TSF TASK 4 - GRIP MARCH'21

ADITYA AMBWANI

Exploratory Data Analysis - Terrorism

Objective:-

- Perform 'Exploratory Data Analysis' on dataset 'Global Terrorism'
- As a security/defense analyst, try to find out the hot zone of terrorism.
- What all security issues and insights you can derive by EDA?

In [4]:

```
# Importing libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
!pip install wordcloud
```

```
Requirement already satisfied: wordcloud in c:\users\adi\anaconda3\lib\site-packages (1.
8.1)
Requirement already satisfied: pillow in c:\users\adi\anaconda3\lib\site-packages (from
wordcloud) (8.0.1)
Requirement already satisfied: matplotlib in c:\users\adi\anaconda3\lib\site-packages (f
rom wordcloud) (3.3.2)
Requirement already satisfied: numpy>=1.6.1 in c:\users\adi\anaconda3\lib\site-packages
(from wordcloud) (1.19.2)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\adi\anaconda3\lib\site-pack
ages (from matplotlib->wordcloud) (1.3.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in c:\users\adi
\anaconda3\lib\site-packages (from matplotlib->wordcloud) (2.4.7)
Requirement already satisfied: cycler>=0.10 in c:\users\adi\anaconda3\lib\site-packages
(from matplotlib->wordcloud) (0.10.0)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\adi\anaconda3\lib\site-p
ackages (from matplotlib->wordcloud) (2.8.1)
Requirement already satisfied: certifi>=2020.06.20 in c:\users\adi\anaconda3\lib\site-pa
ckages (from matplotlib->wordcloud) (2020.6.20)
Requirement already satisfied: six in c:\users\adi\anaconda3\lib\site-packages (from cyc
ler>=0.10->matplotlib->wordcloud) (1.15.0)
```

In [5]:

```
# Reading our dataset
data=pd.read_csv('globalterrorismdb_0718dist.csv',encoding='Latin1')
data
```

Out[5]:

	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	country_txt	re
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	

	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	country_txt	re
3	197001000002	1970	1	0	NaN	0	NaN	78	Greece	
4	197001000003	1970	1	0	NaN	0	NaN	101	Japan	
...	...	...	...	...	...	...	...	...	...	
181686	201712310022	2017	12	31	NaN	0	NaN	182	Somalia	
181687	201712310029	2017	12	31	NaN	0	NaN	200	Syria	
181688	201712310030	2017	12	31	NaN	0	NaN	160	Philippines	
181689	201712310031	2017	12	31	NaN	0	NaN	92	India	
181690	201712310032	2017	12	31	NaN	0	NaN	160	Philippines	

181691 rows × 135 columns

In [6]:
```
# Calculating coloumns and rows
data.shape
```

Out[6]:
```
(181691, 135)
```

In [7]:
```
# Seeing all the headings of coloumn
data.columns.values
```

Out[7]:
```
array(['eventid', 'iyear', 'imonth', 'iday', 'approxdate', 'extended',
       'resolution', 'country', 'country_txt', 'region', 'region_txt',
       'provstate', 'city', 'latitude', 'longitude', 'specificity',
       'vicinity', 'location', 'summary', 'crit1', 'crit2', 'crit3',
       'doubtterr', 'alternative', 'alternative_txt', 'multiple',
       'success', 'suicide', 'attacktype1', 'attacktype1_txt',
       'attacktype2', 'attacktype2_txt', 'attacktype3', 'attacktype3_txt',
       'targtype1', 'targtype1_txt', 'targsubtype1', 'targsubtype1_txt',
       'corp1', 'target1', 'natlty1', 'natlty1_txt', 'targtype2',
       'targtype2_txt', 'targsubtype2', 'targsubtype2_txt', 'corp2',
```

```
       'target2', 'natlty2', 'natlty2_txt', 'targtype3', 'targtype3_txt',
       'targsubtype3', 'targsubtype3_txt', 'corp3', 'target3', 'natlty3',
       'natlty3_txt', 'gname', 'gsubname', 'gname2', 'gsubname2',
       'gname3', 'gsubname3', 'motive', 'guncertain1', 'guncertain2',
       'guncertain3', 'individual', 'nperps', 'nperpcap', 'claimed',
       'claimmode', 'claimmode_txt', 'claim2', 'claimmode2',
       'claimmode2_txt', 'claim3', 'claimmode3', 'claimmode3_txt',
       'compclaim', 'weaptype1', 'weaptype1_txt', 'weapsubtype1',
       'weapsubtype1_txt', 'weaptype2', 'weaptype2_txt', 'weapsubtype2',
       'weapsubtype2_txt', 'weaptype3', 'weaptype3_txt', 'weapsubtype3',
       'weapsubtype3_txt', 'weaptype4', 'weaptype4_txt', 'weapsubtype4',
       'weapsubtype4_txt', 'weapdetail', 'nkill', 'nkillus', 'nkillter',
       'nwound', 'nwoundus', 'nwoundte', 'property', 'propextent',
       'propextent_txt', 'propvalue', 'propcomment', 'ishostkid',
       'nhostkid', 'nhostkidus', 'nhours', 'ndays', 'divert',
       'kidhijcountry', 'ransom', 'ransomamt', 'ransomamtus',
       'ransompaid', 'ransompaidus', 'ransomnote', 'hostkidoutcome',
       'hostkidoutcome_txt', 'nreleased', 'addnotes', 'scite1', 'scite2',
       'scite3', 'dbsource', 'INT_LOG', 'INT_IDEO', 'INT_MISC', 'INT_ANY',
       'related'], dtype=object)
```

