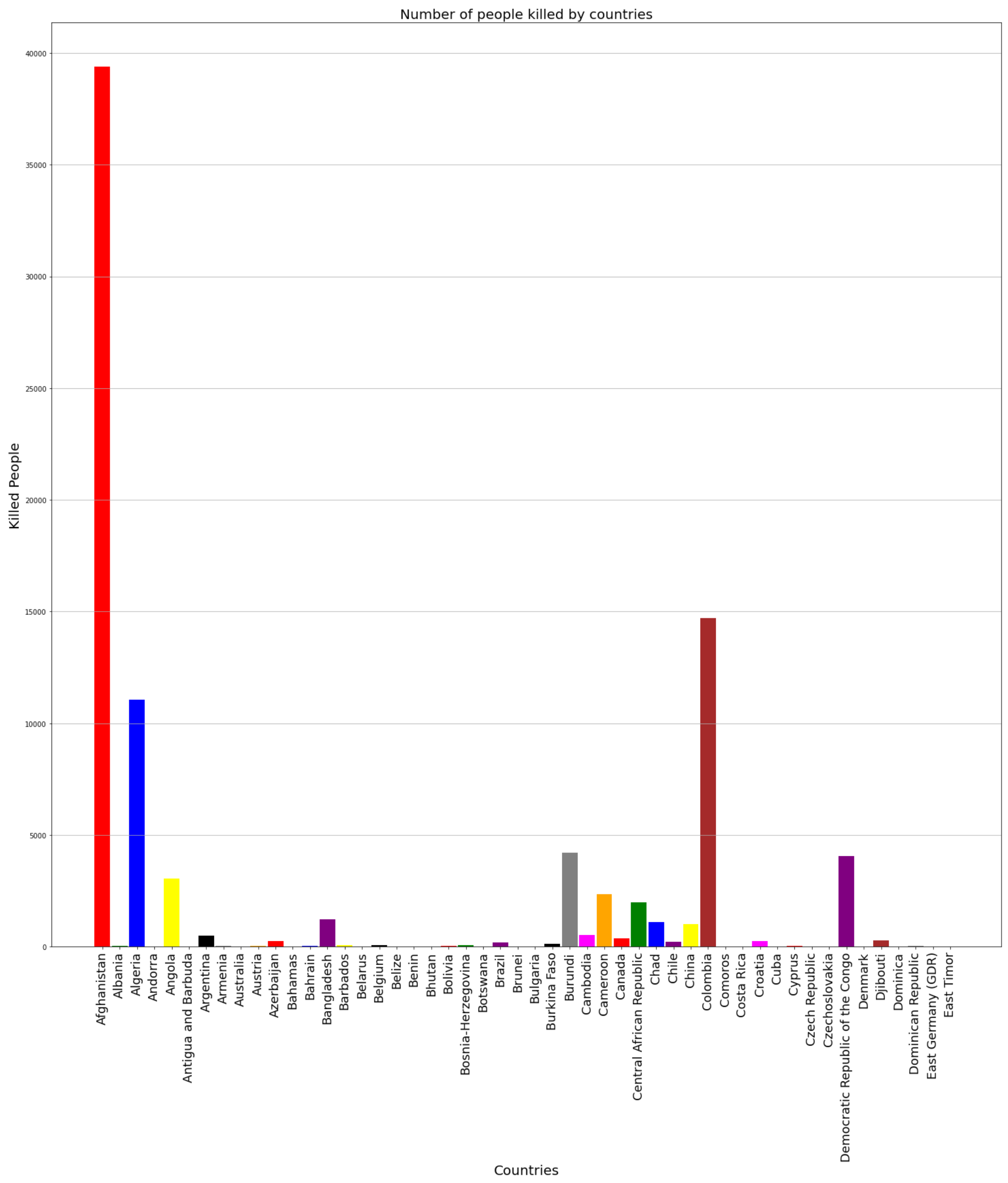
1 rows × 205 columns

In [85]:

