Author: Raghul V

Task: Exploratory Data Analysis (Terrorism)

Graduate Rotational Internship Program @ THE SPARKS FOUNDATION

This is an exploratory data analysis task performed with the global terrorism dataset to find out the hot zones of terrorism.

Technical Requirements

In [1]:

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

In [2]:

```
!pip install folium
```

```
Requirement already satisfied: folium in g:\anaconda\lib\site-packages (0.12.1)
Requirement already satisfied: numpy in g:\anaconda\lib\site-packages (from folium) (1.19.2)
Requirement already satisfied: branca>=0.3.0 in g:\anaconda\lib\site-packages (from folium) (0.4.2)
Requirement already satisfied: jinja2>=2.9 in g:\anaconda\lib\site-packages (from folium) (2.11.2)
Requirement already satisfied: requests in g:\anaconda\lib\site-packages (from folium) (2.24.0)
Requirement already satisfied: MarkupSafe>=0.23 in g:\anaconda\lib\site-packages (from jinja2>=2.9->folium) (1.1.1)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in g:\anaconda\lib\site-packages (from requests->folium) (1.25.11)
Requirement already satisfied: chardet<4,>=3.0.2 in g:\anaconda\lib\site-packages (from requests->folium) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in g:\anaconda\lib\site-packages (from requests->folium) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in g:\anaconda\lib\site-packages (from requests->folium) (2021.5.30)
```

In [3]:

```
import folium
from folium.plugins import MarkerCluster
```

In [4]:

```
df = pd.read_csv(r"C:\Users\Raghul\JARVIS\Prediction Using\Data\global_terrorism.csv",encoding='latin1')
```

```
G:\ANACONDA\lib\site-packages\IPython\core\interactiveshell.py:3146: DtypeWarning: Columns (4,6,31,33,61,62,63,76,79,90,92,94,96,114,115,121) 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 [5]:

```
df.head()
```

Out[5]:

	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
0	197000000001	1970	7	2	NaN	0	NaN	58	Dominican Republic	2	...	NaN	NaN	NaN	NaN	PGIS	0	0	0	0	NaN
1	197000000002	1970	0	0	NaN	0	NaN	130	Mexico	1	...	NaN	NaN	NaN	NaN	PGIS	0	1	1	1	NaN
2	197001000001	1970	1	0	NaN	0	NaN	160	Philippines	5	...	NaN	NaN	NaN	NaN	PGIS	-9	-9	1	1	NaN
3	197001000002	1970	1	0	NaN	0	NaN	78	Greece	8	...	NaN	NaN	NaN	NaN	PGIS	-9	-9	1	1	NaN
4	197001000003	1970	1	0	NaN	0	NaN	101	Japan	4	...	NaN	NaN	NaN	NaN	PGIS	-9	-9	1	1	NaN

5 rows × 135 columns

In [6]:

```
df.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
```

In [7]:

```
df.describe()
```

Out[7]:

	eventid	iyear	imonth	iday	extended	country	region	latitude	longitude	specificity	...	ransomamt	ransomamtus	ransompaid	ransompaidus	hostkidoutcome	nreleased	INT_LOG	INT_IDEO	INT_MISC	INT_ANY
count	1.816910e+05	181691.000000	181691.000000	181691.000000	181691.000000	181691.000000	181691.000000	177135.000000	1.771340e+05	181685.000000	...	1.350000e+03	5.630000e+02	7.740000e+02	552.000000	10991.000000	10400.000000	181691.000000	181691.000000	181691.000000	181691.000000
mean	2.002705e+11	2002.638997	6.467277	15.505644	0.045346	131.968501	7.160938	23.498343	-4.586957e+02	1.451452	...	3.172530e+06	5.784865e+05	7.179437e+05	240.378623	4.629242	-29.018269	-4.543731	-4.464398	0.090010	-3.945952
std	1.325957e+09	13.259430	3.388303	8.814045	0.208063	112.414535	2.933408	18.569242	2.047790e+05	0.995430	...	3.021157e+07	7.077924e+06	1.014392e+07	2940.967293	2.035360	65.720119	4.543547	4.637152	0.568457	4.691325
min	1.970000e+11	1970.000000	0.000000	0.000000	0.000000	4.000000	1.000000	-53.154613	-8.618590e+07	1.000000	...	-9.900000e+01	-9.900000e+01	-9.900000e+01	-99.000000	1.000000	-99.000000	-9.000000	-9.000000	-9.000000	-9.000000
25%	1.991021e+11	1991.000000	4.000000	8.000000	0.000000	78.000000	5.000000	11.510046	4.545640e+00	1.000000	...	0.000000e+00	0.000000e+00	-9.900000e+01	0.000000	2.000000	-99.000000	-9.000000	-9.000000	0.000000	-9.000000
50%	2.009022e+11	2009.000000	6.000000	15.000000	0.000000	98.000000	6.000000	31.467463	4.324651e+01	1.000000	...	1.500000e+04	0.000000e+00	0.000000e+00	0.000000	4.000000	0.000000	-9.000000	-9.000000	0.000000	0.000000
75%	2.014081e+11	2014.000000	9.000000	23.000000	0.000000	160.000000	10.000000	34.685087	6.871033e+01	1.000000	...	4.000000e+05	0.000000e+00	1.273412e+03	0.000000	7.000000	1.000000	0.000000	0.000000	0.000000	0.000000
max	2.017123e+11	2017.000000	12.000000	31.000000	1.000000	1004.000000	12.000000	74.633553	1.793667e+02	5.000000	...	1.000000e+09	1.320000e+08	2.750000e+08	48000.000000	7.000000	2769.000000	1.000000	1.000000	1.000000	1.000000

8 rows × 77 columns

In [8]:

```
df.corr()
```

Out[8]:

	eventid	iyear	imonth	iday	extended	country	region	latitude	longitude	specificity	...	ransomamt	ransomamtus	ransompaid	ransompaidus	hostkidoutcome	nreleased	INT_LOG	INT_IDEO	INT_MISC	INT_ANY
eventid	1.000000	0.999996	0.002706	0.018336	0.091761	-0.135039	0.401371	0.166886	0.003907	0.030641	...	-0.009990	-0.018001	-0.014094	-0.165422	0.256113	-0.181612	-0.143600	-0.133252	-0.077852	-0.175605
iyear	0.999996	1.000000	0.000139	0.018254	0.091754	-0.135023	0.401384	0.166933	0.003917	0.030626	...	-0.009984	-0.018216	-0.014238	-0.165375	0.256092	-0.181556	-0.143601	-0.133253	-0.077847	-0.175596
imonth	0.002706	0.000139	1.000000	0.005497	-0.000468	-0.006305	-0.002999	-0.015978	-0.003880	0.003621	...	-0.000710	0.046989	0.058878	-0.016597	0.011295	-0.011535	-0.002302	-0.002034	-0.002554	-0.006336
iday	0.018336	0.018254	0.005497	1.000000	-0.004700	0.003468	0.009710	0.003423	-0.002285	-0.006991	...	0.012755	-0.010502	0.003148	-0.006581	-0.006706	0.001765	-0.001540	-0.001621	-0.002027	-0.001199
extended	0.091761	0.091754	-0.000468	-0.004700	1.000000	-0.020466	0.038389	-0.024749	0.000523	0.057897	...	-0.008114	0.028177	0.001966	0.009367	0.233293	-0.192155	0.071768	0.075147	0.027335	0.080767
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
nreleased	-0.181612	-0.181556	-0.011535	0.001765	-0.192155	-0.044331	-0.149511	0.002790	-0.017745	-0.030631	...	0.054571	0.034843	0.049322	0.016832	-0.555478	1.000000	0.039388	0.040947	0.085055	0.064759
INT_LOG	-0.143600	-0.143601	-0.002302	-0.001540	0.071768	0.069904	-0.082584	-0.099827	0.002272	0.073022	...	0.035821	0.031079	0.007029	-0.045504	-0.015442	0.039388	1.000000	0.996211	0.052537	0.891051
INT_IDEO	-0.133252	-0.133253	-0.002034	-0.001621	0.075147	0.067564	-0.071917	-0.094470	0.002268	0.071333	...	0.039053	0.041983	0.013162	-0.039844	-0.016234	0.040947	0.996211	1.000000	0.082014	0.893811
INT_MISC	-0.077852	-0.077847	-0.002554	-0.002027	0.027335	0.207281	0.043139	0.097652	0.000371	-0.019197	...	0.023815	0.125162	0.037227	0.129274	-0.119776	0.085055	0.052537	0.082014	1.000000	0.252193
INT_ANY	-0.175605	-0.175596	-0.006336	-0.001199	0.080767	0.153118	-0.047900	-0.041530	0.002497	0.061389	...	0.028054	0.053484	0.007275	0.056438	-0.061946	0.064759	0.891051	0.893811	0.252193	1.000000

77 rows × 77 columns

In [9]:

```
df.columns
```

Out[9]:

```
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)
```

United States of America - Terror attacks (Death and Injuries)

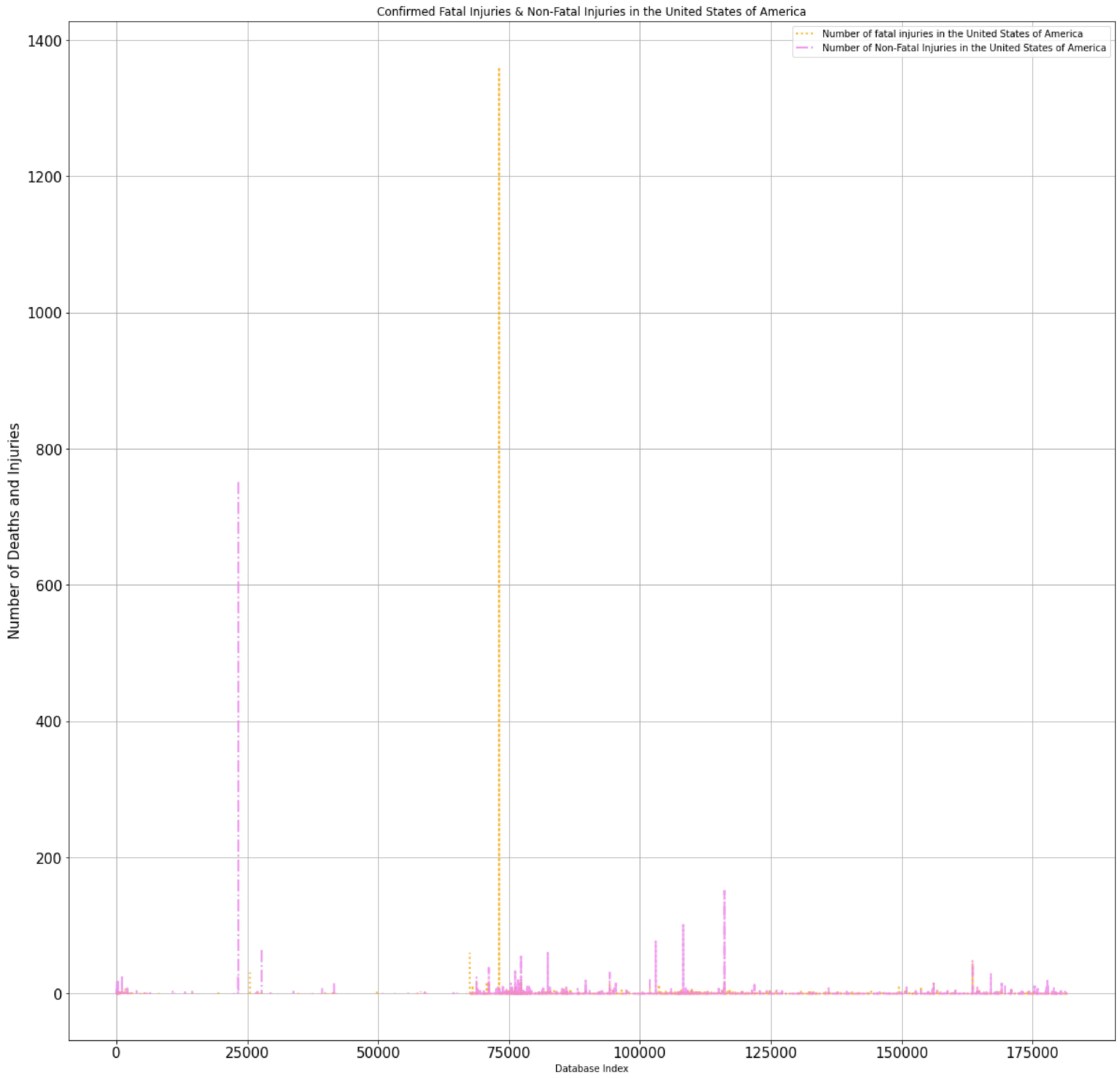
In [10]:

```
df.nkillus.plot(kind = 'line', color = 'orange', label = 'Number of fatal injuries in the United States of America', linewidth = 2, alpha = 0.8, grid = True,
                linestyle = ':', figsize = (20,20), fontsize=15)
df.nwoundus.plot(color = "violet", label = 'Number of Non-Fatal Injuries in the United States of America', linewidth = 2, alpha = 0.8, grid = True,
                linestyle = '-.', figsize = (20,20), fontsize=15)

plt.legend(loc='upper right')

plt.xlabel('Database Index', fontsize=10)
plt.ylabel('Number of Deaths and Injuries', fontsize=15)