In [8]:
```
# Checking for some more information on dataset
data.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 [9]:
```
# Describing the dataset in a summarized way
data.describe()
```

Out[9]:

	eventid	iyear	imonth	iday	extended	country	
count	1.816910e+05	181691.000000	181691.000000	181691.000000	181691.000000	181691.000000	18169
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	1
max	2.017123e+11	2017.000000	12.000000	31.000000	1.000000	1004.000000	1

8 rows × 77 columns

In [10]:
```
# Checking for null values in our dataset
data.isnull().sum()
```

Out[10]:
```
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 [11]:
```
# Dropping all the null values
# data.dropna(axis=1,inplace=True)
# data.isnull().sum()
```

In [12]:
```
# Seeing correlation between different variables
data.corr()
```

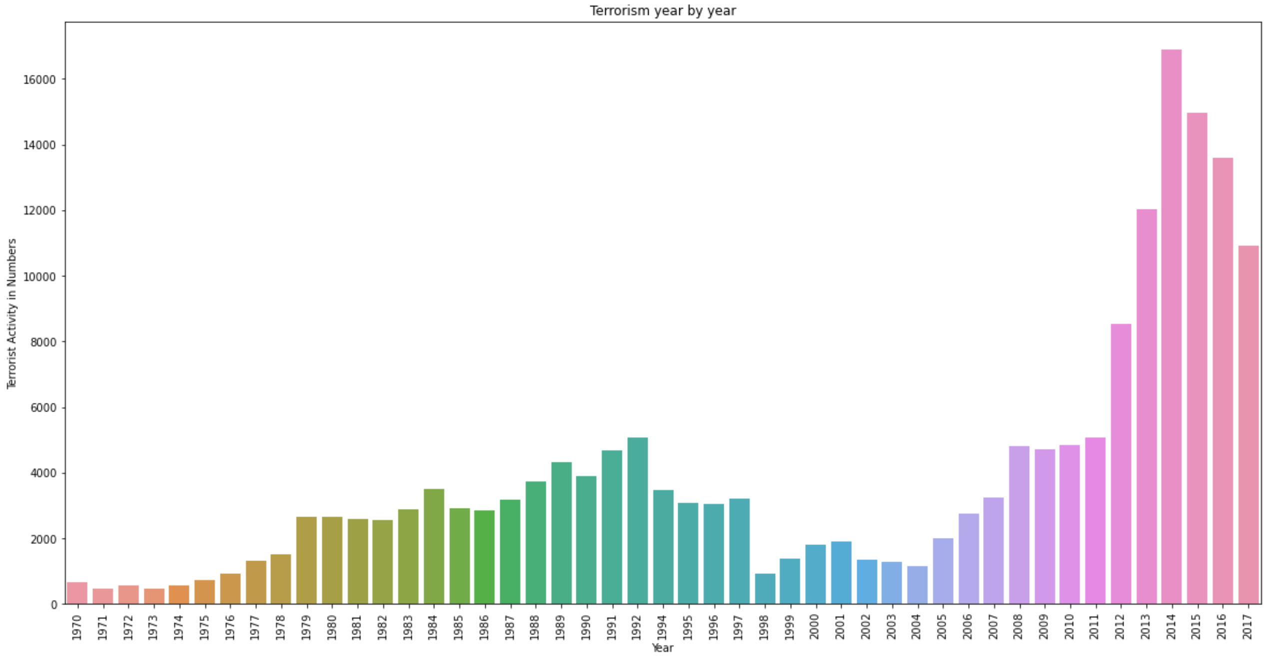
Out[12]:

	eventid	iyear	imonth	iday	extended	country	region	latitude	longitu
eventid	1.000000	0.999996	0.002706	0.018336	0.091761	-0.135039	0.401371	0.166886	0.0039
iyear	0.999996	1.000000	0.000139	0.018254	0.091754	-0.135023	0.401384	0.166933	0.0039
imonth	0.002706	0.000139	1.000000	0.005497	-0.000468	-0.006305	-0.002999	-0.015978	-0.0038
iday	0.018336	0.018254	0.005497	1.000000	-0.004700	0.003468	0.009710	0.003423	-0.0022
extended	0.091761	0.091754	-0.000468	-0.004700	1.000000	-0.020466	0.038389	-0.024749	0.0005
...	...	...	...	...	...	...	...	...	
nreleased	-0.181612	-0.181556	-0.011535	0.001765	-0.192155	-0.044331	-0.149511	0.002790	-0.0177
INT_LOG	-0.143600	-0.143601	-0.002302	-0.001540	0.071768	0.069904	-0.082584	-0.099827	0.0022
INT_IDEO	-0.133252	-0.133253	-0.002034	-0.001621	0.075147	0.067564	-0.071917	-0.094470	0.0022
INT_MISC	-0.077852	-0.077847	-0.002554	-0.002027	0.027335	0.207281	0.043139	0.097652	0.0003
INT_ANY	-0.175605	-0.175596	-0.006336	-0.001199	0.080767	0.153118	-0.047900	-0.041530	0.0024

77 rows × 77 columns

Visualising Dataset

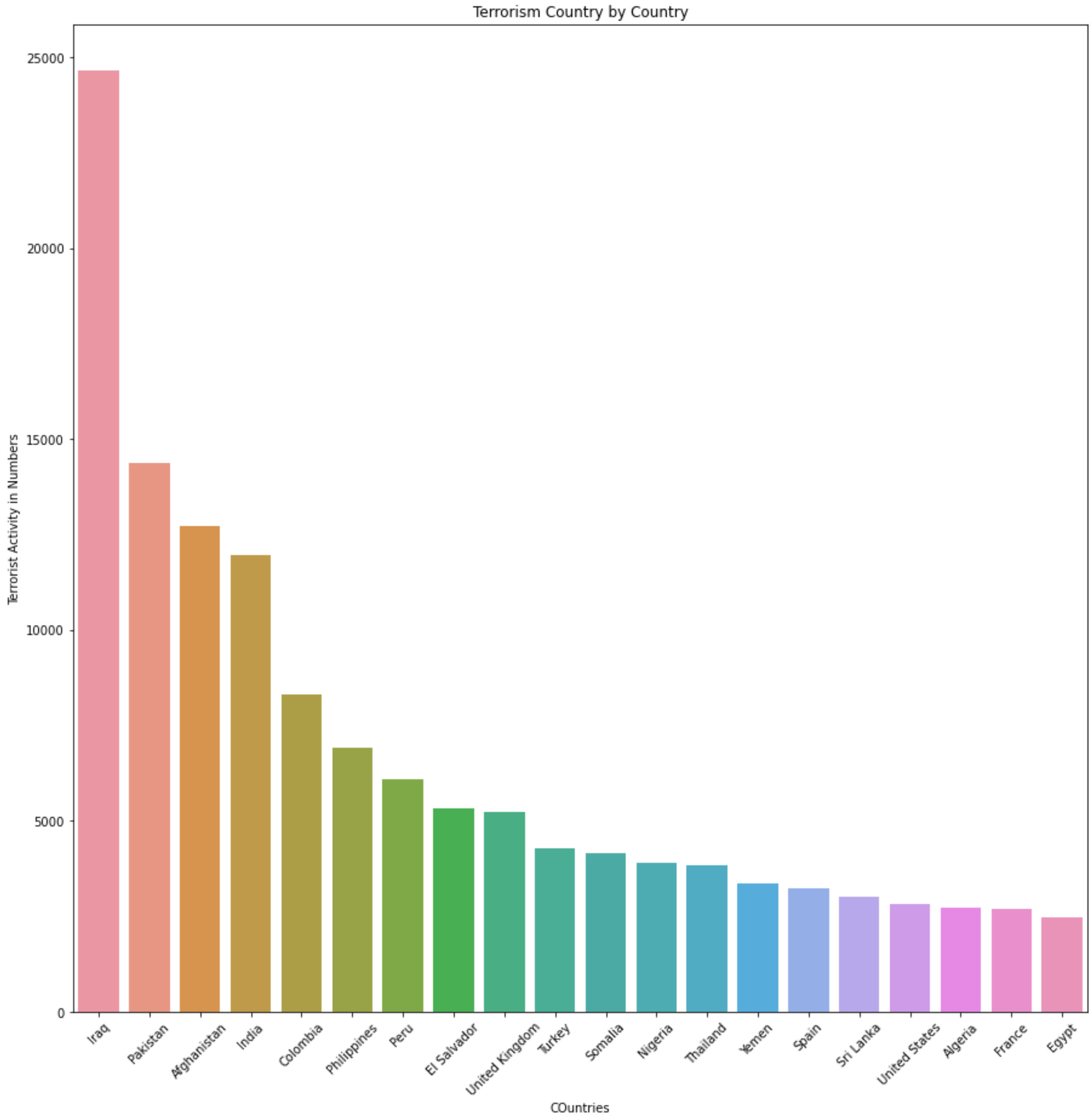
In [13]:
```
# Plotting terrorism year by year
plt.subplots(figsize=(20,10))
sns.countplot(x=data['iyear'])
plt.ylabel('Terrorist Activity in Numbers')
plt.xlabel('Year')
plt.title("Terrorism year by year")
plt.xticks(rotation=90)
plt.show()
```



Conclusion:- Most number of terrorist attacks happened in 2014.