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

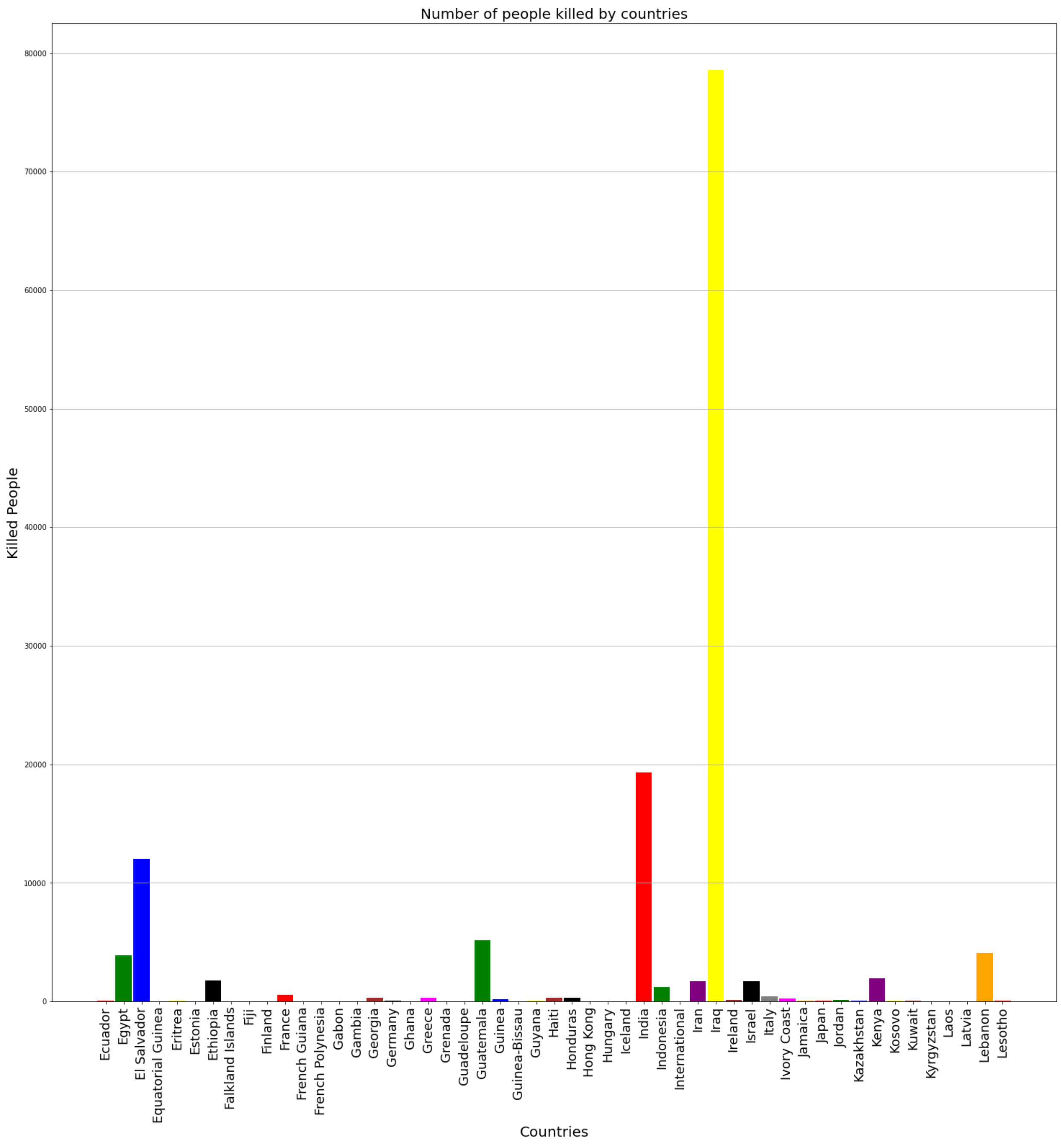
In [86]:

```
labels = countryKillFormatData.columns.tolist()
labels = labels[:50] #50 bar provides nice view
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', 'orange'] # color list for
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
Countries
0
5000
10000
15000
20000
25000
30000
35000
40000
Afghanistan
Albania
Algeria
Andorra
Angola
Antigua and Barbuda
Argentina
Armenia
Australia
Austria
Azerbaijan
Bahamas
Bahrain
Bangladesh
Barbados
Belarus
Belgium
Belize
Benin
Bhutan
Bolivia
Bosnia-Herzegovina
Botswana
Brazil
Brunei
Bulgaria
Burkina Faso
Burundi
