EDA on Global Terrorism Dataset

Project by Ashmita Bhattacharjee

mimibhattaworkpurpose0408@gmail.com

In [1]:

```
import math
import warnings
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from dython import nominal
warnings.filterwarnings('ignore')
```

In [2]:

```
df = pd.read_csv('globalterrorismdb_0718dist.csv')
df.head()
```

Out[2]:

	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	country_txt	regi
0	197000000001	1970	7	2	NaN	0	NaN	58	Dominican Republic	
1	197000000002	1970	0	0	NaN	0	NaN	130	Mexico	
2	197001000001	1970	1	0	NaN	0	NaN	160	Philippines	
3	197001000002	1970	1	0	NaN	0	NaN	78	Greece	
4	197001000003	1970	1	0	NaN	0	NaN	101	Japan	

5 rows × 135 columns

In [3]:

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

Out[3]:

```
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 [4]:

```
df.shape
```

Out[4]:

```
(181691, 135)
```

In [5]:

```
df.columns
```

Out[5]:

```
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 [18]:

```
df.describe()
```

Out[18]:

	eventid	iyear	imonth	iday	extended	country	
count	1.816910e+05	181691.000000	181691.000000	181691.000000	181691.000000	181691.000000	1
mean	2.002705e+11	2002.638997	6.467277	15.505644	0.045346	131.968501	
std	1.325957e+09	13.259430	3.388303	8.814045	0.208063	112.414535	
min	1.970000e+11	1970.000000	0.000000	0.000000	0.000000	4.000000	
25%	1.991021e+11	1991.000000	4.000000	8.000000	0.000000	78.000000	
50%	2.009022e+11	2009.000000	6.000000	15.000000	0.000000	98.000000	
75%	2.014081e+11	2014.000000	9.000000	23.000000	0.000000	160.000000	
max	2.017123e+11	2017.000000	12.000000	31.000000	1.000000	1004.000000	

8 rows × 77 columns

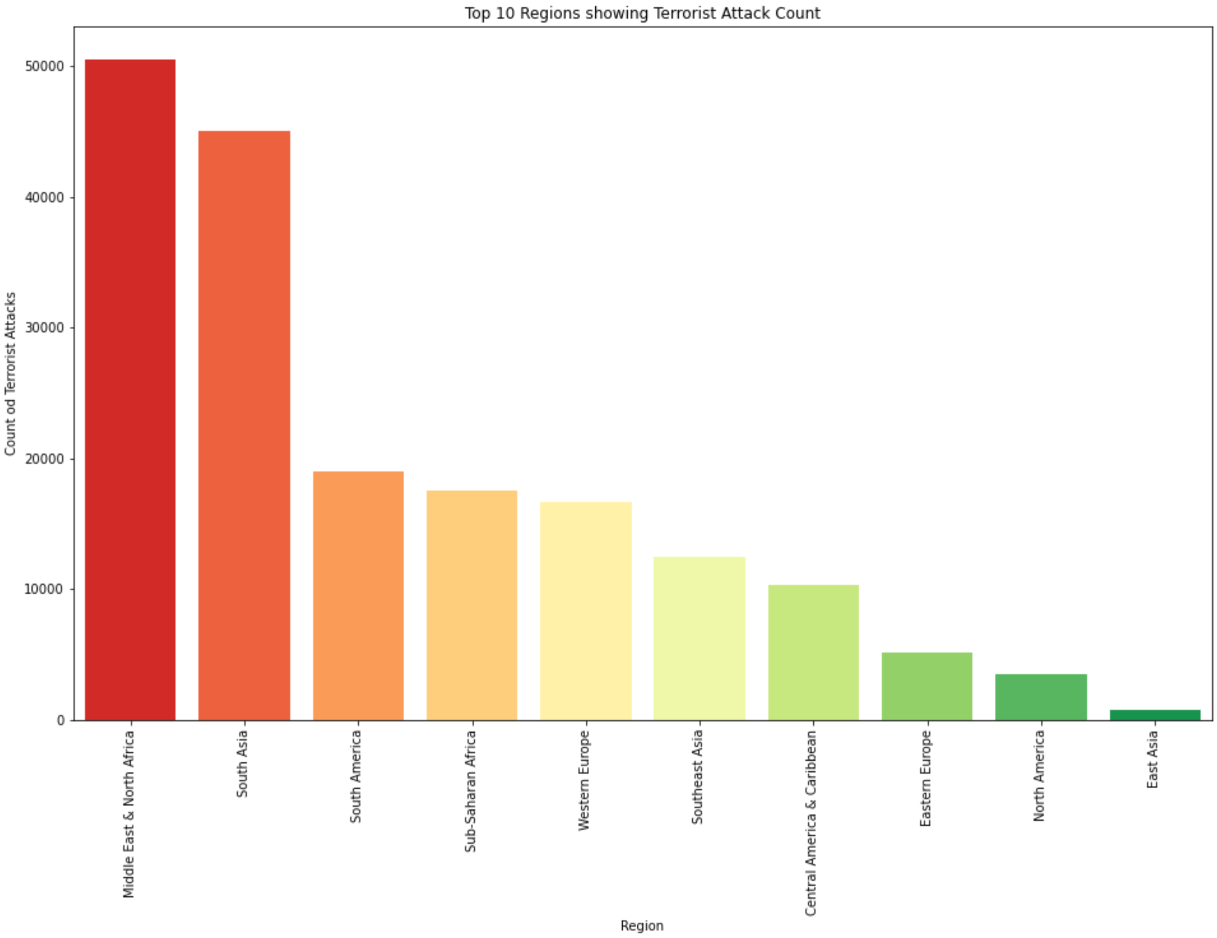
In [37]:

```
print("Region with the most attacks:",df['region_txt'].value_counts().idxmax())
print("Country with the most attacks:",df['country_txt'].value_counts().idxmax())
print("City with the most attacks:",df['city'].value_counts().index[1])
print("Year with the most attacks:",df['iyear'].value_counts().idxmax())
print("Group suffering most attacks:",df['targtype1_txt'].value_counts().index[0])
print("Month with the most attacks:",df['imonth'].value_counts().idxmax())
print("Terrorist Group with the most attacks:",df['gname'].value_counts().index[1])
print("Most Attack Types:",df['attacktype1_txt'].value_counts().idxmax())
```

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

In [39]:

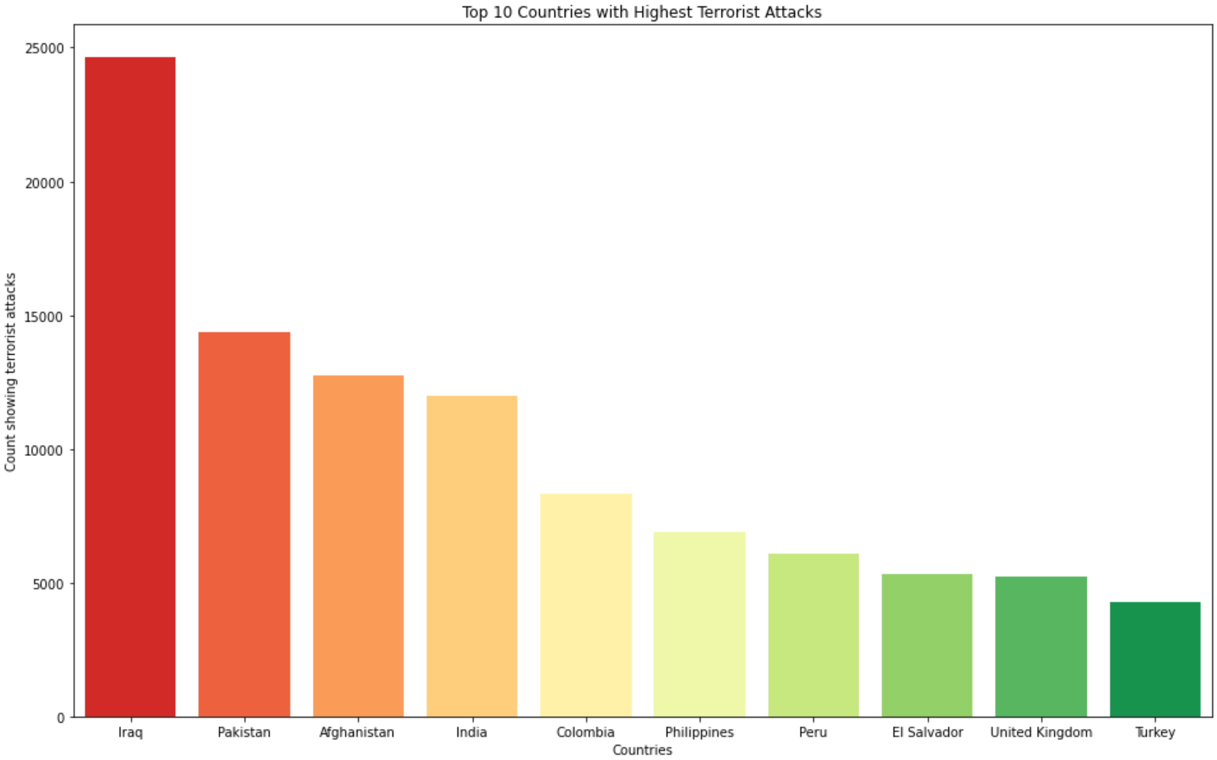
```
plt.figure(figsize=(16,10))
region=df['region_txt'].value_counts().index
sns.countplot(df['region_txt'],order=region[:10], palette = 'RdYlGn', saturation = 1
plt.xlabel("Region")
plt.ylabel('Count od Terrorist Attacks')
plt.title("Top 10 Regions showing Terrorist Attack Count")
plt.xticks(rotation= 90)
plt.show()
print("Region with the most attacks:",df['region_txt'].value_counts().idxmax())
print("With a count of", df['region_txt'].value_counts()[1])
```



```
Region with the most attacks: Middle East & North Africa
With a count of 44974
```

In [41]:

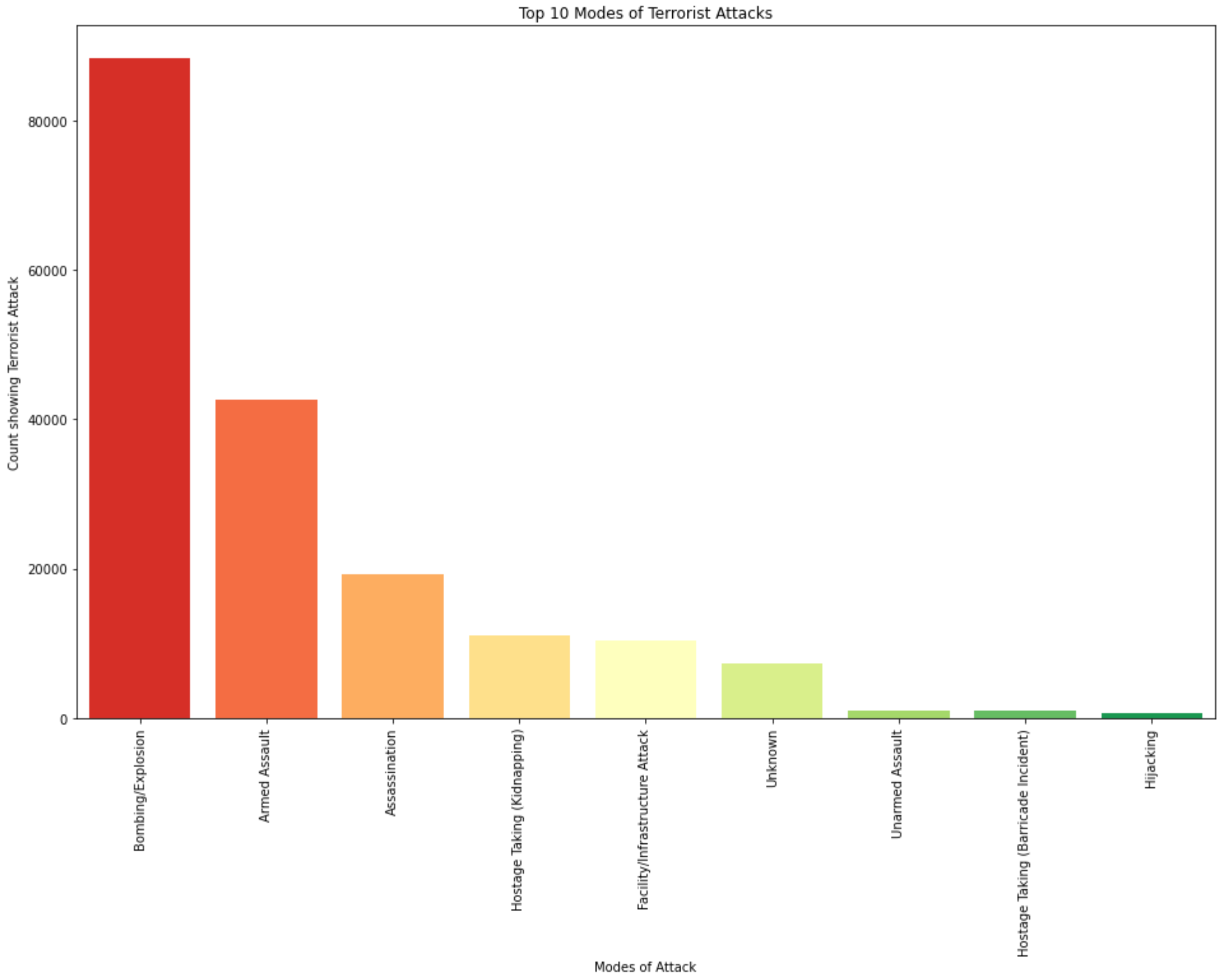
```
plt.figure(figsize=(16,10))
country=df['country_txt'].value_counts().index
sns.countplot(df['country_txt'], order = country[:10],palette = 'RdYlGn', saturation
plt.xlabel("Countries")
plt.ylabel('Count showing terrorist attacks')
plt.title('Top 10 Countries with Highest Terrorist Attacks')
plt.show()
print("Country with the most attacks:",df['country_txt'].value_counts().idxmax())
print("With a count of", df['country_txt'].value_counts()[1])
```



```
Country with the most attacks: Iraq
With a count of 14368
```

In [42]:

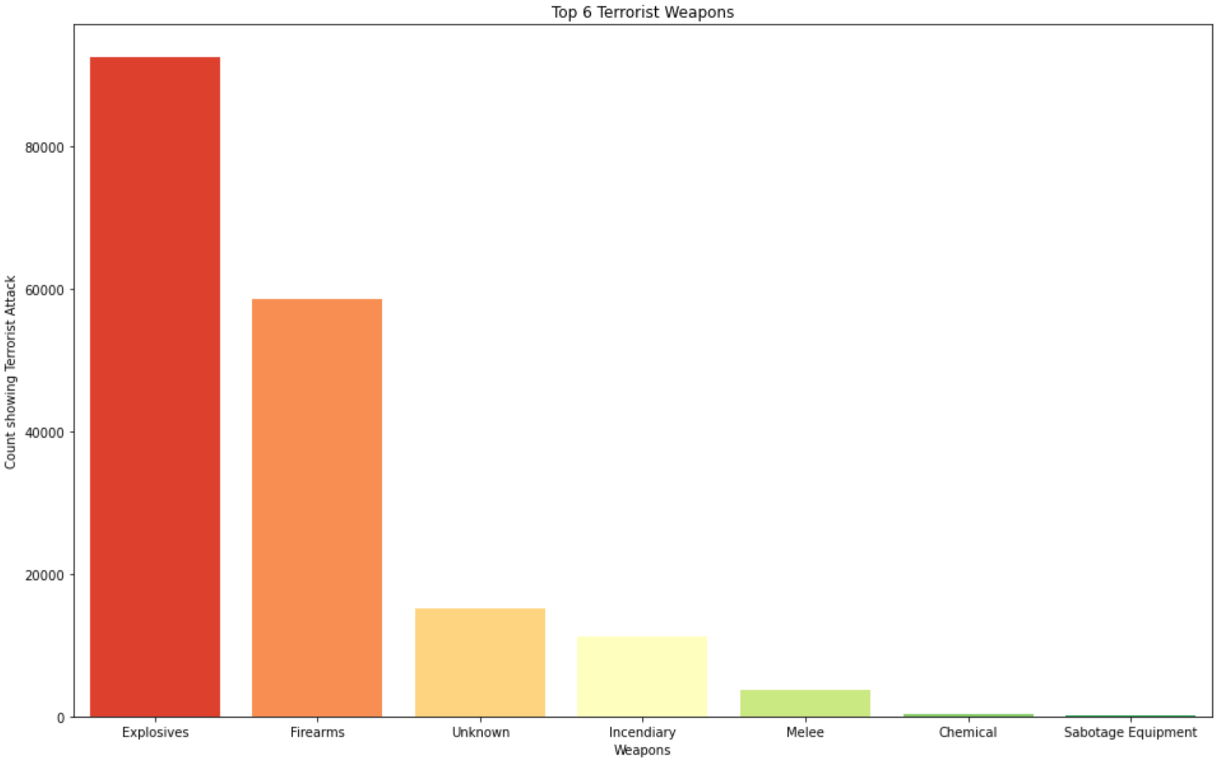
```
plt.figure(figsize=(16,10))
attack=df['attacktype1_txt'].value_counts().index
sns.countplot(df['attacktype1_txt'], order = attack[:10], palette = 'RdYlGn', satura
plt.xlabel("Modes of Attack")
plt.ylabel('Count showing Terrorist Attack')
plt.title('Top 10 Modes of Terrorist Attacks')
plt.xticks(rotation= 90)
plt.show()
print("Most Attack Types:",df['attacktype1_txt'].value_counts().idxmax())
print("With a count of", df['attacktype1_txt'].value_counts()[1])
```



```
Most Attack Types: Bombing/Explosion
With a count of 42669
```

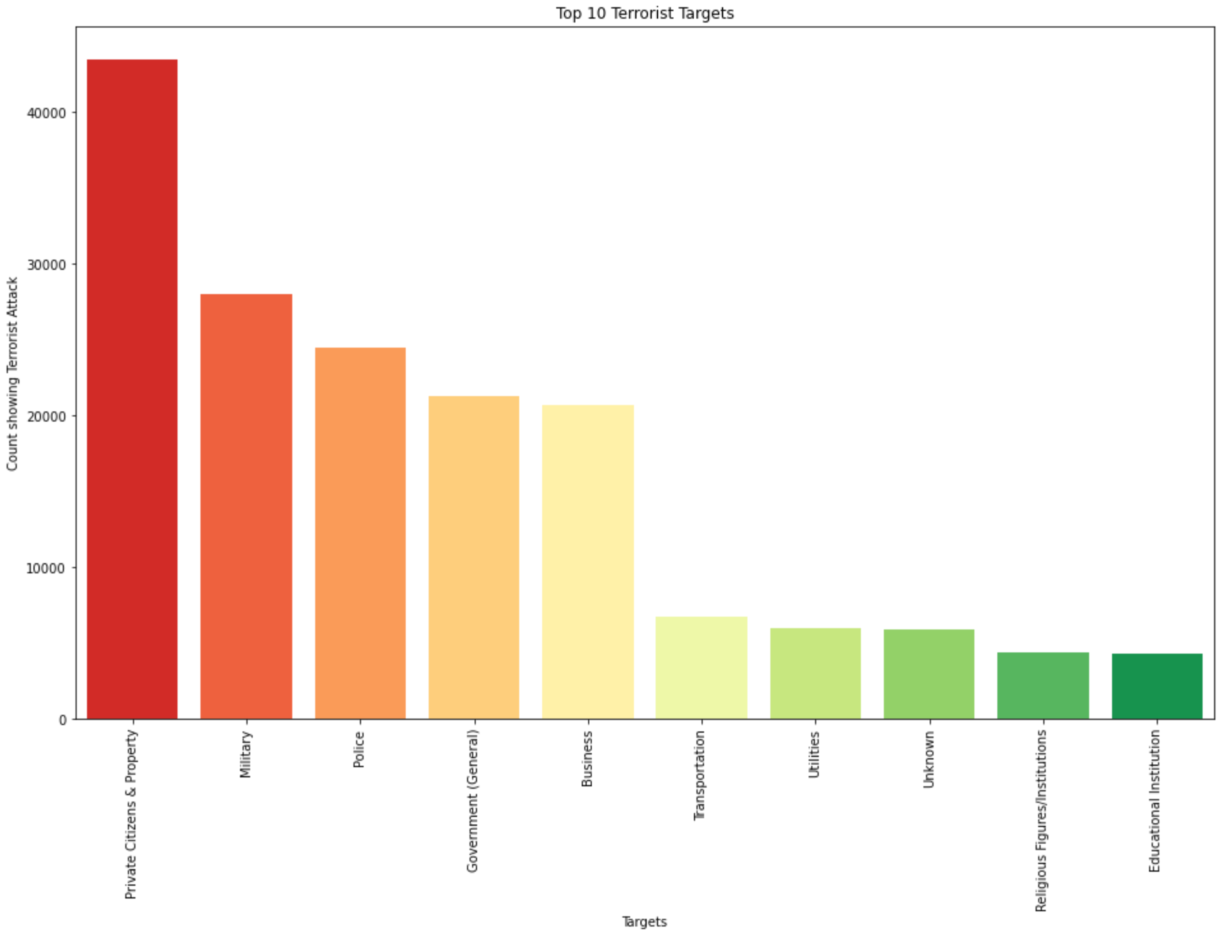
In [43]:

```python
plt.figure(figsize=(16,10))
weapon=df['weaptype1_txt'].value_counts().index
sns.countplot(df['weaptype1_txt'], order=weapon[0:7], palette = 'RdYlGn', saturation
plt.xlabel("Weapons")
plt.ylabel('Count showing Terrorist Attack')
plt.title('Top 6 Terrorist Weapons')
plt.show()
```