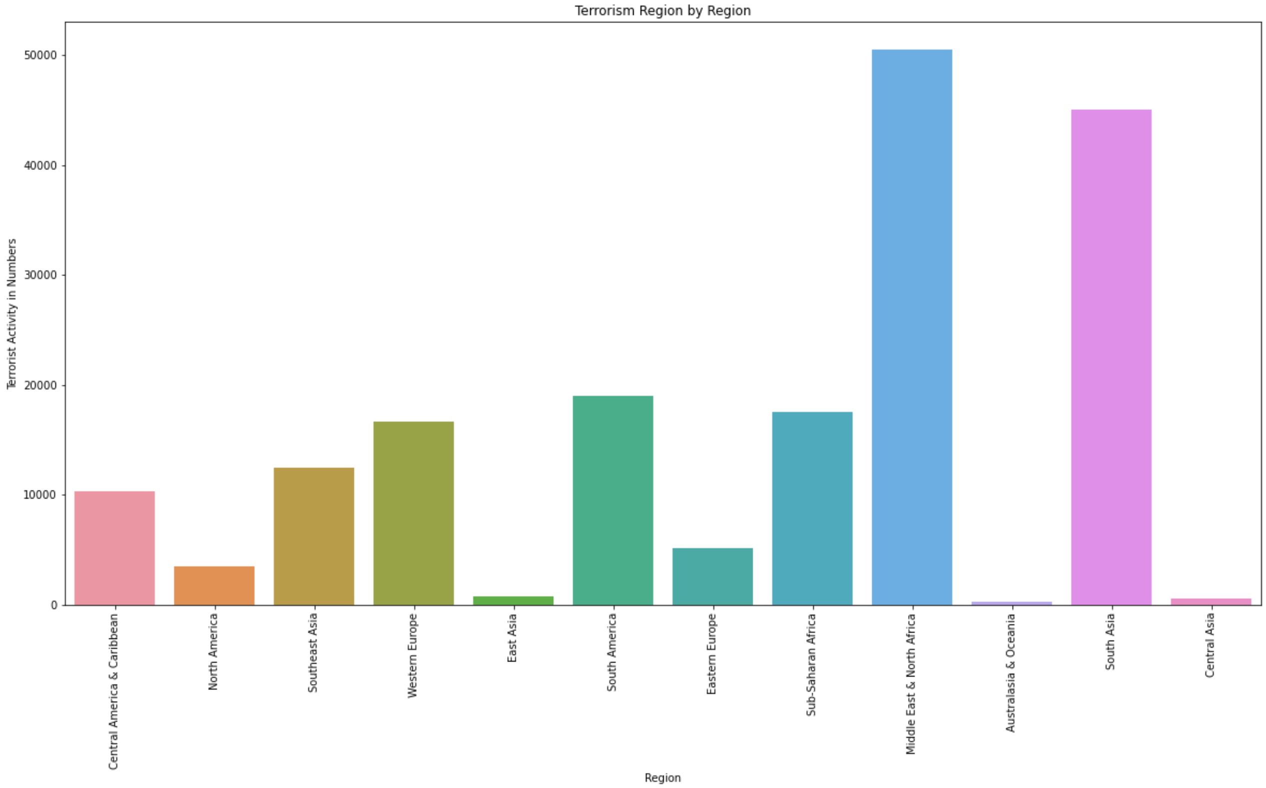
In [14]:

```
# Plotting terrorism country by country
plt.subplots(figsize=(15,15))
sns.barplot(x=data['country_txt'].value_counts()[:20].index,y=data['country_txt'].value_
plt.ylabel('Terrorist Activity in Numbers')
plt.xlabel('COuntries')
plt.title("Terrorism Country by Country")
plt.xticks(rotation=45)
plt.show()
```