Cambodia
Cameroon
Canada
Central African Republic
Chad
Chile
China
Colombia
Comoros
Costa Rica
Croatia
Cuba
Cyprus
Czech Republic
Czechoslovakia
Democratic Republic of the Congo
Denmark
Djibouti
Dominica
Dominican Republic
East Germany (GDR)
East Timor

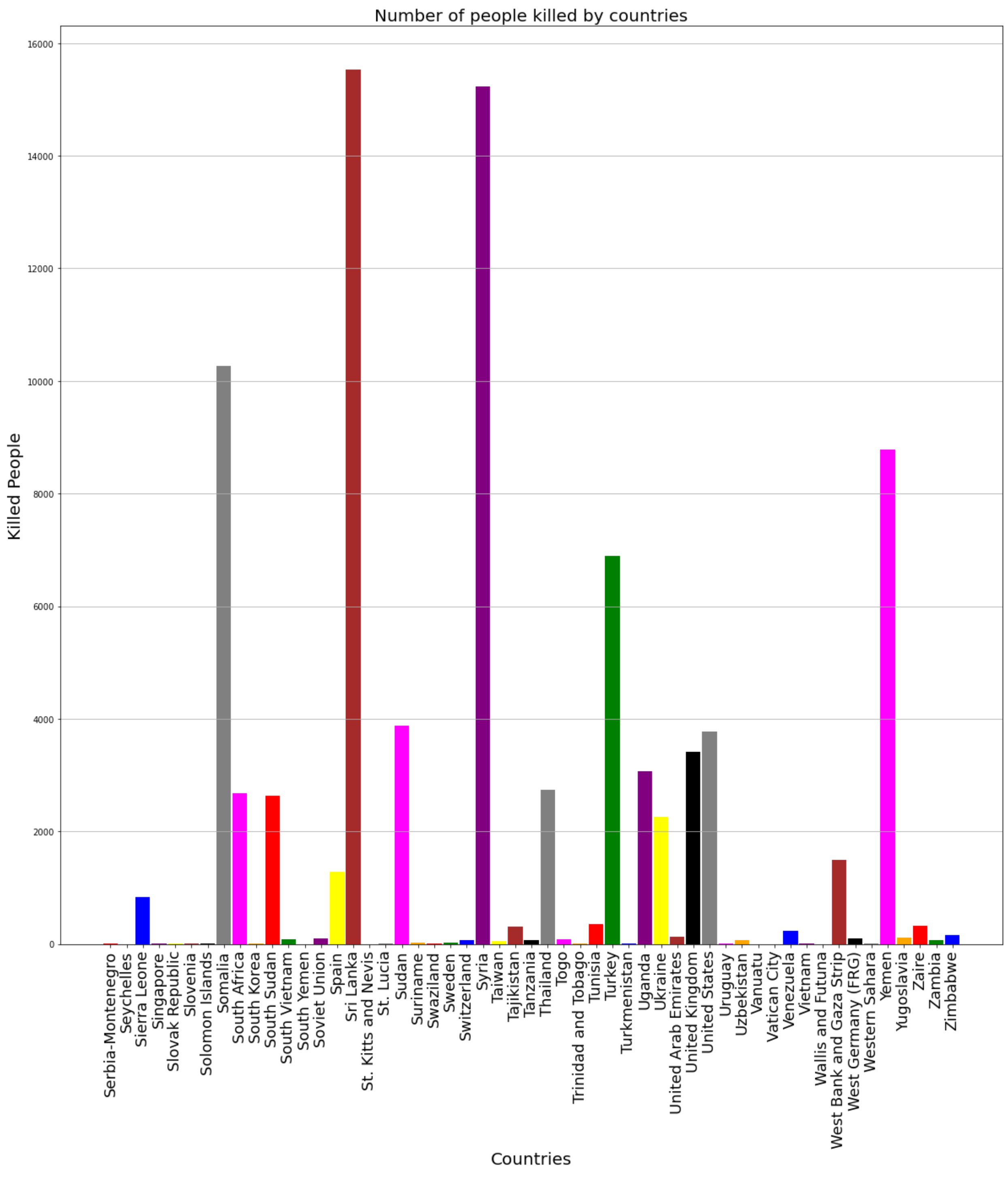
In [87]:

```
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', 'orange']
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()
```



In [88]:

```
labels = countryKillFormatData.columns.tolist()
labels = labels[152:206]
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', 'orange']
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()
```



In [ ]:

```

```