In [45]:

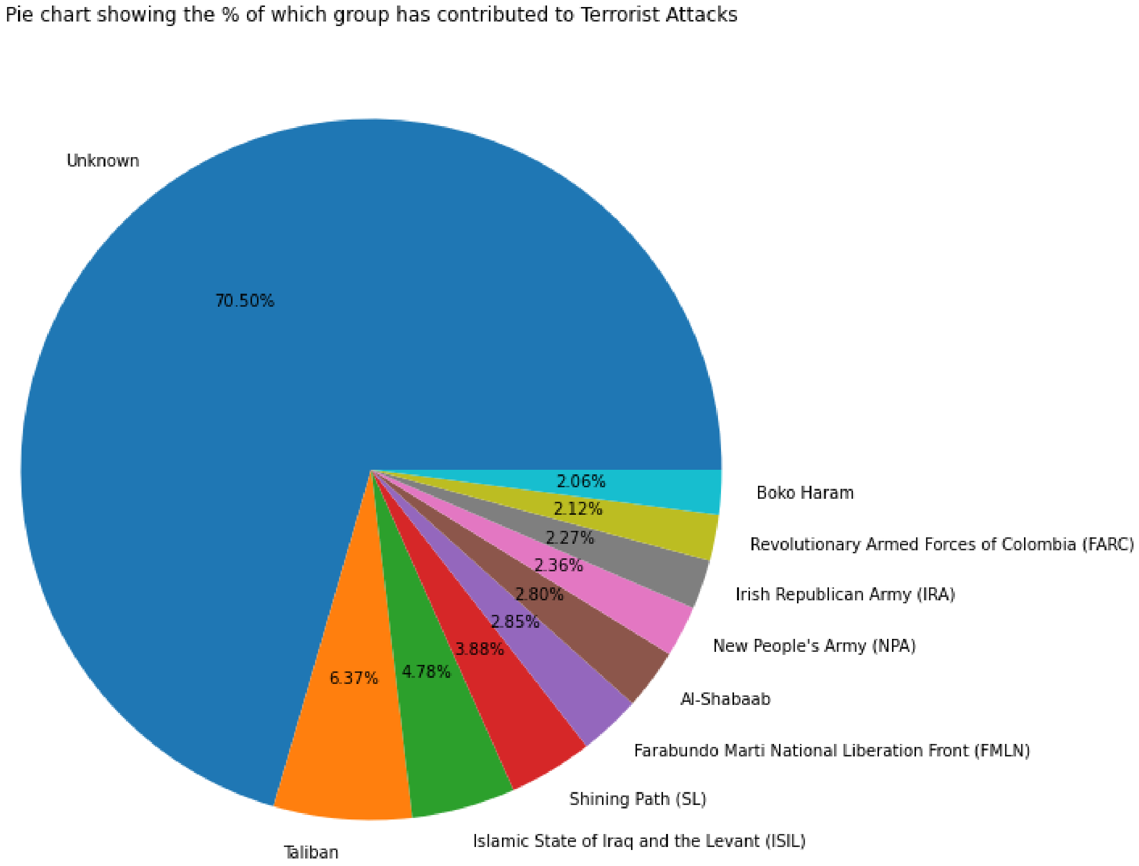
```
plt.figure(figsize=(16,10))
prop=df['targtype1_txt'].value_counts().index
sns.countplot(df['targtype1_txt'], order = prop[:10], palette = 'RdYlGn', saturation
plt.xlabel("Targets")
plt.ylabel('Count showing Terrorist Attack')
plt.title('Top 10 Terrorist Targets')
plt.xticks(rotation= 90)
plt.show()
print("Group suffering most attacks:",df['targtype1_txt'].value_counts().index[0])
print('With a count of', df['targtype1_txt'].value_counts()[0])
```



```
Group suffering most attacks: Private Citizens & Property
With a count of 43511
```

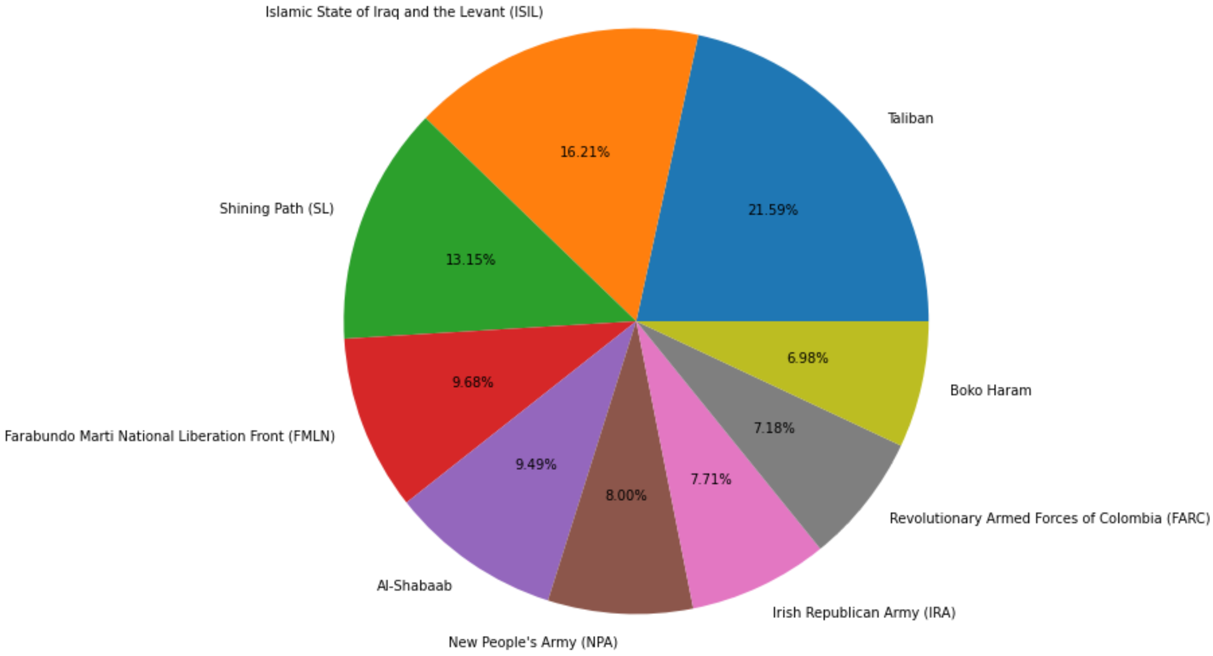
In [47]:

```
plt.figure(figsize=(20,10))
t_group= df['gname'].value_counts()
t_group
plt.pie(t_group[:10], labels=t_group.index[:10], autopct='%.2f%%')
plt.title('Pie chart showing the % of which group has contributed to Terrorist Attac
plt.show()
```



In [48]:

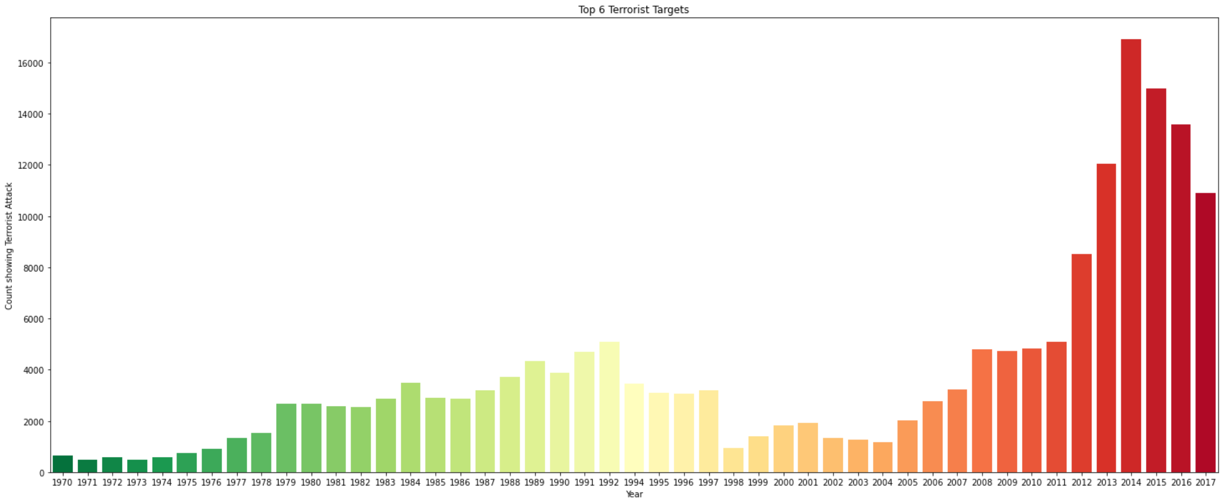
```
plt.figure(figsize=(20,10))
t_group= df['gname'].value_counts()
t_group
plt.pie(t_group[1:10], labels=t_group.index[1:10], autopct='%.2f%%')
plt.show()
```



In [53]:

```
plt.figure(figsize=(25,10))
year=df['iyear'].value_counts().index
sns.countplot(df['iyear'], palette = 'RdYlGn_r', saturation = 1)
```

```
plt.xlabel("Year")
plt.ylabel('Count showing Terrorist Attack')
plt.title('Top 6 Terrorist Targets')
plt.show()
print("Year with the most attacks:",df['iyear'].value_counts().idxmax())
```



```
Year with the most attacks: 2014
```

In [54]:

```
print('With the count of' , df['targtype1_txt'].value_counts()[0])
```

```
With the count of 43511
```