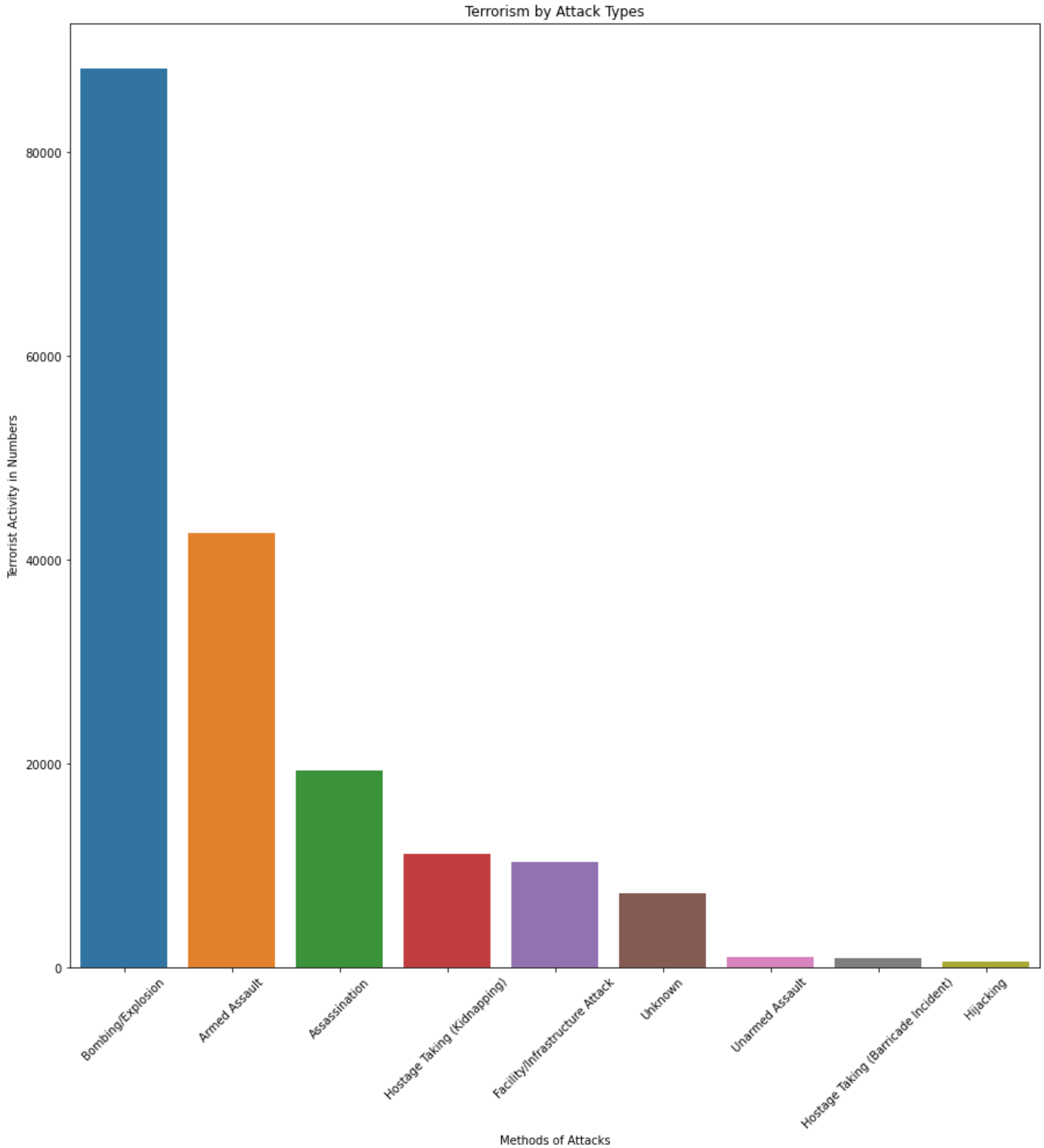
Conclsuion:- Most terrorist activities happend in Iraq till now with more than 23000 attacks in numbers.

In [15]:
```
# Plotting terrorism region by region
plt.subplots(figsize=(20,10))
sns.countplot(x=data['region_txt'])
plt.ylabel('Terrorist Activity in Numbers')
plt.xlabel('Region')
plt.title("Terrorism Region by Region")
plt.xticks(rotation=90)
plt.show()
```



Conclsuion:- Middle East and North Africa has highest number of terrorist activities with nearly 50000 in numbers.

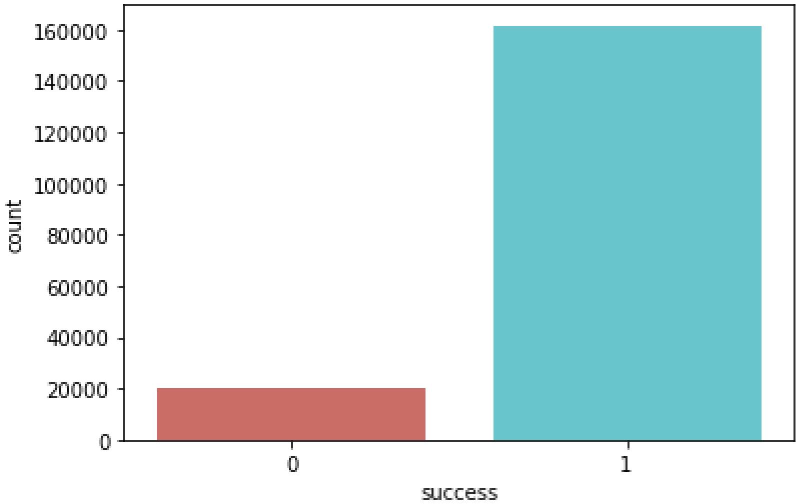
In [16]:
```
# Plotting terrorism by types of terrorist activities
plt.subplots(figsize=(15,15))
sns.barplot(x=data['attacktype1_txt'].value_counts()[:20].index,y=data['attacktype1_txt
plt.ylabel('Terrorist Activity in Numbers')
plt.xlabel('Methods of Attacks')
plt.title("Terrorism by Attack Types")
plt.xticks(rotation=45)
plt.show()
```



Conclsuion:- Bombing/Explosion are the most common methods used for Terrorist Activities with nearly 90000 in numbers.

In [17]:
```
# Successfull Attacks
sns.countplot(x='success', data=data, palette='hls')
```

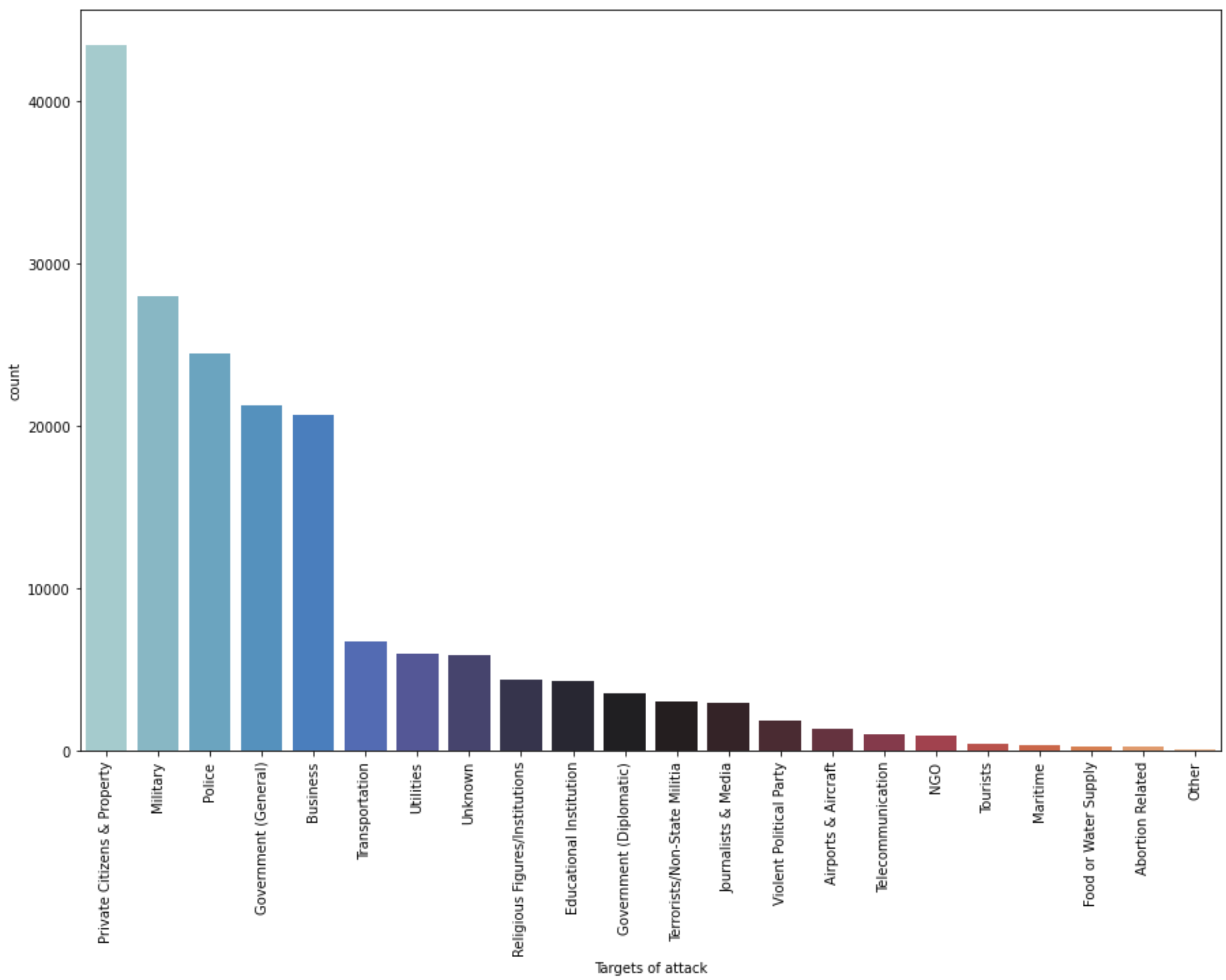
Out[17]: <AxesSubplot:xlabel='success', ylabel='count'>



Conclusion:- Around 160000 terrorist attacks are successfull and around 20000 are unsccessfull till now.