plt.title('Confirmed Fatal Injuries & Non-Fatal Injuries in the United States of America')
plt.show()
```

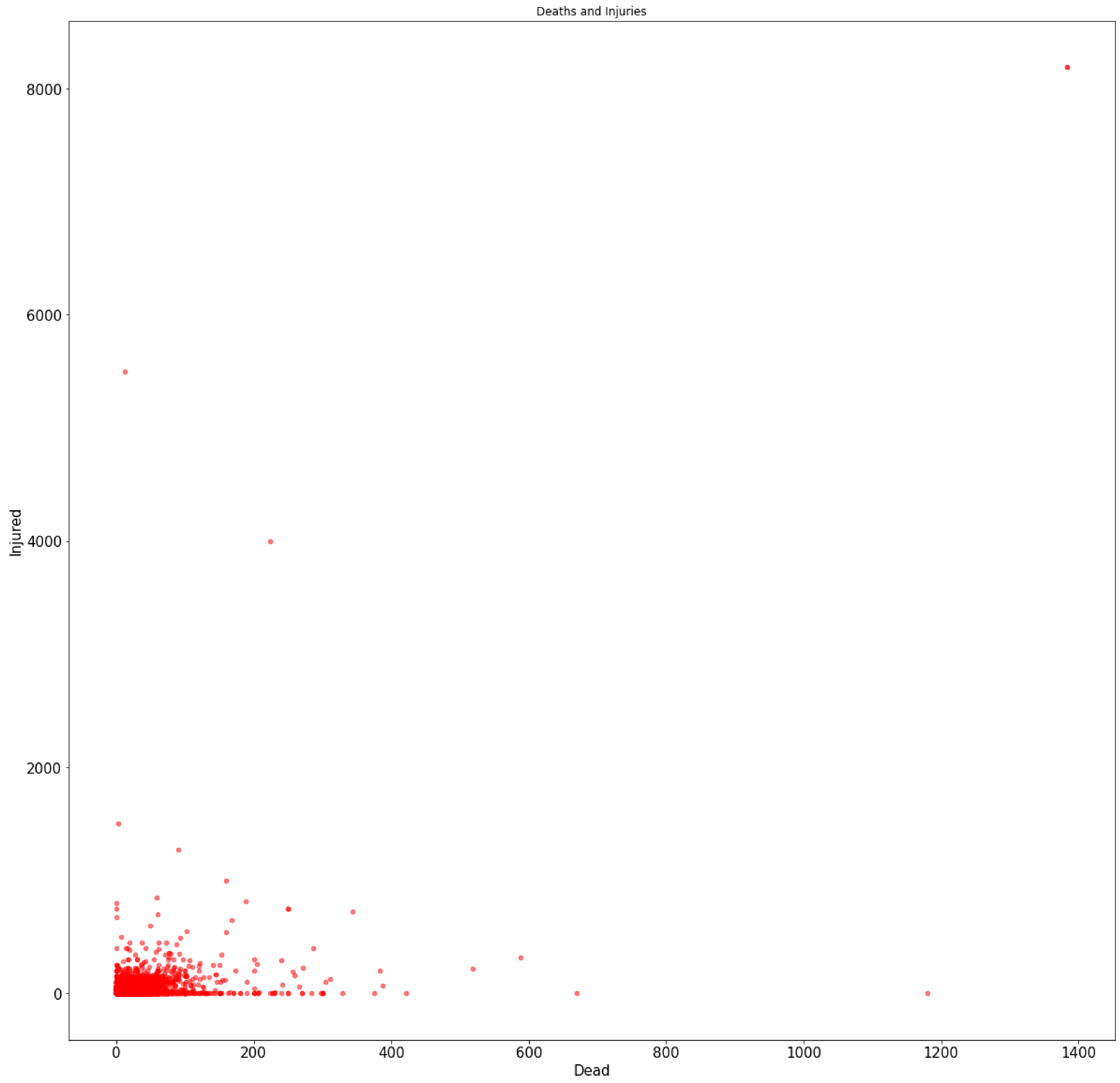


- With respect to the date given, the attack on United States citizen seems very low. But after this date range the terrorist act against United States of America.
- By finding the start date it is easy to manipulate the data to get the factors responsible for increasing terrorist attacks in the US soil.

Deaths and Injuries (in total)

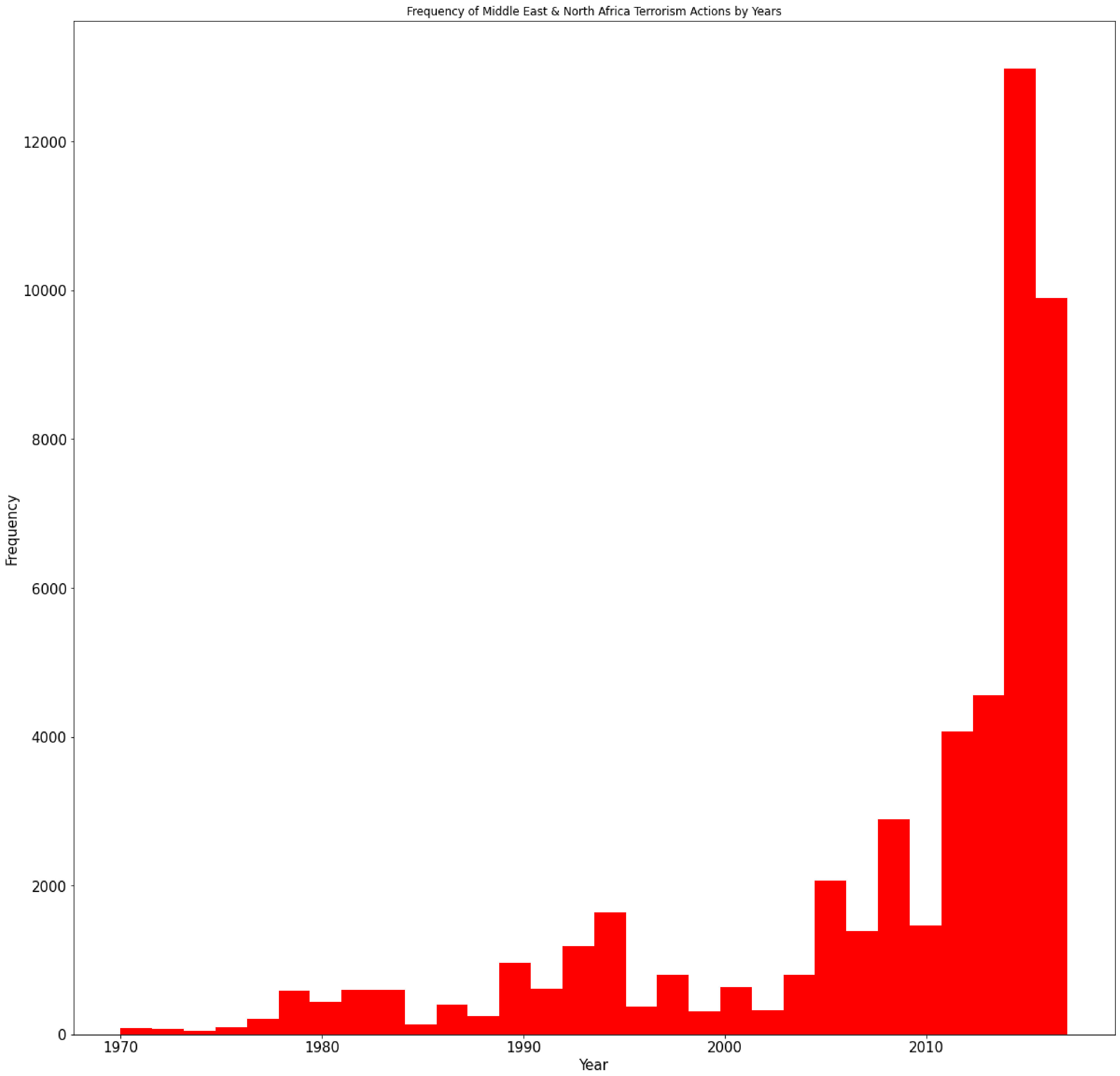
In [11]:

```
df.plot(kind = 'scatter', x = 'nkill', y = 'nwound', alpha = 0.5, color = 'red', figsize = (20,20), fontsize=15)
plt.xlabel('Dead', fontsize=15)
plt.ylabel('Injured', fontsize=15)
plt.title('Deaths and Injuries')
plt.show()
```



In [12]:

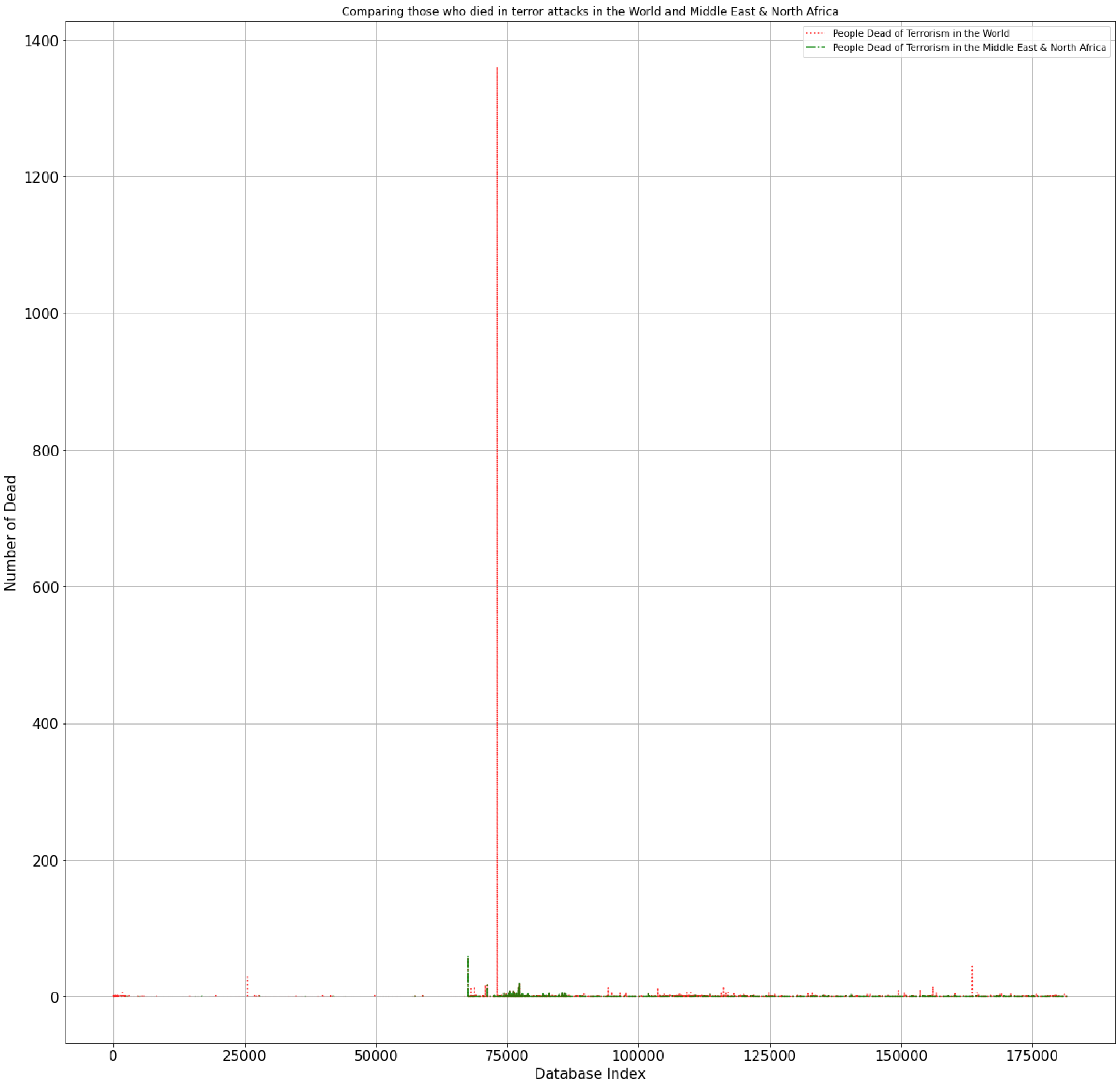
```
middleEastData = df[df['region'] == 10]
middleEastData.iyear.plot(kind = 'hist', bins = 30, figsize = (20,20), color = 'red', fontsize=15)
plt.xlabel('Year', fontsize=15)
plt.ylabel('Frequency', fontsize=15)
plt.title('Frequency of Middle East & North Africa Terrorism Actions by Years')
plt.show()
```



In [13]:

```
df.nkillus.plot(kind = 'line', color = 'red', label = 'People Dead of Terrorism in the World', linewidth = 1.5, alpha = 0.8, grid = True,
                 linestyle = ':', figsize = (20,20), fontsize=15)
middleEastData.nkillus.plot(color = "green", label = 'People Dead of Terrorism in the Middle East & North Africa', linewidth = 1.5, alpha = 0.8,
                            grid = True, linestyle = '-.', figsize = (20,20), fontsize=15)
plt.legend(loc='upper right')
plt.xlabel('Database Index', fontsize=15)
plt.ylabel('Number of Dead', fontsize=15)

plt.title('Comparing those who died in terror attacks in the World and Middle East & North Africa')
plt.show()
```



Terrorist attack on a particular year and their locations

In [14]:

```
filterYear = df['iyear'] == 1970
```

In [15]:

```
filterData = df[filterYear]

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

In [16]:

```
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[16]: Make this Notebook Trusted to load map: File -> Trust Notebook

- It is found that in 1970 the terrorist attack was around 84% in USA, Middle East and North Africa.

In [17]:
```
killData = df.loc[:,'nkill']

# drop the NaN values
print('Number of people killed during the terror attacks:', int(sum(killData.dropna())))
```

```
Number of people killed during the terror attacks: 411868
```

Types of terror attacks

In [18]:
```
attackData = df.loc[:,'attacktype1':'attacktype1_txt']
typeKillData = pd.concat([attackData, killData], axis=1)
```

In [19]:
```
typeKillFormatData = typeKillData.pivot_table(columns='attacktype1_txt', values='nkill', aggfunc='sum')
typeKillFormatData