In [55]:

```
killings= df.groupby('iyear')['nkill'].sum()
killings = pd.DataFrame(killings)
killings= killings.reset_index()
killings = killings.sort_values('nkill', ascending = False)
killings.head()
```

Out[55]:

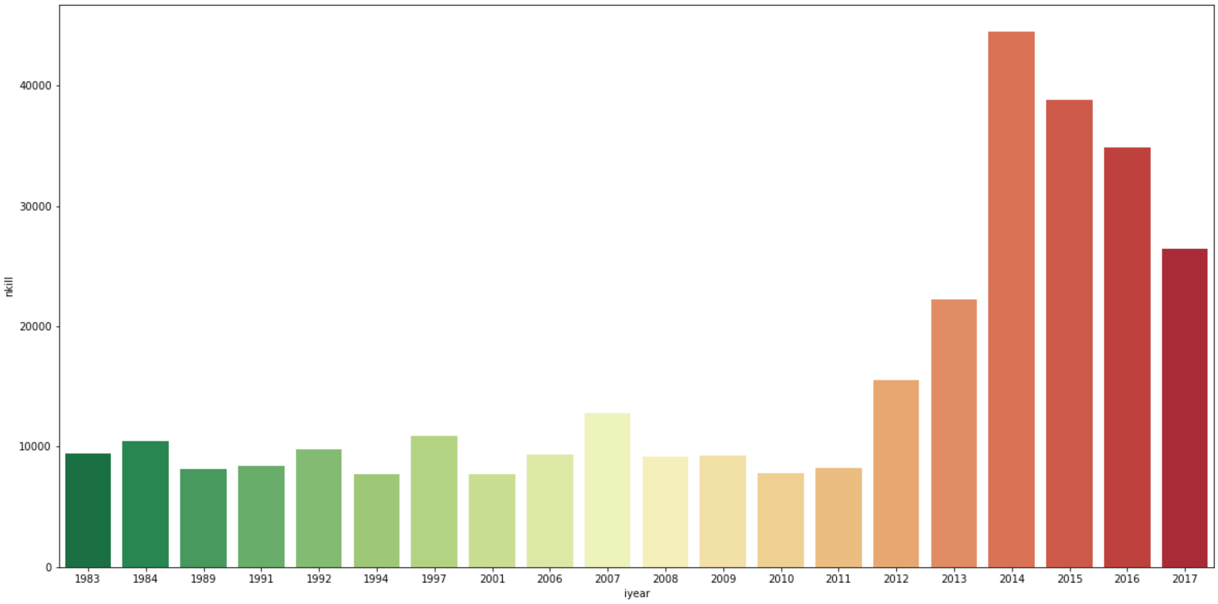
	iyear	nkill
43	2014	44490.0
44	2015	38853.0
45	2016	34871.0
46	2017	26445.0
42	2013	22273.0

In [56]:

```
plt.figure(figsize=(20,10))
sns.barplot(killings.iyear[0:20] , killings.nkill[0:20], palette='RdYlGn_r')
```

Out[56]:

```
<AxesSubplot:xlabel='iyear', ylabel='nkill'>
```

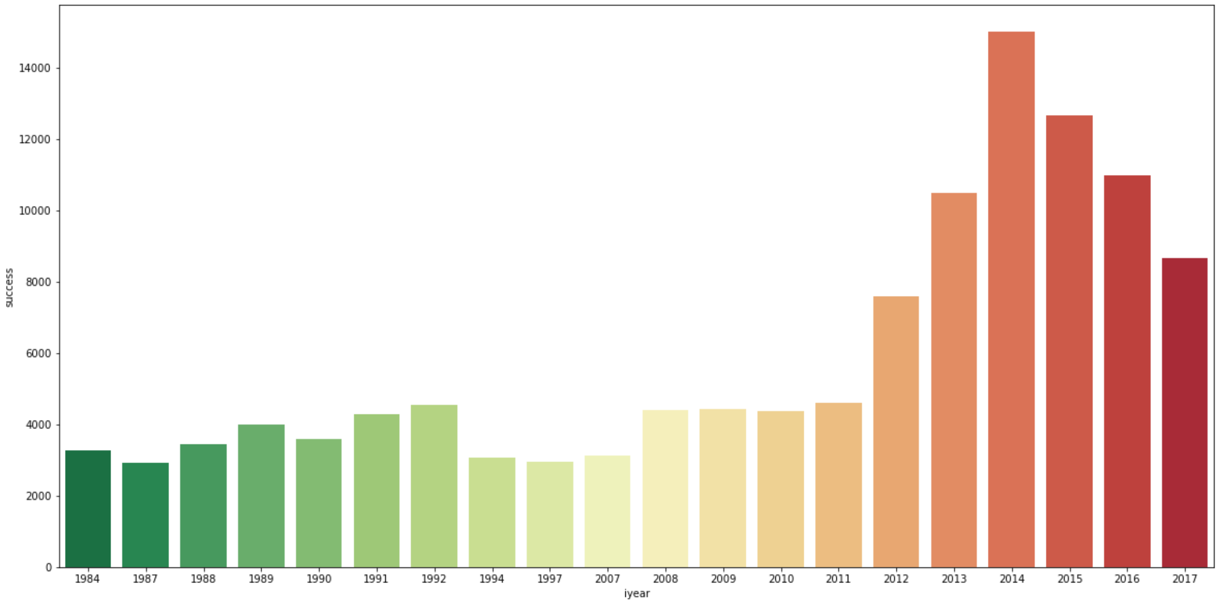


In [57]:

```
success= df.groupby('iyear')['success'].sum()
success = pd.DataFrame(success)
success= success.reset_index()
success = success.sort_values('success', ascending = False)
killings.head()
plt.figure(figsize=(20,10))
sns.barplot(success.iyear[0:20] , success.success[0:20], palette='RdYlGn_r')
```

Out[57]:

```
<AxesSubplot:xlabel='iyear', ylabel='success'>
```



In [60]:

```
success[:1]
```

Out[60]:

	iyear	success
43	2014	15015

In [61]:

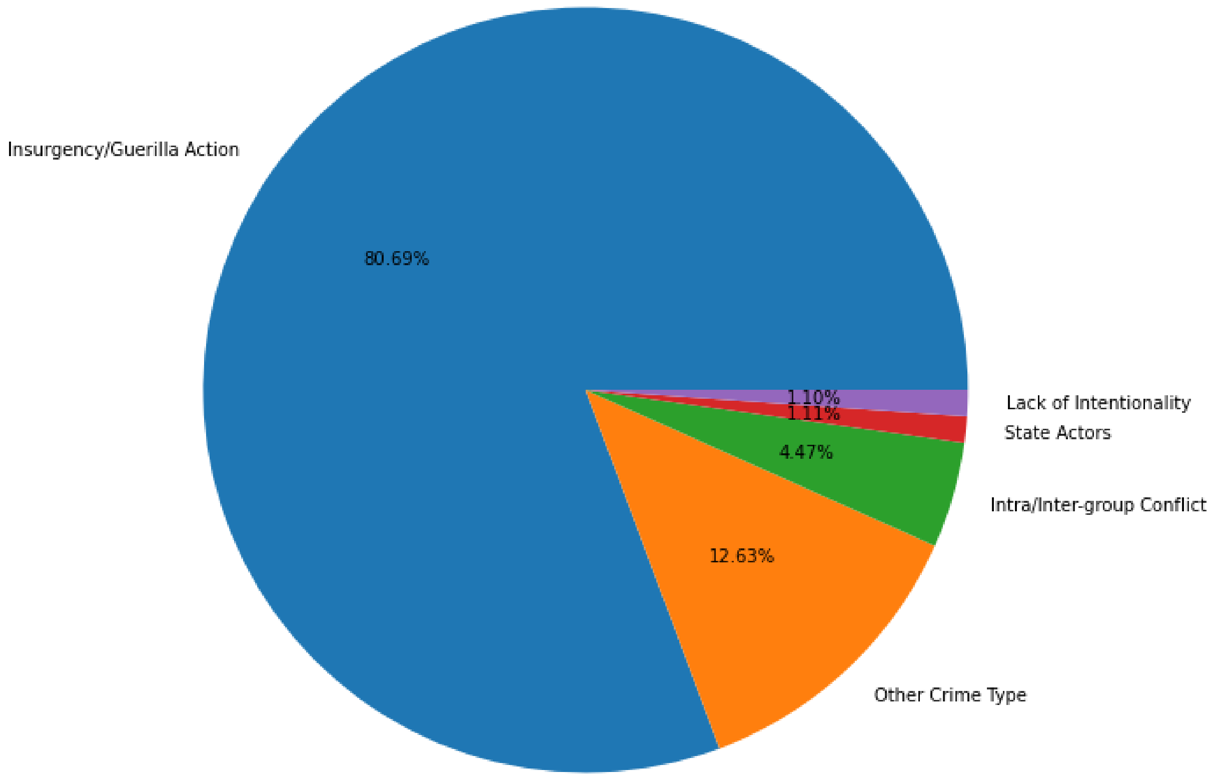
```
df.alternative_txt.unique()
```

Out[61]:

```
array([nan, 'Other Crime Type', 'Insurgency/Guerilla Action',
       'Intra/Inter-group Conflict', 'State Actors',
       'Lack of Intentionality'], dtype=object)
```

In [62]:

```
plt.figure(figsize=(20,10))
t_actions= df['alternative_txt'].value_counts()
t_actions
plt.pie(t_actions, labels=t_actions.index, autopct='%.2f%%')
plt.show()
```



In [63]:

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

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

In [66]:

```
attackData = df.loc[:, 'attacktype1_txt']

typeKill = pd.concat([attackData, killData], axis=1)
typeKill.head()
```

Out[66]:

	attacktype1_txt	nkill
0	Assassination	1.0
1	Hostage Taking (Kidnapping)	0.0
2	Assassination	1.0
3	Bombing/Explosion	NaN
4	Facility/Infrastructure Attack	NaN

In [67]:

```
values = typeKill.value_counts()
```

In [68]:

```
typeKillAgg = typeKill.pivot_table(columns='attacktype1_txt', values='nkill', aggfun
typeKillAgg
```

Out[68]:

attacktype1_txt	Armed Assault	Assassination	Bombing/Explosion	Facility/Infrastructure Attack	Hijacking	Hos Ta (Barri Incic
nkill	160297.0	24920.0	157321.0	3642.0	3718.0	44

In [79]:

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

In [81]:

```
countryKillData.head()
```

Out[81]:

	country_txt	nkill
0	Dominican Republic	1.0
1	Mexico	0.0
2	Philippines	1.0
3	Greece	NaN
4	Japan	NaN

In [82]:

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

Out[82]:

country_txt	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Austra
nkill	39384.0	42.0	11066.0	0.0	3043.0	0.0	490.0	37.0	2

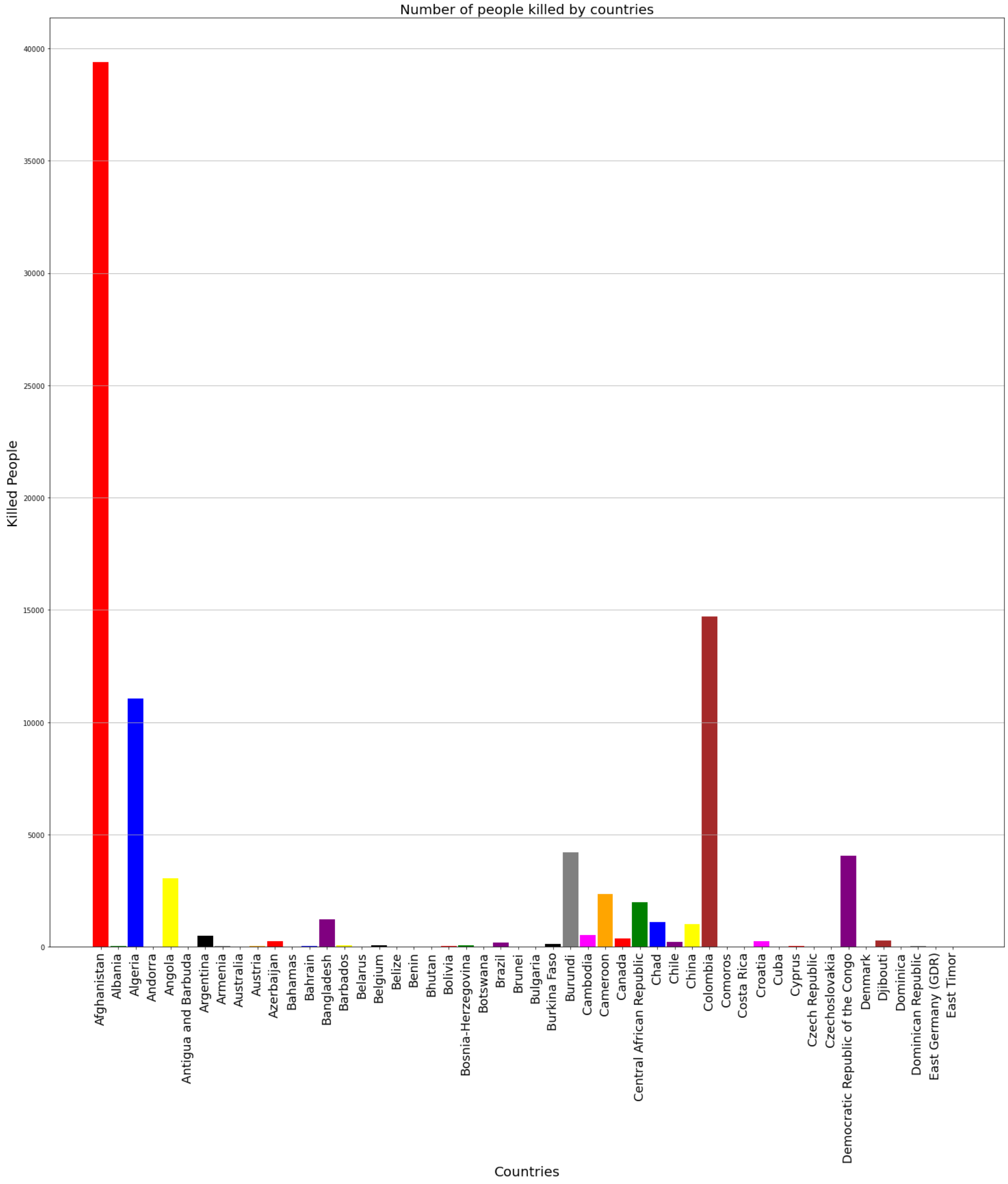
1 rows × 205 columns

In [83]:

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

In [84]:

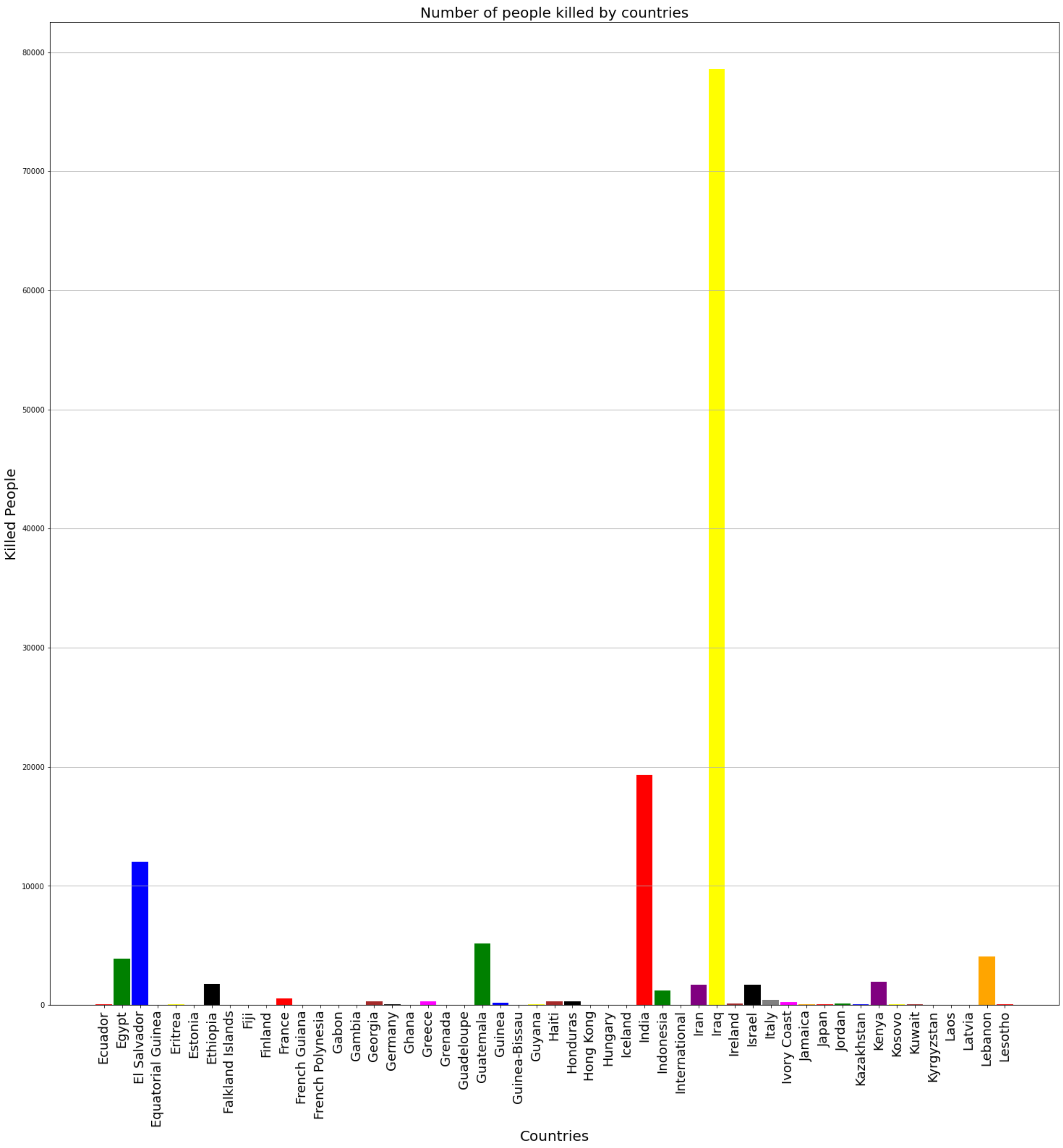
```
labels = countryKillFormatData.columns.tolist()
labels = labels[:50] #50 bar provides nice view
index = np.arange(len(labels))
transpoze = countryKillFormatData.T
values = transpoze.values.tolist()
values = values[:50]
values = [int(i[0]) for i in values]
colors = ['red', 'green', 'blue', 'purple', 'yellow', 'brown', 'black', 'gray', 'mag
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()
```



In [85]:

```
labels = countryKillFormatData.columns.tolist()
labels = labels[50:101]
```

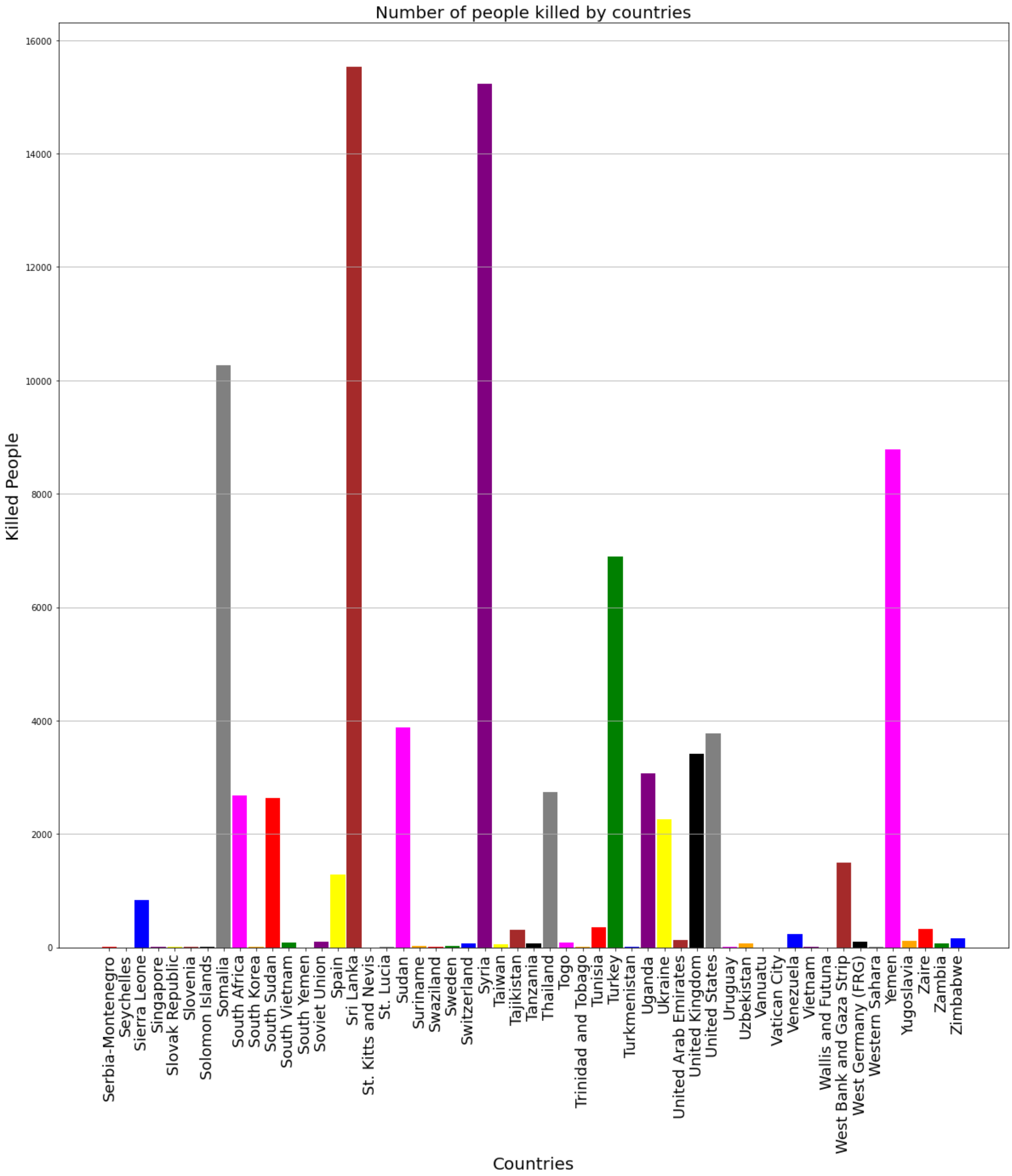
```
index = np.arange(len(labels))
transpoze = countryKillFormatData.T
values = transpoze.values.tolist()
values = values[50:101]
values = [int(i[0]) for i in values]
colors = ['red', 'green', 'blue', 'purple', 'yellow', 'brown', 'black', 'gray', 'mag
fig, ax = plt.subplots(1, 1)
ax.yaxis.grid(True)
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 [86]:

```
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', 'mag
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()
```



Terrorist activities are maximum in the Middle East and northern Africa. They are seen to be the places of serious terrorist attacks. From the charts we also see that appears that Iraq, Afghanistan and Pakistan are the most damaged countries.

In [ ]:

In [ ]:

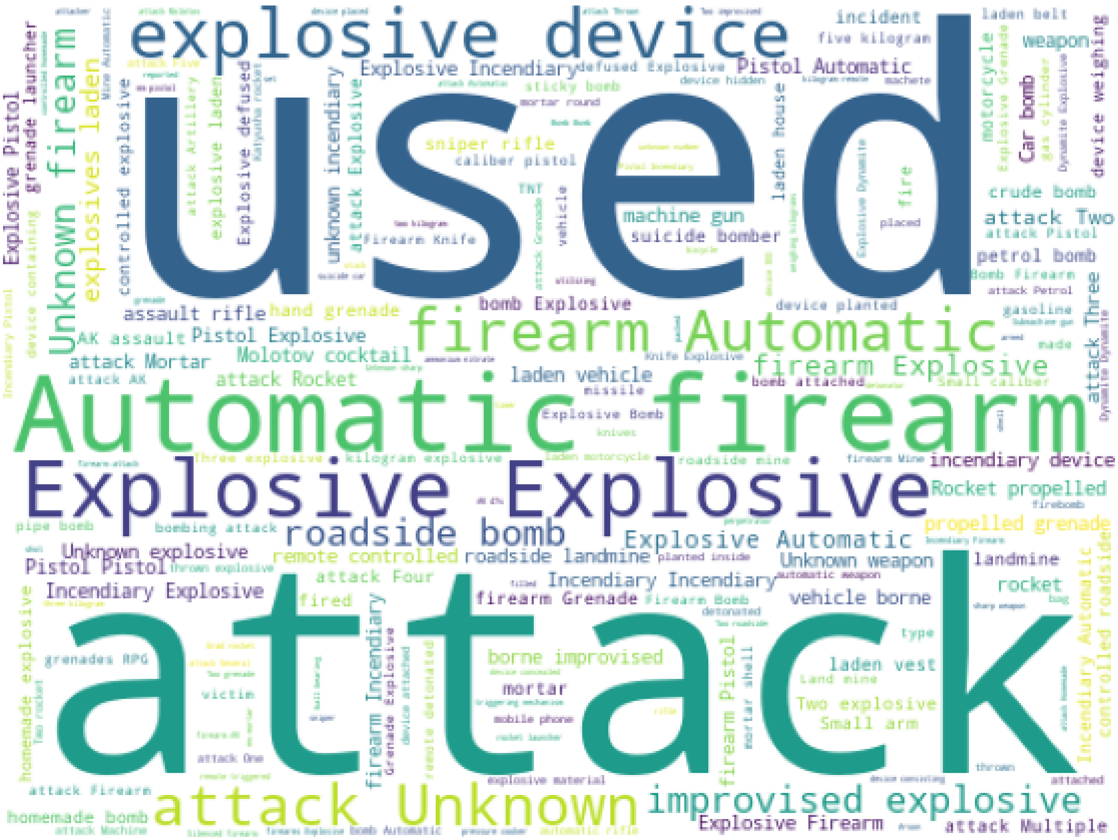
In [ ]:

In [ ]:

In [ ]:

In [87]:

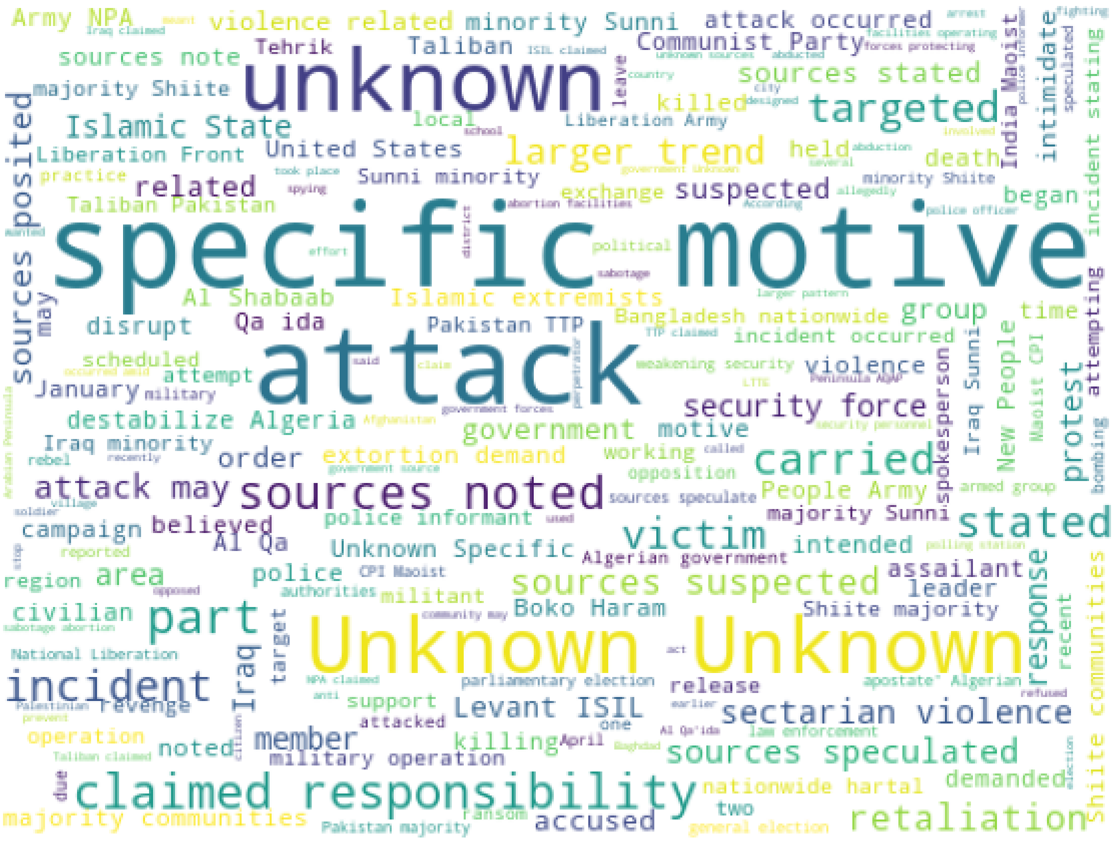
```
from wordcloud import WordCloud
from scipy import signal
weapdetail = df.weapdetail.dropna(False)
plt.subplots(figsize=(10,10))
wordcloud = WordCloud(background_color = 'white',
                      width = 512,
                      height = 384).generate(' '.join(weapdetail))
plt.axis('off')
plt.imshow(wordcloud)
plt.show()
```



In [88]:

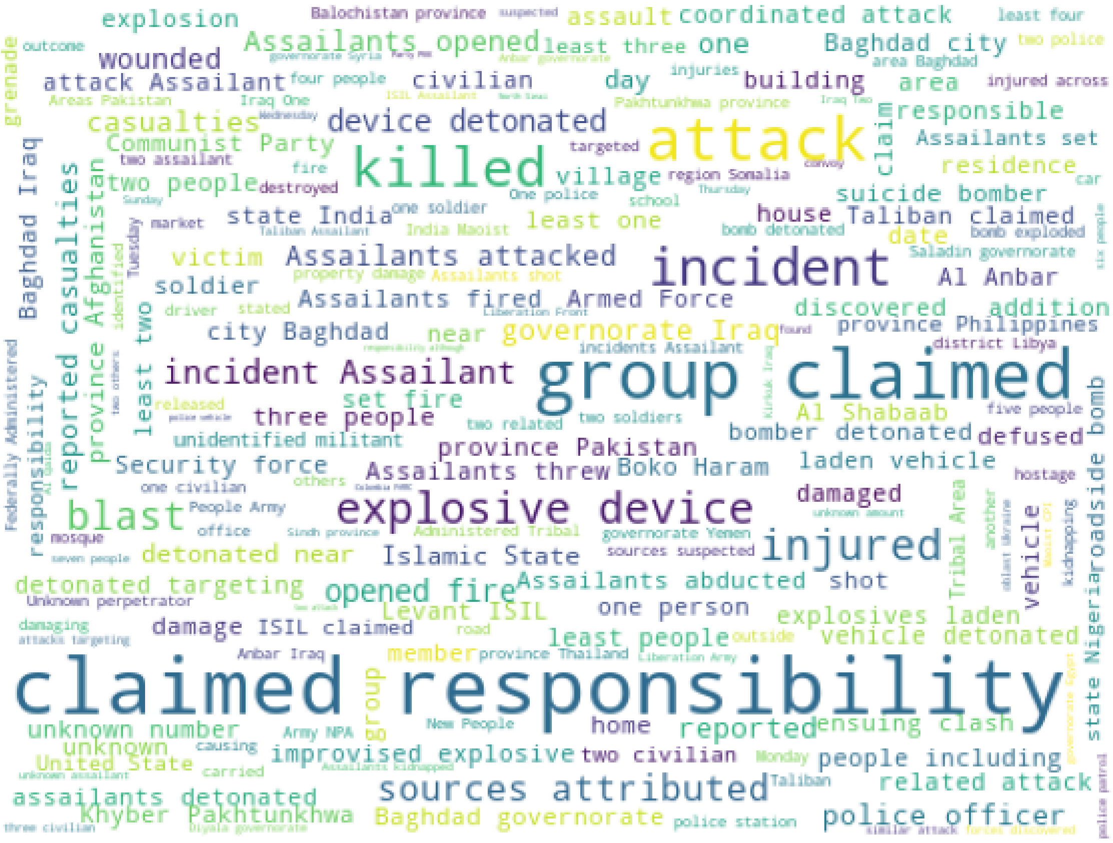
```
from wordcloud import WordCloud
from scipy import signal
motive = df.motive.dropna(False)
plt.subplots(figsize=(10,10))
```

```
wordcloud = WordCloud(background_color = 'white',
                      width = 512,
                      height = 384).generate(' '.join(motive))
plt.axis('off')
plt.imshow(wordcloud)
plt.show()
```



In [89]:

```
from wordcloud import WordCloud
from scipy import signal
news = df.summary.dropna(False)
plt.subplots(figsize=(10,10))
wordcloud = WordCloud(background_color = 'white',
                      width = 512,
                      height = 384).generate(' '.join(news))
plt.axis('off')
plt.imshow(wordcloud)
plt.show()
```



In [90]:

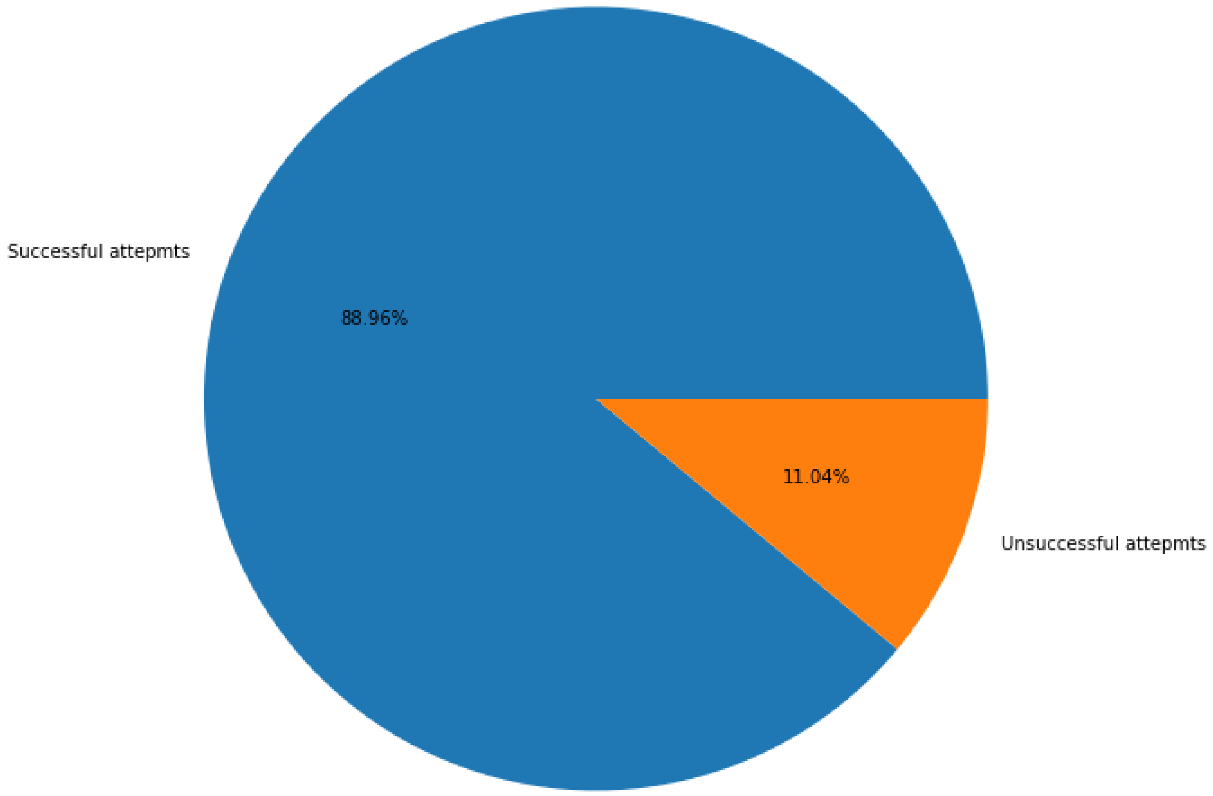
```
df.success.value_counts(normalize=True)
```

Out[90]:

```
1    0.889598
0    0.110402
Name: success, dtype: float64
```

In [91]:

```
plt.figure(figsize=(20,10))
df_success= df.success.value_counts()
plt.pie(df_success, labels = ['Successful attepmts' , 'Unsuccessful attepmts'], auto
plt.show()
```



In [92]:

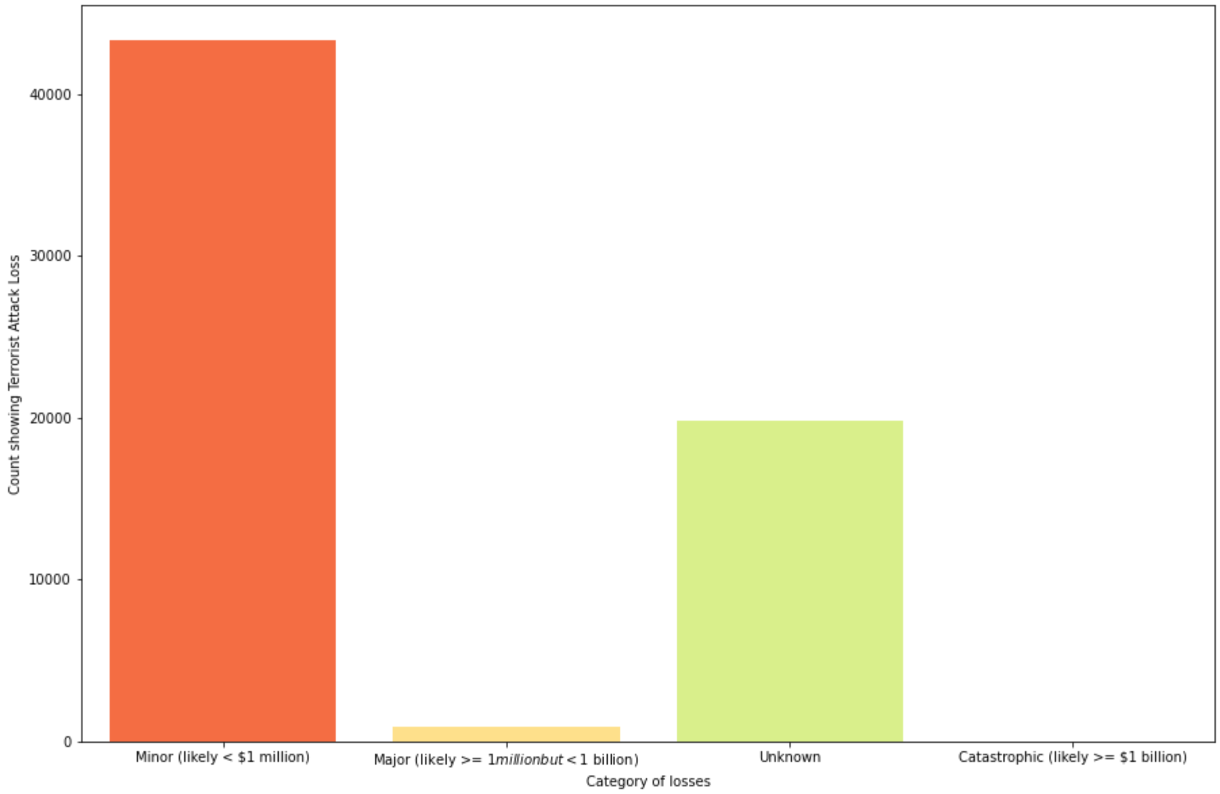
```
df.propextent_txt.value_counts()
```

Out[92]:

```
Minor (likely < $1 million)                        43304
Unknown                                            19846
Major (likely >= $1 million but < $1 billion)        909
Catastrophic (likely >= $1 billion)                    6
Name: propextent_txt, dtype: int64
```

In [94]:

```
plt.figure(figsize=(15,10))
year=df['propextent_txt'].value_counts().index
sns.countplot(df['propextent_txt'], palette = 'RdYlGn', saturation = 1)
plt.xlabel("Category of losses")
plt.ylabel('Count showing Terrorist Attack Loss')
plt.show()
```



In [3]:

```
df1= df[df.columns[df.isnull().mean()<=0.45]]
```

In [4]:

```
df1.columns
```

Out[4]:

```
Index(['eventid', 'iyear', 'imonth', 'iday', 'extended', 'country',
       'country_txt', 'region', 'region_txt', 'provstate', 'city', 'latitude',
       'longitude', 'specificity', 'vicinity', 'summary', 'crit1', 'crit2',
       'crit3', 'doubtterr', 'multiple', 'success', 'suicide', 'attacktype1',
       'attacktype1_txt', 'targtype1', 'targtype1_txt', 'targsubtype1',
       'targsubtype1_txt', 'corp1', 'target1', 'natlty1', 'natlty1_txt',
       'gname', 'guncertain1', 'individual', 'nperps', 'nperpcap', 'claimed',
       'weaptype1', 'weaptype1_txt', 'weapsubtype1', 'weapsubtype1_txt',
       'weapdetail', 'nkill', 'nkillus', 'nkillter', 'nwound', 'nwoundus',
       'nwoundte', 'property', 'ishostkid', 'scite1', 'dbsource', 'INT_LOG',
       'INT_IDEO', 'INT_MISC', 'INT_ANY'],
      dtype='object')
```

In [5]:

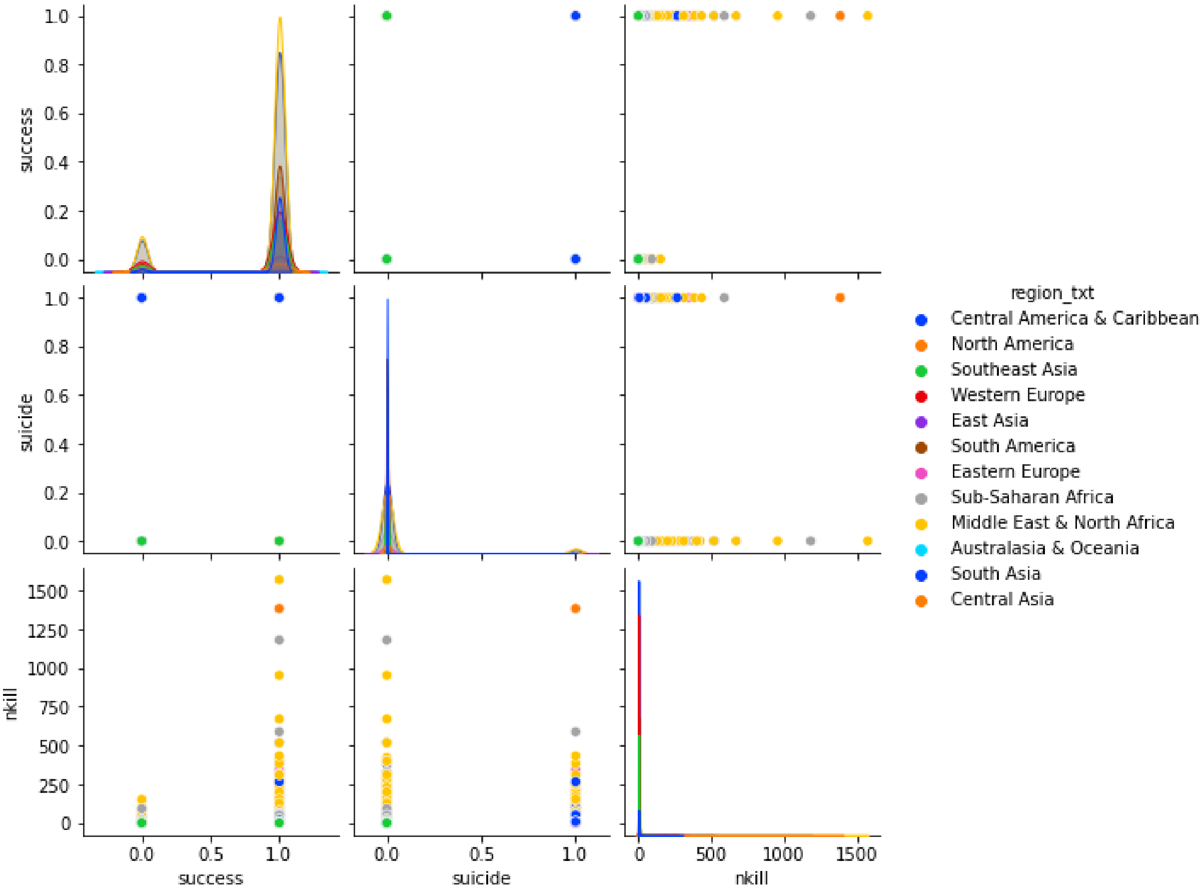
```
df1_success=df1.loc[:,['success','country_txt', 'region_txt','provstate','city','sui
```

In [12]:

```
import seaborn as sns
sns.pairplot(df1_success, hue = 'region_txt', palette = 'bright')
```

Out[12]:

```
<seaborn.axisgrid.PairGrid at 0x1f5ed949a60>
```



In [98]:

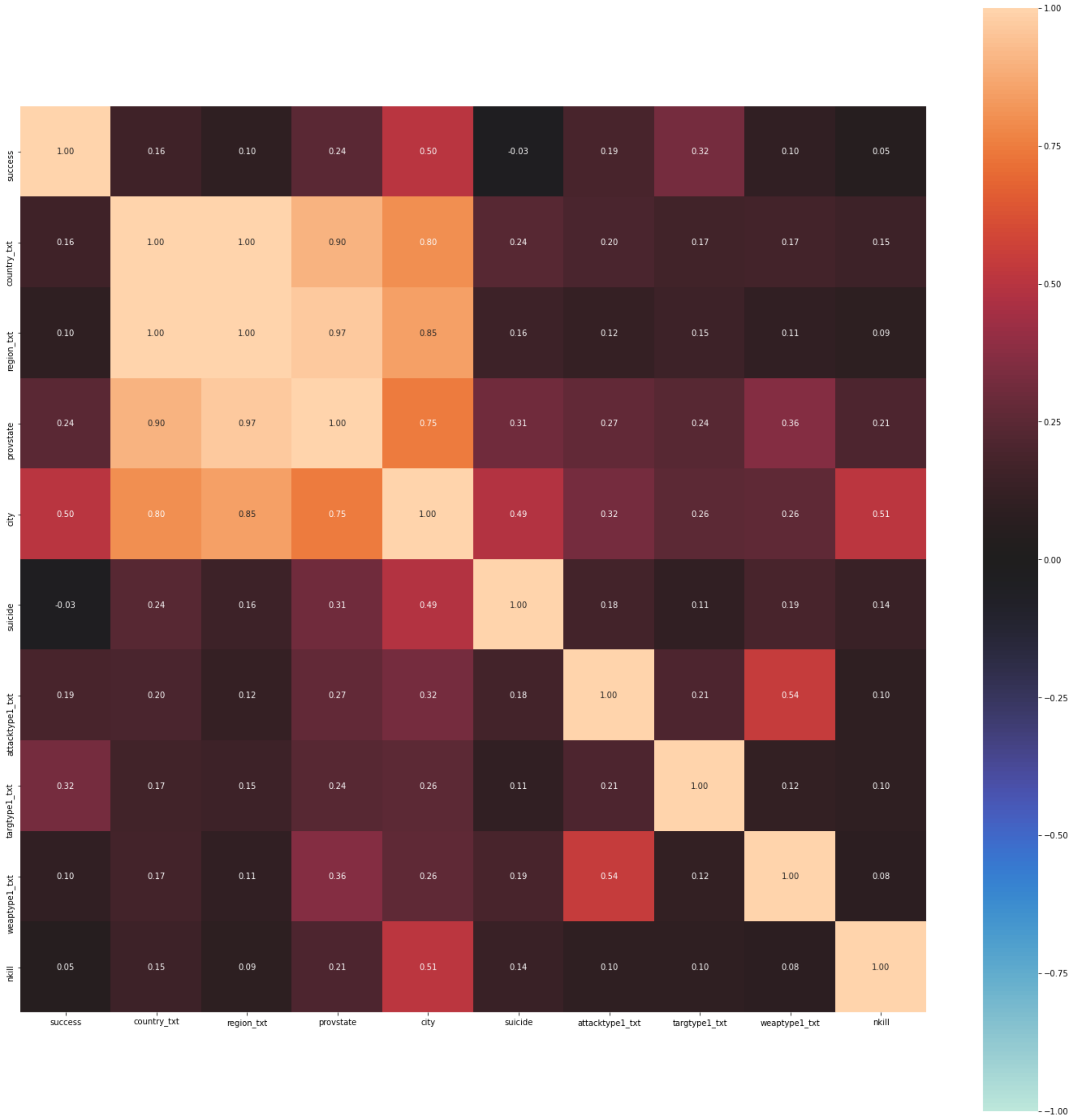
```
df1_success.head()
```

Out[98]:

	success	country_txt	region_txt	provstate	city	suicide	attacktype1_txt	targtype1_txt
0	1	Dominican Republic	Central America & Caribbean	NaN	Santo Domingo	0	Assassination	Private Citizens & Property
1	1	Mexico	North America	Federal	Mexico city	0	Hostage Taking (Kidnapping)	Government (Diplomatic)
2	1	Philippines	Southeast Asia	Tarlac	Unknown	0	Assassination	Journalists & Media
3	1	Greece	Western Europe	Attica	Athens	0	Bombing/Explosion	Government (Diplomatic)
4	1	Japan	East Asia	Fukouka	Fukouka	0	Facility/Infrastructure Attack	Government (Diplomatic)

In [99]:

```
nominal.associations(df1_success)
```



```
Out[99]: {'corr':                  success  country_txt  region_txt  provstate      city  \
          success          1.000000     0.158759    0.096993   0.243625  0.503338
          country_txt      0.158759     1.000000    0.999469   0.902442  0.804118
          region_txt       0.096993     0.999469    1.000000   0.965093  0.850009
          provstate        0.243625     0.902442    0.965093   1.000000  0.745677
          city             0.503338     0.804118    0.850009   0.745677  1.000000
          suicide         -0.031155     0.235070    0.156143   0.307699  0.486420
          attacktype1_txt  0.189691     0.197661    0.122945   0.270913  0.323185
          targtype1_txt    0.324246     0.168503    0.150892   0.238596  0.257328
          weaptype1_txt    0.103838     0.174963    0.107463   0.359392  0.259322
          nkill            0.049619     0.149046    0.090377   0.207097  0.511938

                            suicide  attacktype1_txt  targtype1_txt  weaptype1_txt  \
          success         -0.031155         0.189691       0.324246       0.103838
          country_txt      0.235070         0.197661       0.168503       0.174963
          region_txt       0.156143         0.122945       0.150892       0.107463
          provstate        0.307699         0.270913       0.238596       0.359392
          city             0.486420         0.323185       0.257328       0.259322
          suicide          1.000000         0.177893       0.107878       0.188939
          attacktype1_txt  0.177893         1.000000       0.210434       0.541028
          targtype1_txt    0.107878         0.210434       1.000000       0.116891
          weaptype1_txt    0.188939         0.541028       0.116891       1.000000
          nkill            0.136385         0.098263       0.100145       0.080035

                              nkill
          success          0.049619
          country_txt      0.149046
          region_txt       0.090377
          provstate        0.207097
          city             0.511938
          suicide          0.136385
          attacktype1_txt  0.098263
          targtype1_txt    0.100145
          weaptype1_txt    0.080035
          nkill            1.000000  ,
          'ax': <AxesSubplot:>}
```

Thank You for your time

```
In [ ]:
```