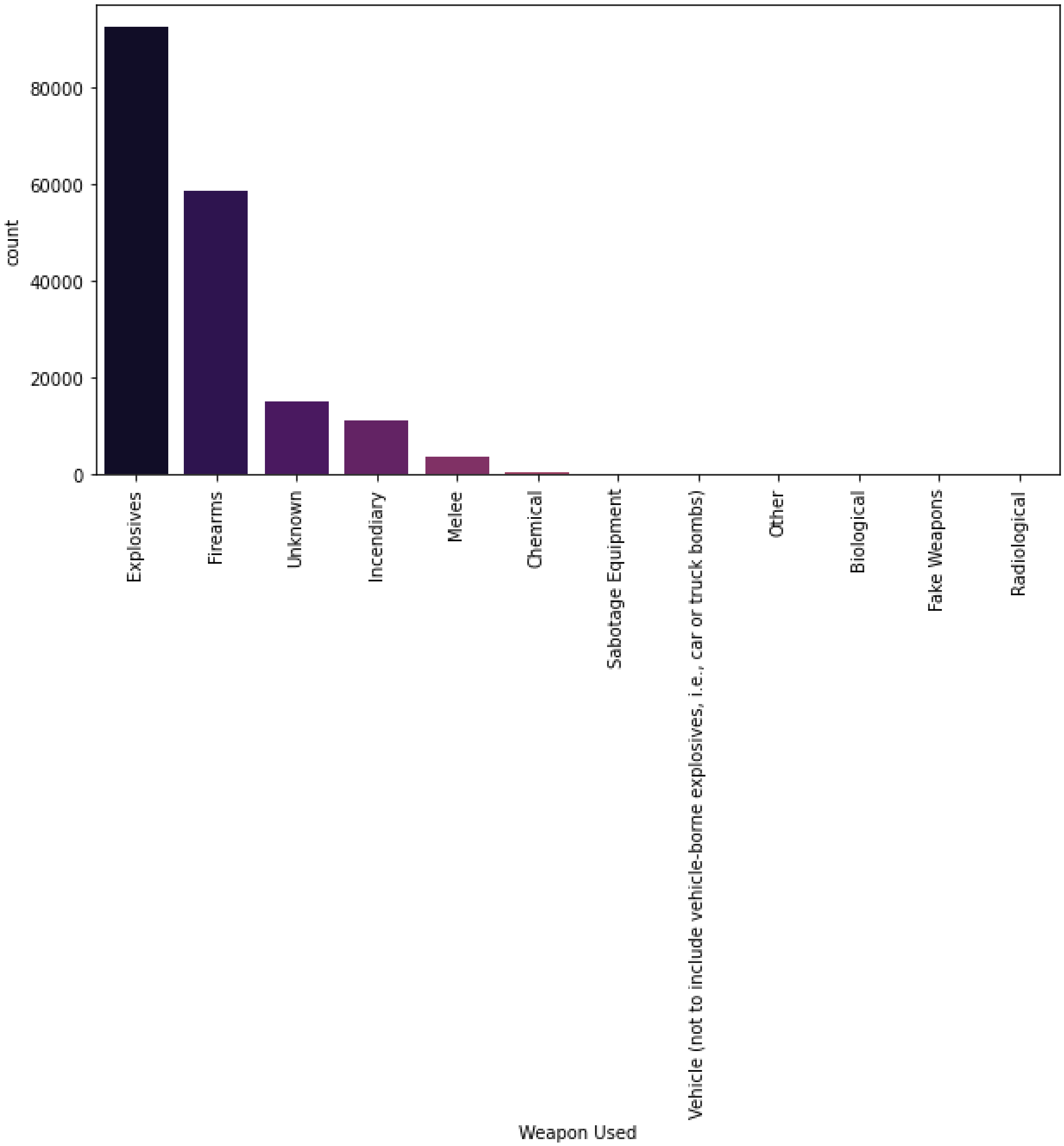
In [18]:

```
# Plotting targets of attack
fig, ax = plt.subplots(figsize=(15,10))
ax = sns.countplot(x = 'targtype1_txt', data=data, palette='icefire', order=data['targt
_ = plt.xlabel('Targets of attack')
_ = plt.setp(ax.get_xticklabels(), rotation = 90)
```



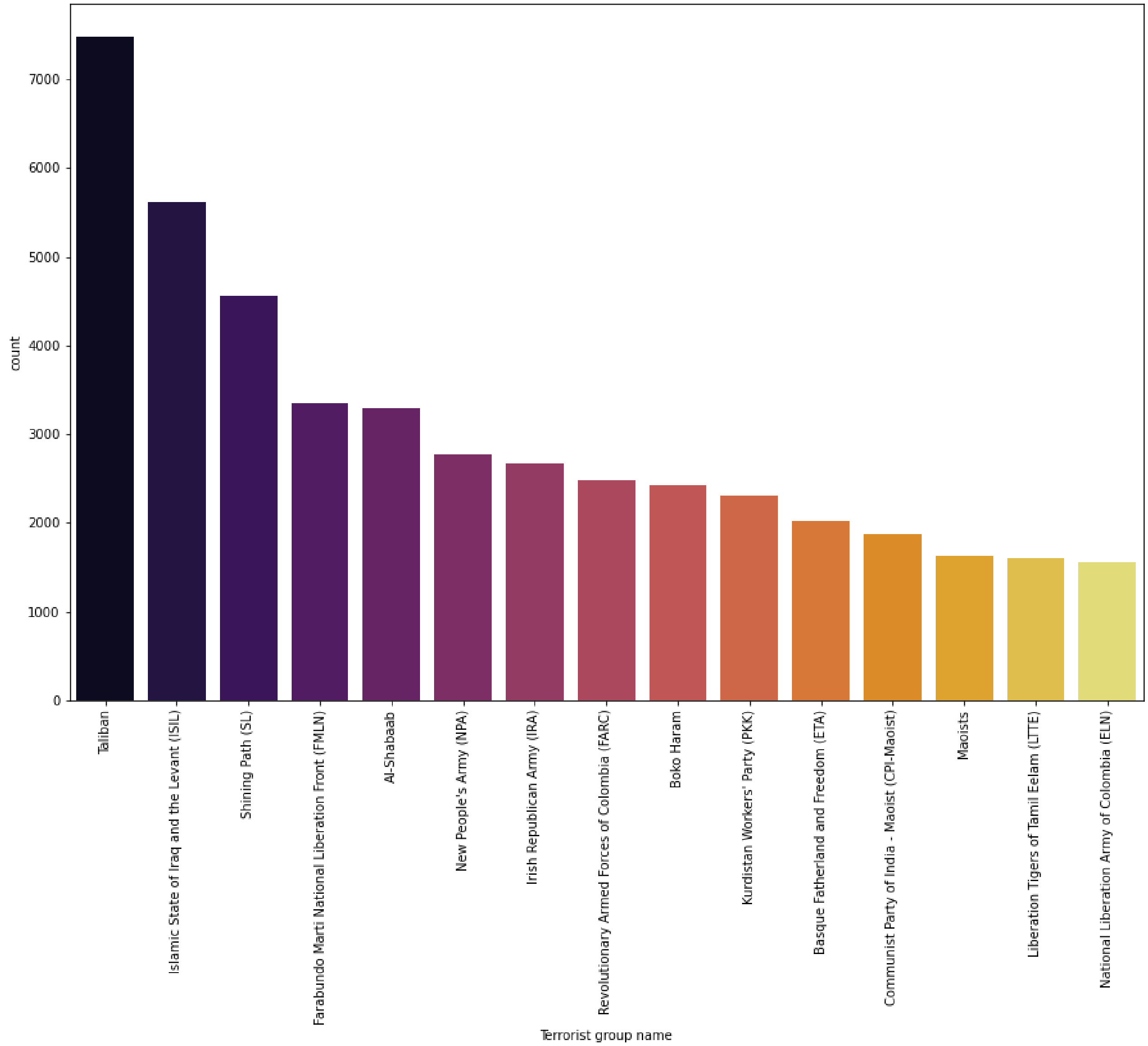
Conclusion:- From the above figure we can see that private citizens and property is the highest target and the top five targets are private citizens and property, military, police, government(general) where people can be found in high numbers and also law & order is affected.

In [19]:
```
# Plotting weapons of attacks
fig, ax = plt.subplots(figsize=(10, 5))
ax = sns.countplot(x='weaptype1_txt', data=data, palette='inferno', order=data['weaptyp
_ = plt.xlabel('Weapon Used')
_ = plt.setp(ax.get_xticklabels(), rotation = 90)
```



Conclusion:- Weapon that is mostly used in terrorist activities is Explosives.

In [20]:
```
fig, ax = plt.subplots(figsize=(15,10))
ax = sns.countplot(x='gname', data=data, palette='inferno', order=data['gname'].value_c
_ = plt.xlabel('Terrorist group name')
_ = plt.setp(ax.get_xticklabels(), rotation=90)
```



Conclusion:- Taliban is the most active terrorist group as the most number of activities has been done by this group.

In [21]:

```
# Summary of our dataset
print("Country with the most attacks:",data['country_txt'].value_counts().idxmax())
print("City with the most attacks:",data['city'].value_counts().index[1])
print("Region with the most attacks:",data['region_txt'].value_counts().idxmax())
print("Country with the most attacks:",data['country_txt'].value_counts().idxmax())
print("Year with the most attacks:",data['iyear'].value_counts().idxmax())
print("Month with the most attacks:",data['imonth'].value_counts().idxmax())
print("Country with the most attacks:",data['country_txt'].value_counts().idxmax())
print("Country with the most attacks:",data['gname'].value_counts().index[1])
print("Most attacks types:",data['attacktype1_txt'].value_counts().idxmax())
```

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

In [35]:

```
from scipy import signal
from wordcloud import WordCloud
plt.subplots(figsize=(15,20))
wordcloud=WordCloud(background_color='Black').generate(''.join(data['provstate'].dropna
plt.axis('on')
plt.imshow(wordcloud)
plt.show()
```