```

Out[19]:

attacktype1_txt	Armed Assault	Assassination	Bombing/Explosion	Facility/Infrastructure Attack	Hijacking	Hostage Taking (Barricade Incident)	Hostage Taking (Kidnapping)	Unarmed Assault	Unknown
nkill	160297.0	24920.0	157321.0	3642.0	3718.0	4478.0	24231.0	880.0	32381.0

In [20]:
```
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      float64
 1   Assassination                        1 non-null      float64
 2   Bombing/Explosion                    1 non-null      float64
 3   Facility/Infrastructure Attack       1 non-null      float64
 4   Hijacking                            1 non-null      float64
 5   Hostage Taking (Barricade Incident)  1 non-null      float64
 6   Hostage Taking (Kidnapping)          1 non-null      float64
 7   Unarmed Assault                      1 non-null      float64
 8   Unknown                              1 non-null      float64
dtypes: float64(9)
memory usage: 80.0+ bytes
```

In [21]:
```
countryData = df.loc[:,'country':'country_txt']
countryKillData = pd.concat([countryData, killData], axis=1)
```

In [22]:
```
countryKillFormatData = countryKillData.pivot_table(columns='country_txt', values='nkill', aggfunc='sum')
countryKillFormatData
```

Out[22]:

country_txt	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	...	Vietnam	Wallis and Futuna	West Bank and Gaza Strip	West Germany (FRG)	Western Sahara	Yemen	Yugoslavia	Zaire	Zambia	Zimbabwe
nkill	39384.0	42.0	11066.0	0.0	3043.0	0.0	490.0	37.0	23.0	30.0	...	1.0	0.0	1500.0	97.0	1.0	8776.0	119.0	324.0	70.0	154.0

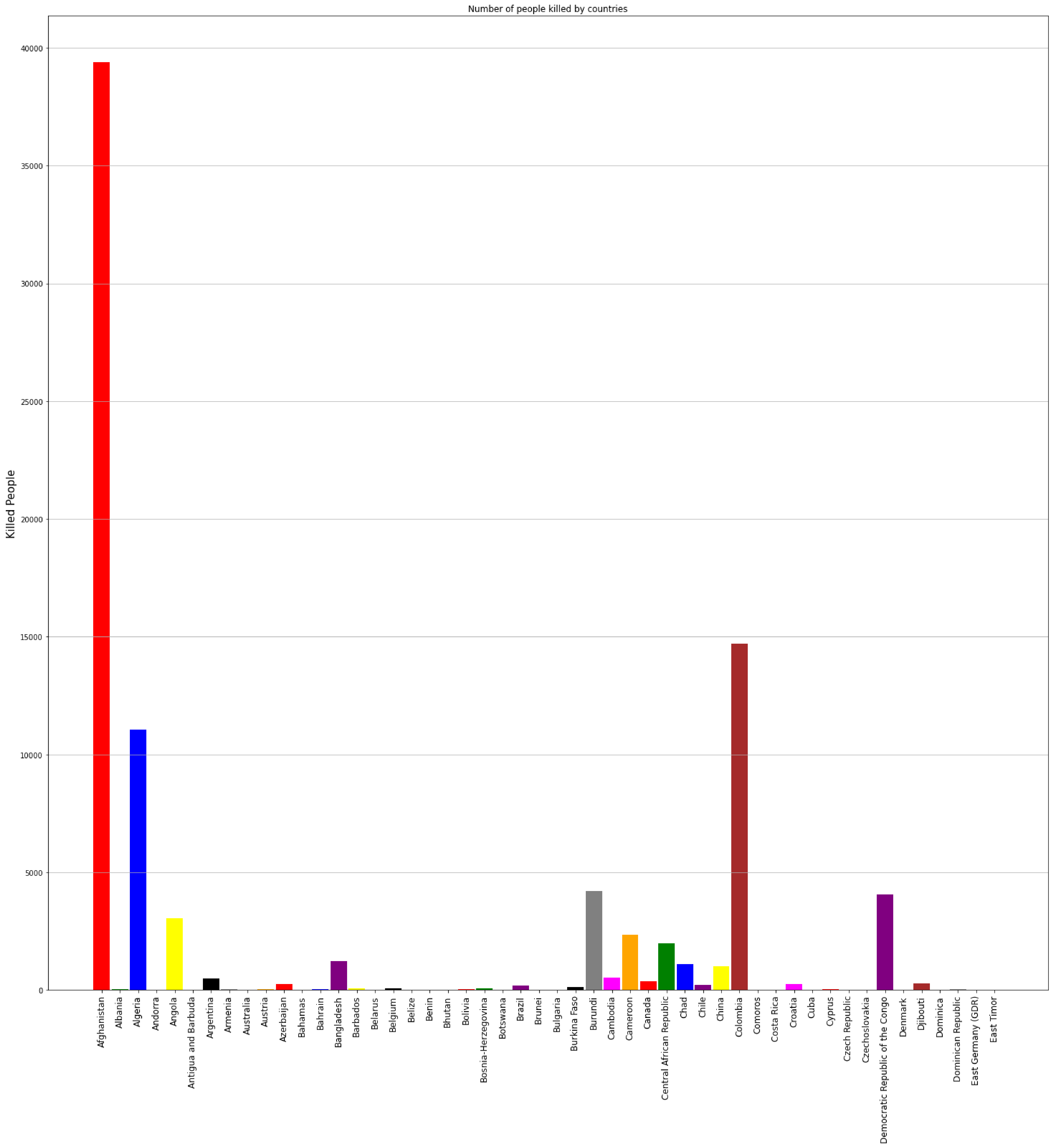
1 rows × 205 columns

In [23]:
```
countryKillFormatData.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1 entries, nkill to nkill
Columns: 205 entries, Afghanistan to Zimbabwe
dtypes: float64(205)
memory usage: 1.6+ KB
```

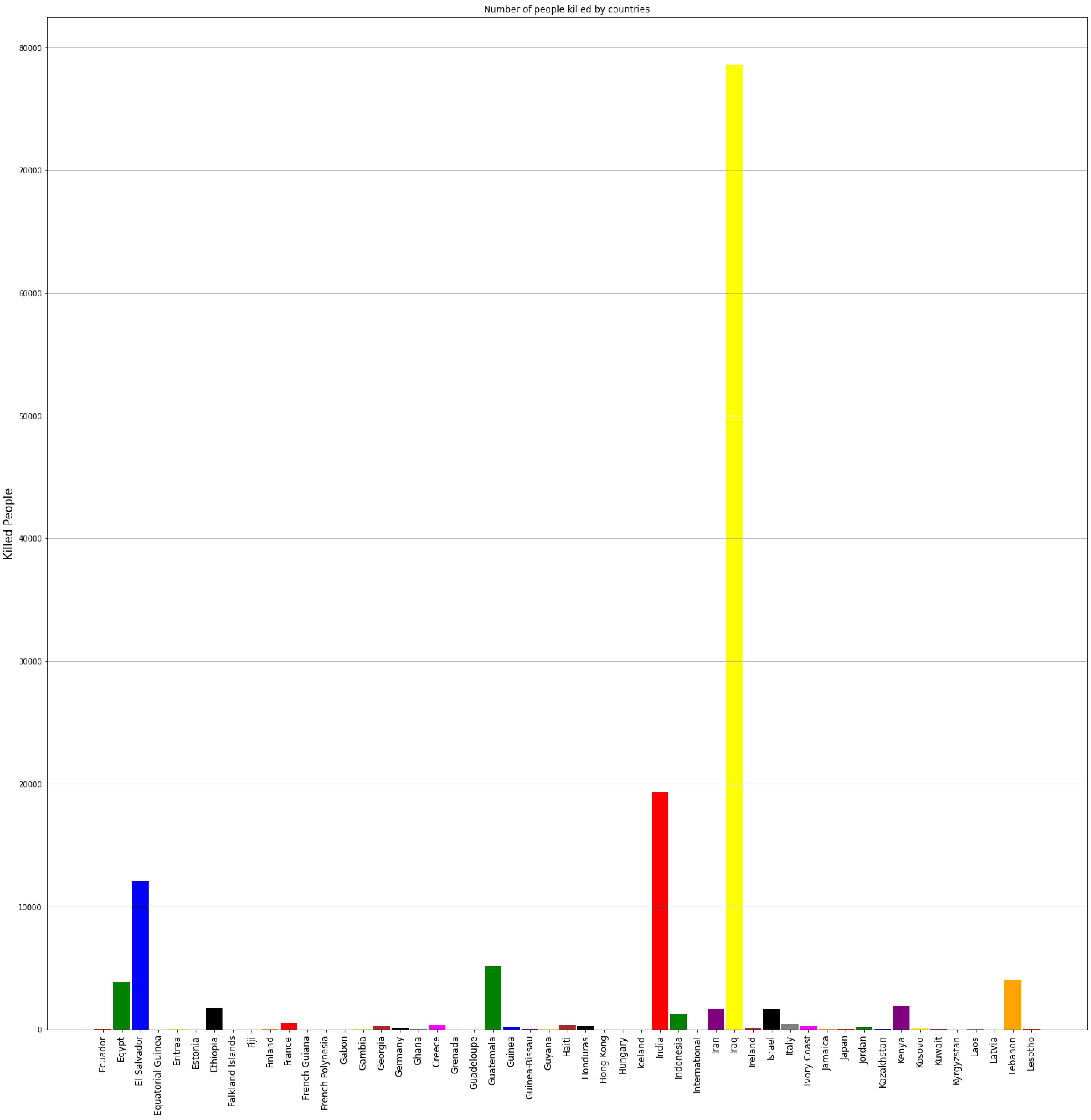
In [24]:
```
fig_size = plt.rcParams["figure.figsize"]
fig_size[0]=25
fig_size[1]=25
plt.rcParams["figure.figsize"] = fig_size
```

In [25]:
```
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]
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=15)
plt.xticks(index, labels, fontsize=12, rotation=90)
plt.title('Number of people killed by countries')
plt.show()
```



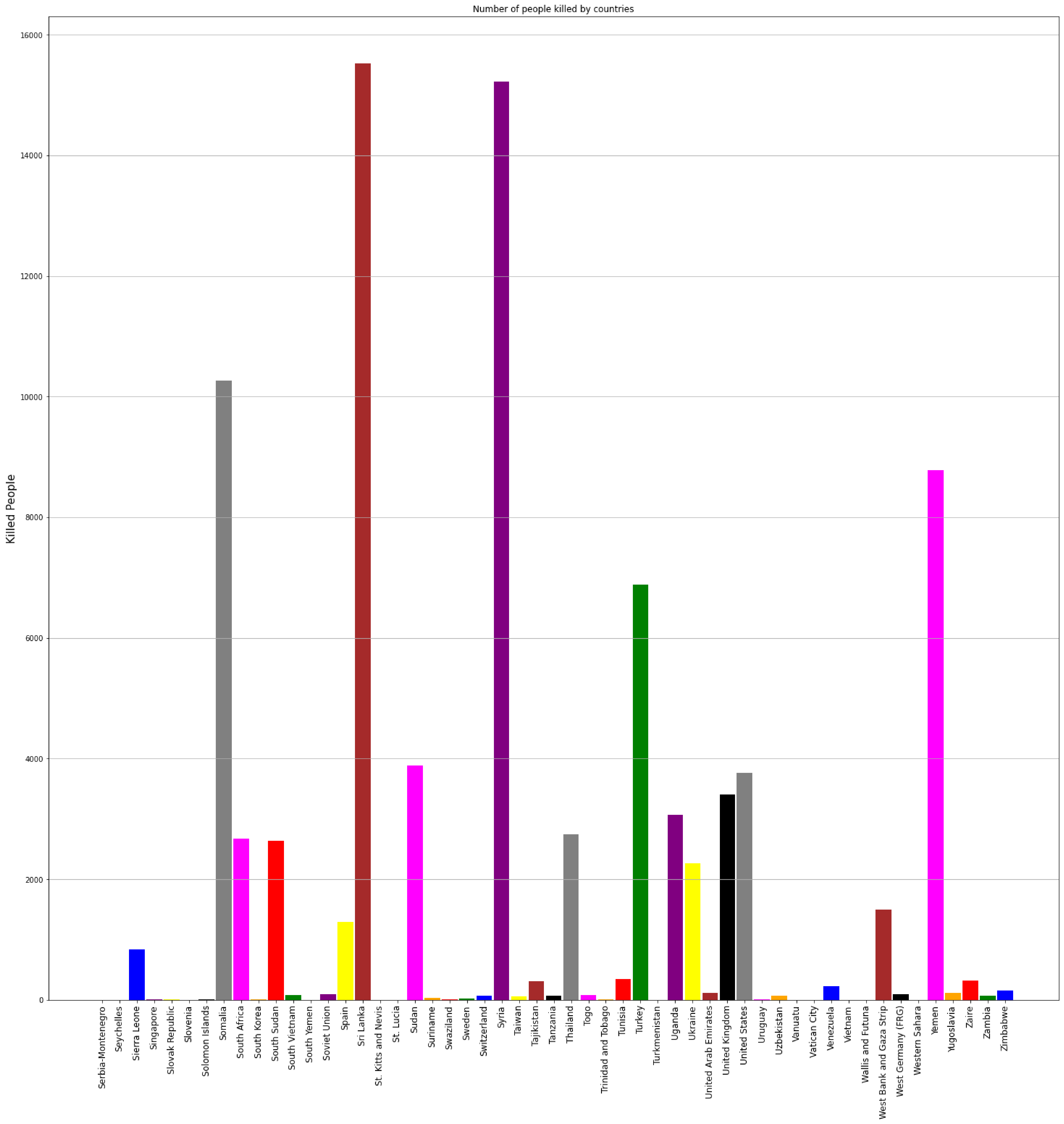
In [26]:

```
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]=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=15)
plt.xticks(index, labels, fontsize=12, rotation=90)
plt.title('Number of people killed by countries')
plt.show()
```



In [27]:

```
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=15)
plt.xticks(index, labels, fontsize=12, rotation=90)
plt.title('Number of people killed by countries')
plt.show()
```



Conclusion

- The exploratory data analysis task was carried out successfully to analyze and visualize the hot zones of terrorism.
    - We can conclude that the middle east and north africa are the hot zones which has faced a serious terror attacks upto date.
    - Also, a common perception of people on muslim community is been broken. (i.e,) When we interpret the visualization from the above study, it seems that Iraq, Iran, Afghanistan and Pakistan are the highly damaged countries due to terror attacks and those are the countries with muslim majority.

In [ ]: