THE SPARKS FOUNDATION

DATA SCIENCE AND BUSINESS ANALYTICS INTERNSHIP

AUTHOR: MEGHA

TASK 5: Exploratory Data Analysis - Terrorism

In [1]:
```
#importing required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [6]:
```
t = pd.read_csv('Global Terrorism.csv', encoding= 'latin1' , low_memory=False)
t.head()
```

Out[6]:

	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	country_txt	region	...	addnotes	scite1	scite2	scite3	dbsource
0	1.970000e+11	1970	7	2	NaN	0	NaN	58	Dominican Republic	2	...	NaN	NaN	NaN	NaN	PGIS
1	1.970000e+11	1970	0	0	NaN	0	NaN	130	Mexico	1	...	NaN	NaN	NaN	NaN	PGIS
2	1.970010e+11	1970	1	0	NaN	0	NaN	160	Philippines	5	...	NaN	NaN	NaN	NaN	PGIS
3	1.970010e+11	1970	1	0	NaN	0	NaN	78	Greece	8	...	NaN	NaN	NaN	NaN	PGIS
4	1.970010e+11	1970	1	0	NaN	0	NaN	101	Japan	4	...	NaN	NaN	NaN	NaN	PGIS

5 rows × 135 columns

In [7]:
```
t.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 181691 entries, 0 to 181690
Columns: 135 entries, eventid to related
```

```
dtypes: float64(56), int64(21), object(58)
memory usage: 187.1+ MB
```

In [9]:
```
t.nunique()
```

Out[9]:
```
eventid         576
iyear            47
imonth           13
iday             32
approxdate     2244
               ...
INT_LOG           3
INT_IDEO          3
INT_MISC          3
INT_ANY           3
related       14306
Length: 135, dtype: int64
```

In [10]:
```
t.describe()
```

Out[10]:

	eventid	iyear	imonth	iday	extended	country	region	latitude	longitude	
count	1.816910e+05	181691.000000	181691.000000	181691.000000	181691.000000	181691.000000	181691.000000	177135.000000	1.771340e+05	18
mean	2.002704e+11	2002.638997	6.467277	15.505644	0.045346	131.968501	7.160938	23.498343	-4.586957e+02	
std	1.325955e+09	13.259430	3.388303	8.814045	0.208063	112.414535	2.933408	18.569242	2.047790e+05	
min	1.970000e+11	1970.000000	0.000000	0.000000	0.000000	4.000000	1.000000	-53.154613	-8.618590e+07	
25%	1.991020e+11	1991.000000	4.000000	8.000000	0.000000	78.000000	5.000000	11.510046	4.545640e+00	
50%	2.009020e+11	2009.000000	6.000000	15.000000	0.000000	98.000000	6.000000	31.467463	4.324651e+01	
75%	2.014080e+11	2014.000000	9.000000	23.000000	0.000000	160.000000	10.000000	34.685087	6.871033e+01	
max	2.017120e+11	2017.000000	12.000000	31.000000	1.000000	1004.000000	12.000000	74.633553	1.793667e+02	

8 rows × 77 columns

In [11]:
```
t.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 [12]:
```
t.head()
```

Out[12]:

	eventid	Year	Month	Day	approxdate	extended	resolution	country	Country	region	...	addnotes	scite1	scite2	scite3	dbsource
0	1.970000e+11	1970	7	2	NaN	0	NaN	58	Dominican Republic	2	...	NaN	NaN	NaN	NaN	PGIS
1	1.970000e+11	1970	0	0	NaN	0	NaN	130	Mexico	1	...	NaN	NaN	NaN	NaN	PGIS
2	1.970010e+11	1970	1	0	NaN	0	NaN	160	Philippines	5	...	NaN	NaN	NaN	NaN	PGIS
3	1.970010e+11	1970	1	0	NaN	0	NaN	78	Greece	8	...	NaN	NaN	NaN	NaN	PGIS
4	1.970010e+11	1970	1	0	NaN	0	NaN	101	Japan	4	...	NaN	NaN	NaN	NaN	PGIS

5 rows × 135 columns

- We can see that we have around 135 coulumns.
- We aim at finding the hot zones so we can drop outher columns.
- dropping other columns will be a tedious task, so it is better to create a new variable wherein we will take only important coumns.

In [13]:
```
trr=t[['Year','Month','Day','Country','state','Region','city','latitude','longitude','AttackType','Killed',
                'Wounded','Target','Summary','Group','Target_type','Weapon_type','Motive']]
trr.head()
```

Out[13]:

	Year	Month	Day	Country	state	Region	city	latitude	longitude	AttackType	Killed	Wounded	Target	Summ
0	1970	7	2	Dominican Republic	NaN	Central America & Caribbean	Santo Domingo	18.456792	-69.951164	Assassination	1.0	0.0	Julio Guzman	N
1	1970	0	0	Mexico	Federal	North America	Mexico city	19.371887	-99.086624	Hostage Taking (Kidnapping)	0.0	0.0	Nadine Chaval, daughter	N

	Year	Month	Day	Country	state	Region	city	latitude	longitude	AttackType	Killed	Wounded	Target	Summ
2	1970	1	0	Philippines	Tarlac	Southeast Asia	Unknown	15.478598	120.599741	Assassination	1.0	0.0	Employee	N
3	1970	1	0	Greece	Attica	Western Europe	Athens	37.997490	23.762728	Bombing/Explosion	NaN	NaN	U.S. Embassy	N
4	1970	1	0	Japan	Fukouka	East Asia	Fukouka	33.580412	130.396361	Facility/Infrastructure Attack	NaN	NaN	U.S. Consulate	N

In [14]:
```
trr.shape
```

Out[14]:
```
(181691, 18)
```

In [15]:
```
trr.isnull().sum()
```

Out[15]:
```
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 [16]:
```
trr.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
```

- **We need to make sure if there is any value in day or month as 0.**
- **We have too many missing values in some columns, before handling them we need to know their percentage**

In [17]:
```
count=0
for i in trr['Day']:
    if i ==0:
        count=count+1
    else:
        count=count
print('Number of Days entered as 0: ',count)
```

```
Number of Days entered as 0:  891
```

In [18]:
```
cnt=0
for i in trr['Month']:
    if i ==0:
        cnt=cnt+1
    else:
        cnt=cnt
print('Number of Months entered as 0: ',cnt)
```

```
Number of Months entered as 0:  20
```

We are assigning random dates and months who have their values as 0, so that we don't loose the information of other columns associated with that row.¶

In [20]:
```
trr['Day'] = trr['Day'].apply(lambda x: np.random.randint(1,32) if x == 0 else x)
trr['Month'] = trr['Month'].apply(lambda x: np.random.randint(1,13) if x == 0 else x)
```

```
<ipython-input-20-eebef340eee1>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
-a-view-versus-a-copy
  trr['Day'] = trr['Day'].apply(lambda x: np.random.randint(1,32) if x == 0 else x)
<ipython-input-20-eebef340eee1>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
-a-view-versus-a-copy
  trr['Month'] = trr['Month'].apply(lambda x: np.random.randint(1,13) if x == 0 else x)
```

In [21]:
```
#checking for days
count=0
for i in trr['Day']:
    if i ==0:
        count=count+1
    else:
        count=count
print('Number of Days entered as 0: ',count)
```

```
Number of Days entered as 0:  0
```

In [22]:
```
#checking for months
cnt=0
for i in trr['Month']:
    if i ==0:
        cnt=cnt+1
    else:
        cnt=cnt
print('Number of Months entered as 0: ',cnt)
```

```
Number of Months entered as 0:  0
```

Finding missing value percentage

In [23]:
```
def null_val_(trr):
    null_val = trr.isnull().sum()
    null_val_p = 100 * trr.isnull().sum()/len(trr)
    null_val_ = pd.concat([null_val, null_val_p], axis=1)
    null_val_last = null_val_.rename(
    columns = {0 : 'Null Values', 1 : 'Percentage '})
    return null_val_last
null_val_(trr)
```

Out[23]:

	Null Values	Percentage
Year	0	0.000000
Month	0	0.000000
Day	0	0.000000
Country	0	0.000000
state	421	0.231712
Region	0	0.000000
city	434	0.238867
latitude	4556	2.507554
longitude	4557	2.508104
AttackType	0	0.000000
Killed	10313	5.676120
Wounded	16311	8.977330
Target	636	0.350045
Summary	66129	36.396409
Group	0	0.000000
Target_type	0	0.000000
Weapon_type	0	0.000000

	Null Values	Percentage
Motive	131130	72.171984

In [24]:
```
trr['Motive'].fillna(value='NA', inplace=True)
trr['Summary'].fillna(value='NA', inplace=True)
trr['city'].fillna(value='NA', inplace=True)
trr['Target'].fillna(value='NA', inplace=True)
trr["Killed"].fillna(trr["Killed"].mean(), inplace=True)
trr["Wounded"].fillna(trr["Wounded"].mean(), inplace=True)
trr["latitude"].fillna(trr["latitude"].mean(), inplace=True)
trr["longitude"].fillna(trr["longitude"].mean(), inplace=True)
trr['state'].fillna(value='NA', inplace=True)
```

```
C:\Users\ASUS\anaconda3\anaconda\lib\site-packages\pandas\core\series.py:4517: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning
-a-view-versus-a-copy
  return super().fillna(
```

- We saw that some Coloumns had High missing values, we needed to handle them.
- All the coloumns with 'object' datatype have been replace by NA for the missing values so that we don't loose information associated with them.
- Similarly we have replaced all columns with float datatype with their respective mean.

In [25]:
```
trr.isnull().sum()
```

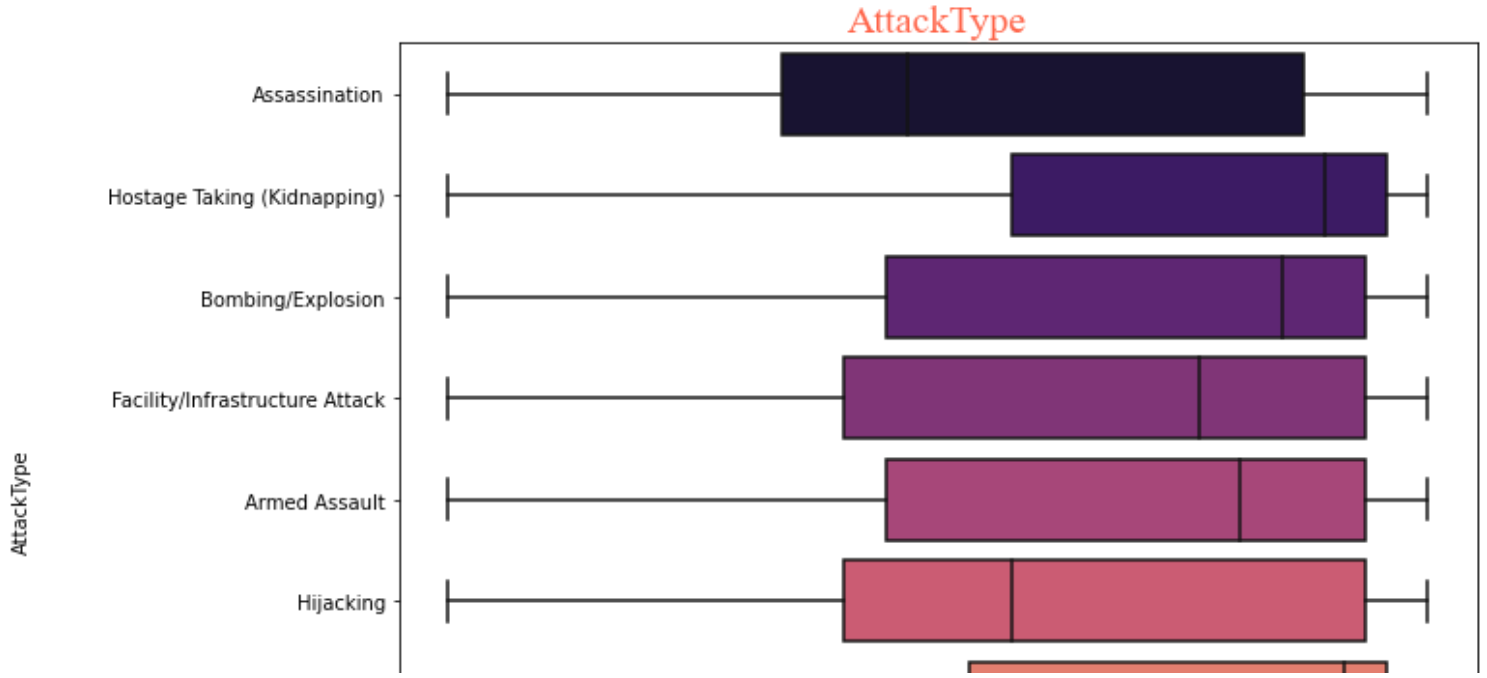
Out[25]:
```
Year           0
Month          0
Day            0
Country        0
state          0
Region         0
city           0
latitude       0
longitude      0
AttackType     0
Killed         0
Wounded        0
```

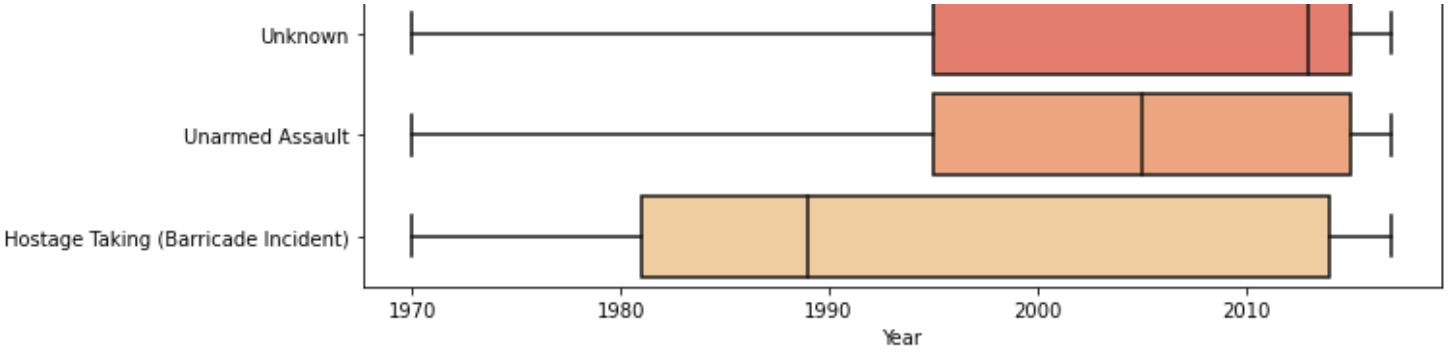
```
Target          0
Summary         0
Group           0
Target_type     0
Weapon_type     0
Motive          0
dtype: int64
```

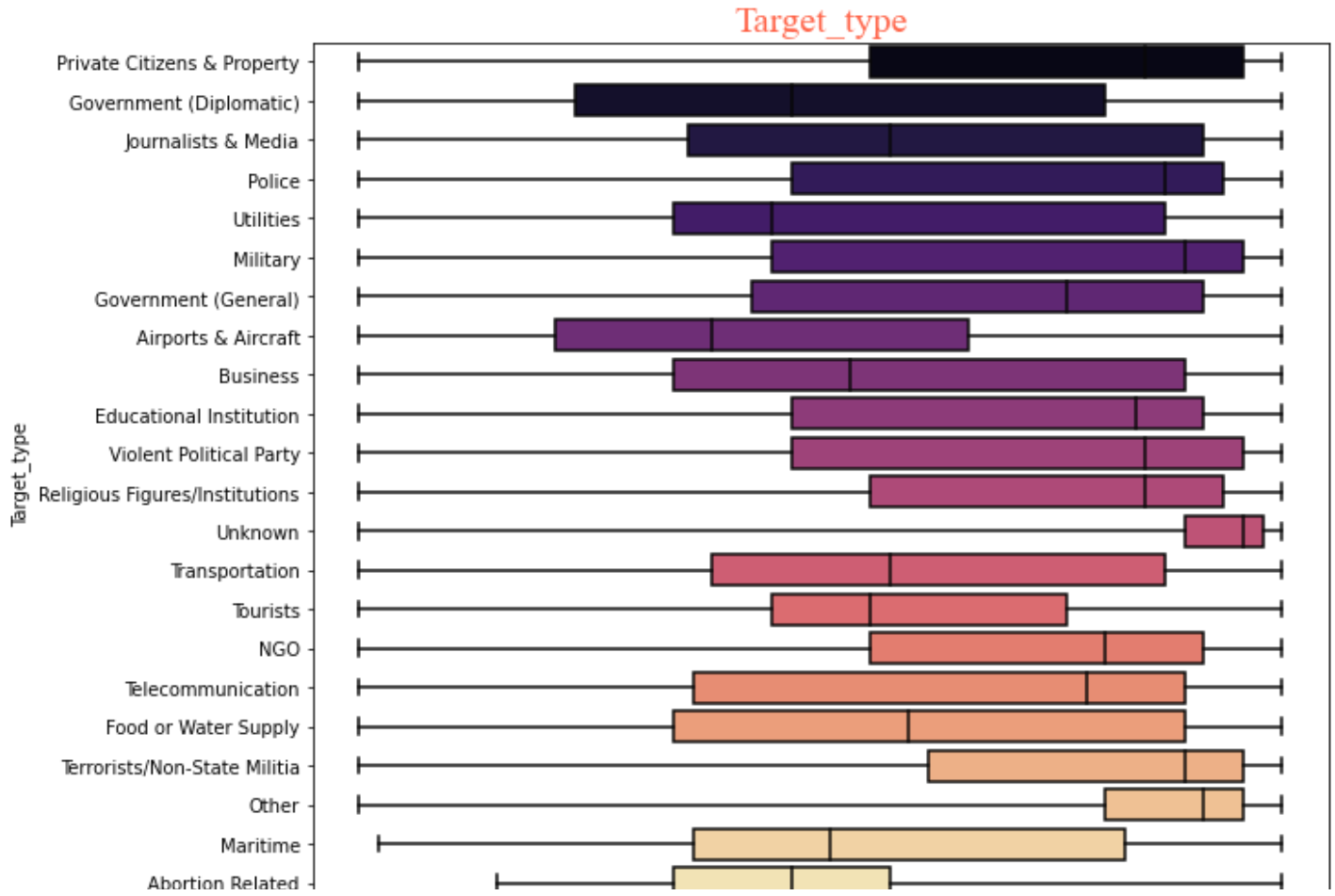
We have successfully handeled our missing values!

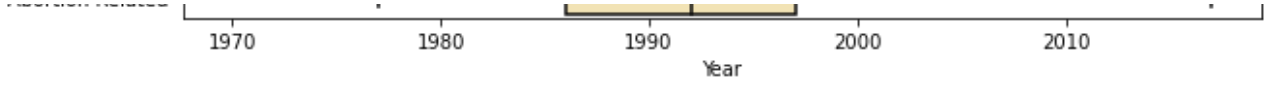
Plotting Boxplots

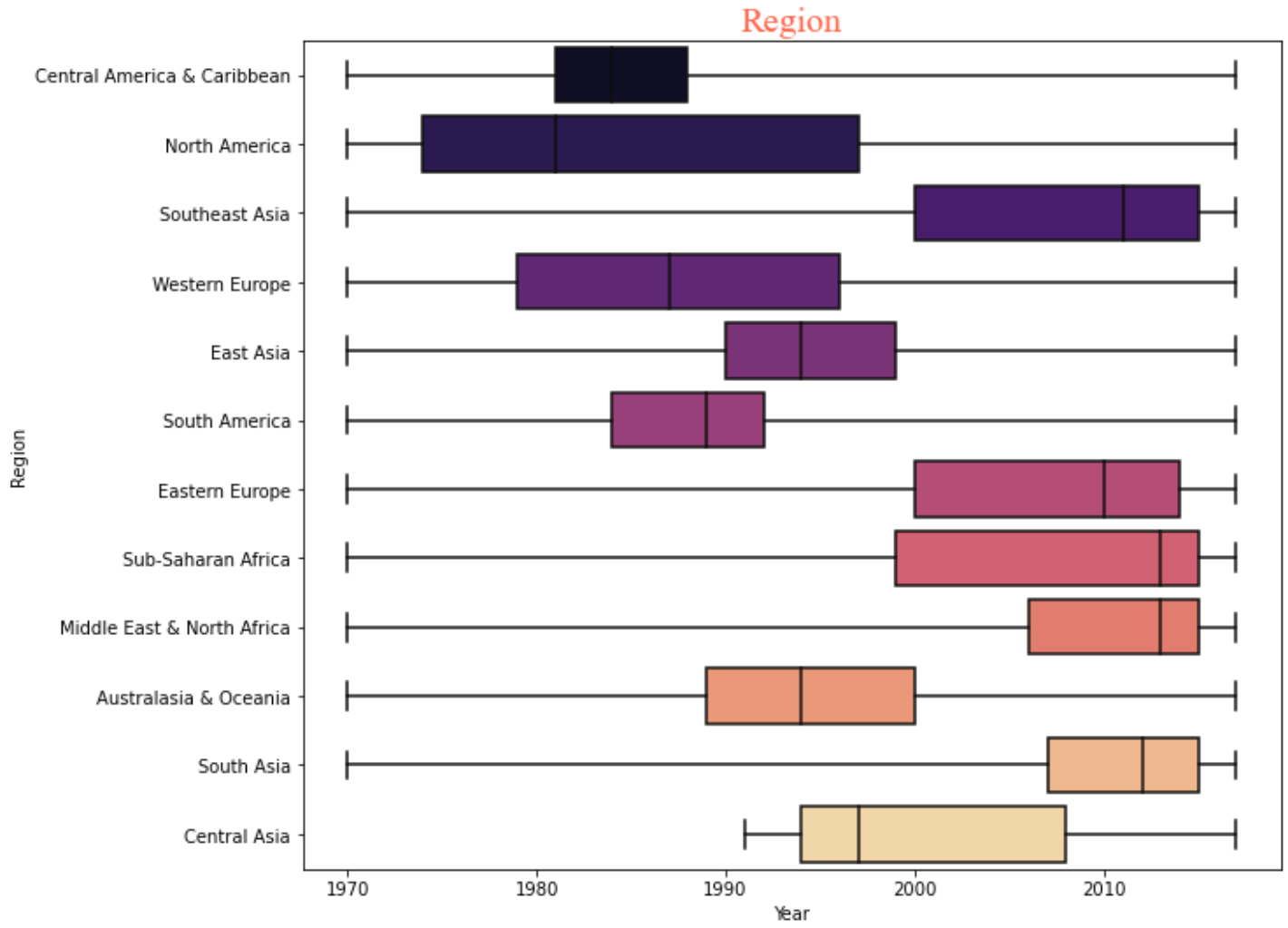
In [26]:
```
title_style = {'family': 'Times New Roman', 'color': 'Tomato', 'size': 20 }
dict = {'AttackType':1,'Target_type':2,  'Region':3, 'Weapon_type':4 }
plt.figure(figsize=(10,40))

for value, i in dict.items():
    plt.subplot(4,1,i)
    sns.boxplot(x="Year", y=value, data=trr, whis=[0, 100], palette="magma")
    plt.title(value , fontdict = title_style)
plt.show()
```

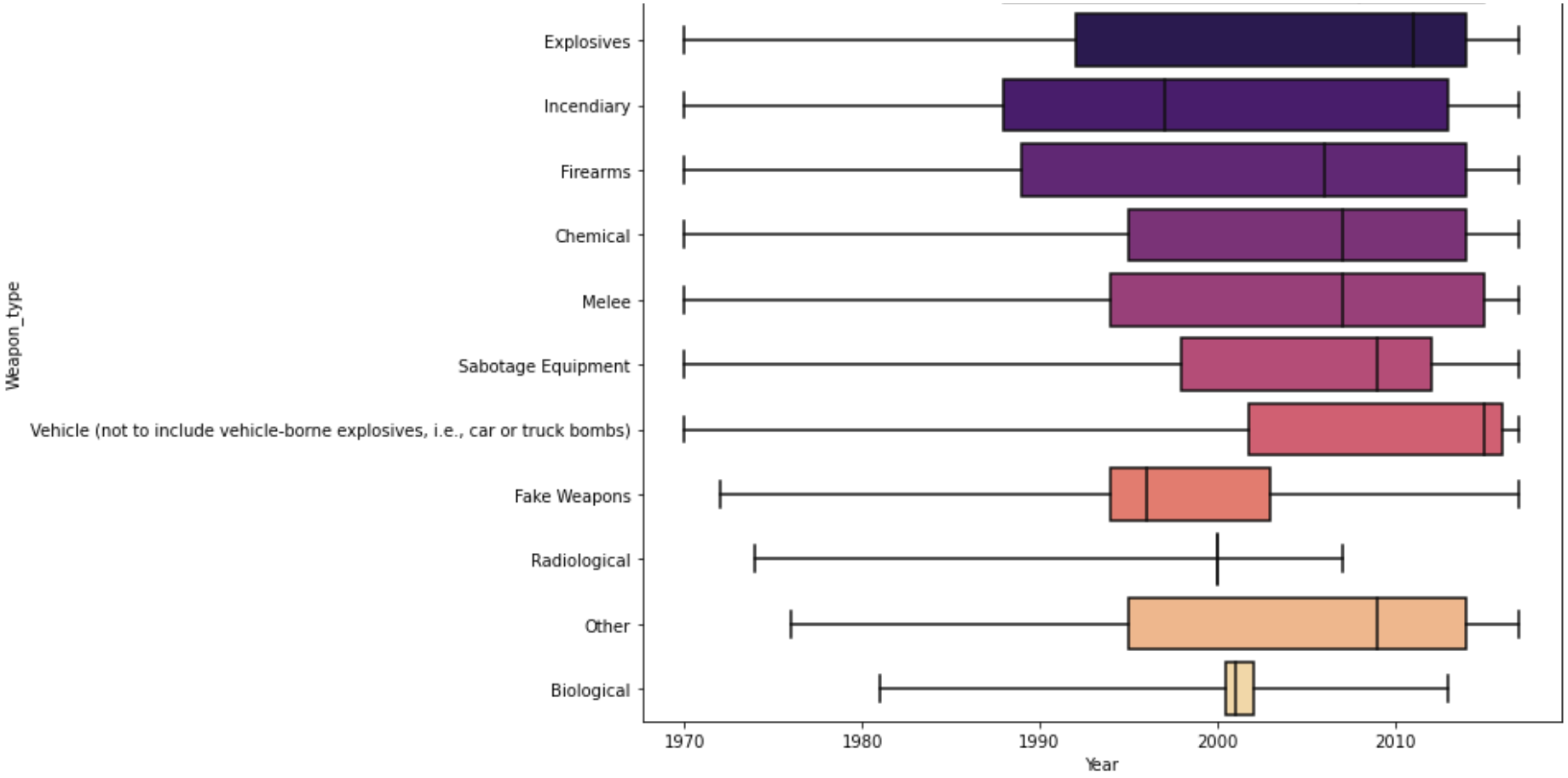






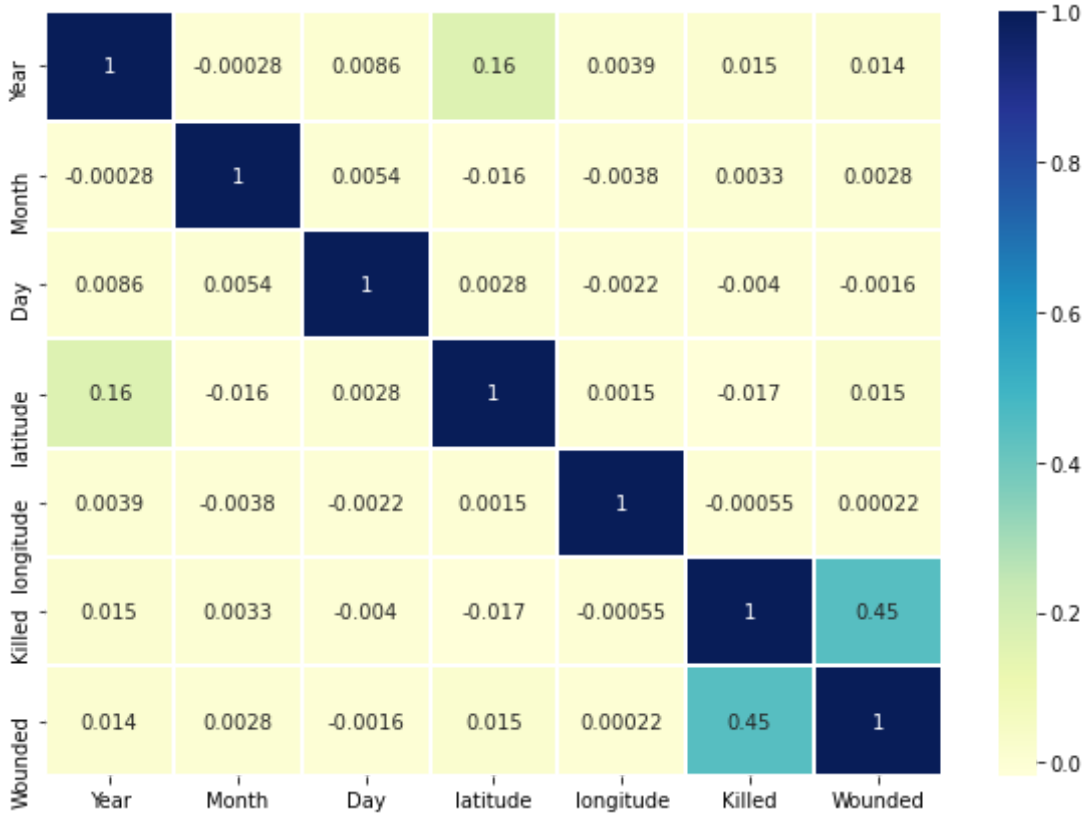






In [27]:
```
plt.figure(figsize = (10,7))
sns.heatmap(trr.corr(), cmap="YlGnBu", annot=True, xticklabels='auto', yticklabels='auto', linewidth=1 )
```

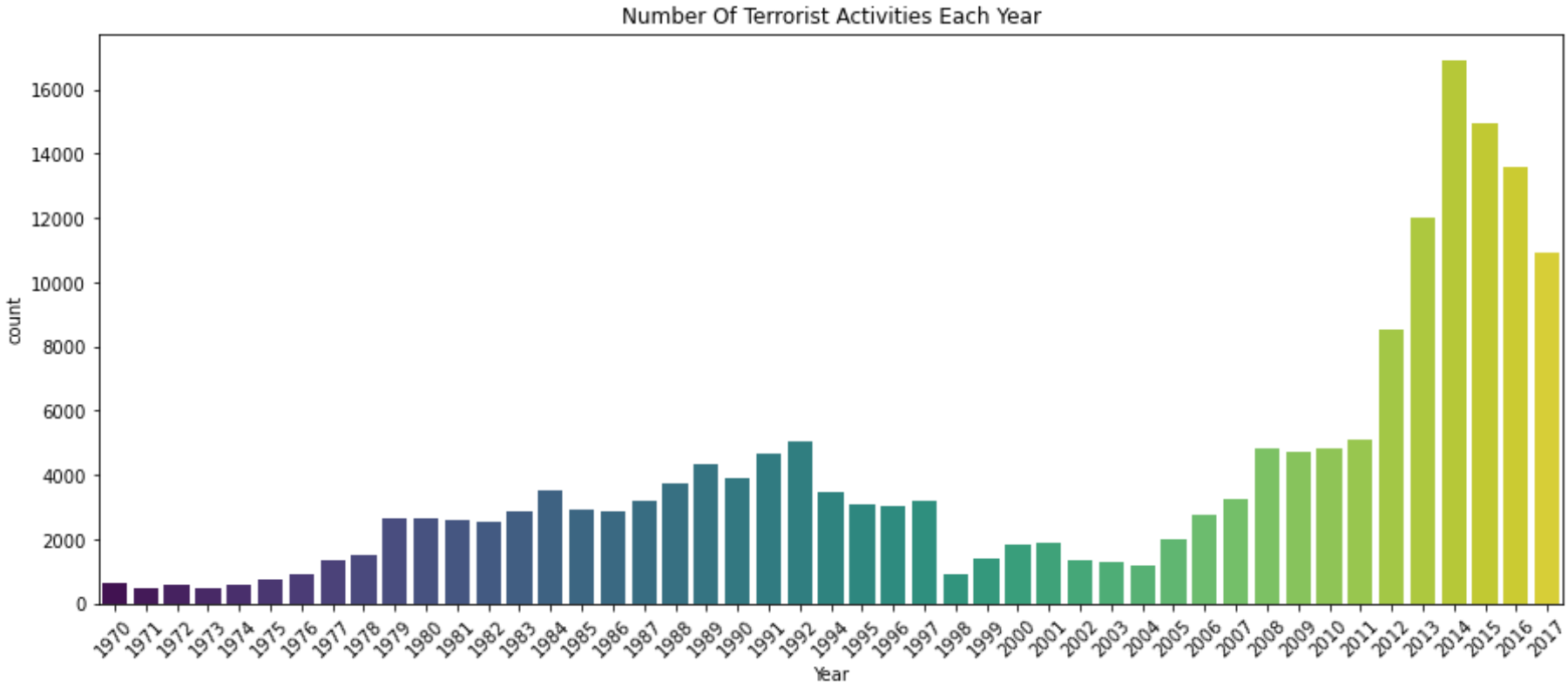
Out[27]: <AxesSubplot:>



Terrorist Activities each year

In [28]:
```
plt.subplots(figsize=(15,6))
sns.countplot('Year',data=trr,palette="viridis")
plt.xticks(rotation=45)
plt.title('Number Of Terrorist Activities Each Year')
plt.show()
```

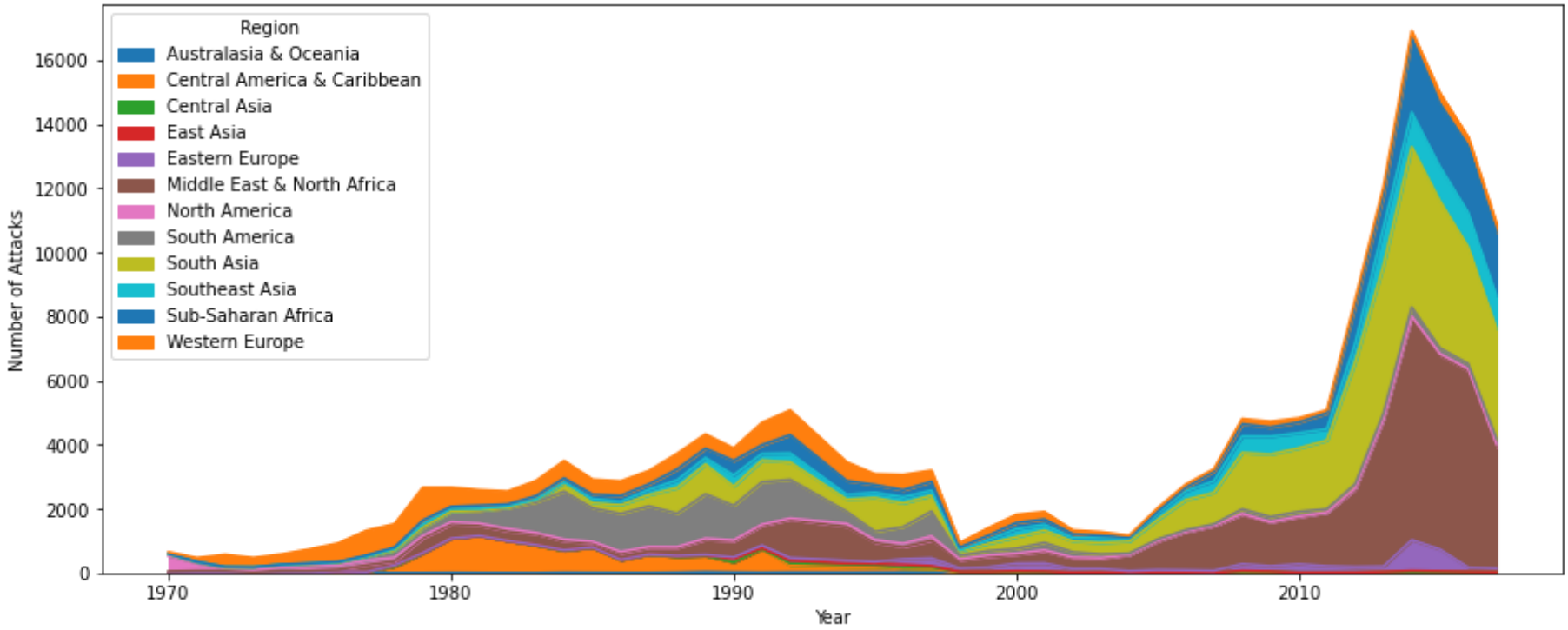
```
C:\Users\ASUS\anaconda3\anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



- **There has been a gradual increase in Terror Activities since 2004.**
- **Highest number of terror activities occurred in the year 2014.**
- **After 2014 the terror activities started to decrease.**

Terrorist Activities in each Year w.r.t region

In [29]:
```
pd.crosstab(trr.Year, trr.Region).plot(kind='area',figsize=(15,6))
plt.ylabel('Number of Attacks')
plt.show()
```



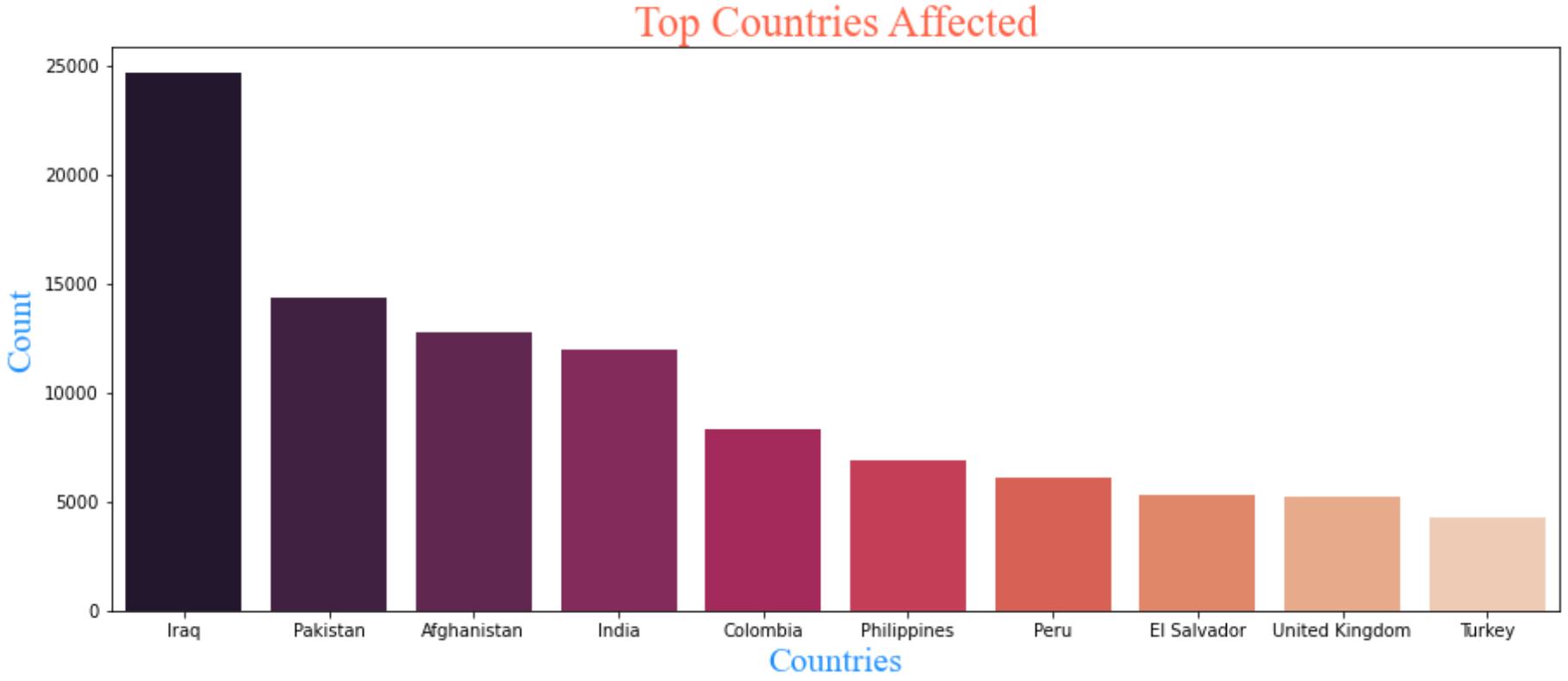
- **Maximum Attacks have been in Central America, followed by Australasia in the year 2010.**
- **There have been very less terror casualties in: East Asia, North America.**
- **In East Asia the number of attack gradually reduced to 0 after the year 1990.**

- **The Worst Attack took place in the US in the year 2001 with a total casualty of 9574.**
- **After the 9/11 attacks the security measures in the US were escalated in such a way that no attacks took place until the year 2017.**
- **Kenya had the 2nd worst attack in the 1998 with 4224 Casualties, after that proper security measures were taken to avoid any such incident.**
- **Russia had its worst attack in the year 2004 with 1071 casualties, since then there has been no major attack.**

Top 10 countries to be affected with Terror Attacks

In [31]:
```
plt.subplots(figsize=(15,6))
style1  = {'family': 'Times New Roman', 'color': 'Tomato', 'size': 25}
style2  = {'family': 'Times New Roman', 'color': 'DodgerBlue', 'size': 20}
sns.barplot(trr['Country'].value_counts()[:10].index, trr['Country'].value_counts()[:10].values,palette='rocket')
plt.title('Top Countries Affected', fontdict=style1 )
plt.xlabel('Countries' , fontdict=style2 )
plt.ylabel('Count', fontdict=style2 )
#plt.xticks(rotation= 90)
plt.show()
```

```
C:\Users\ASUS\anaconda3\anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



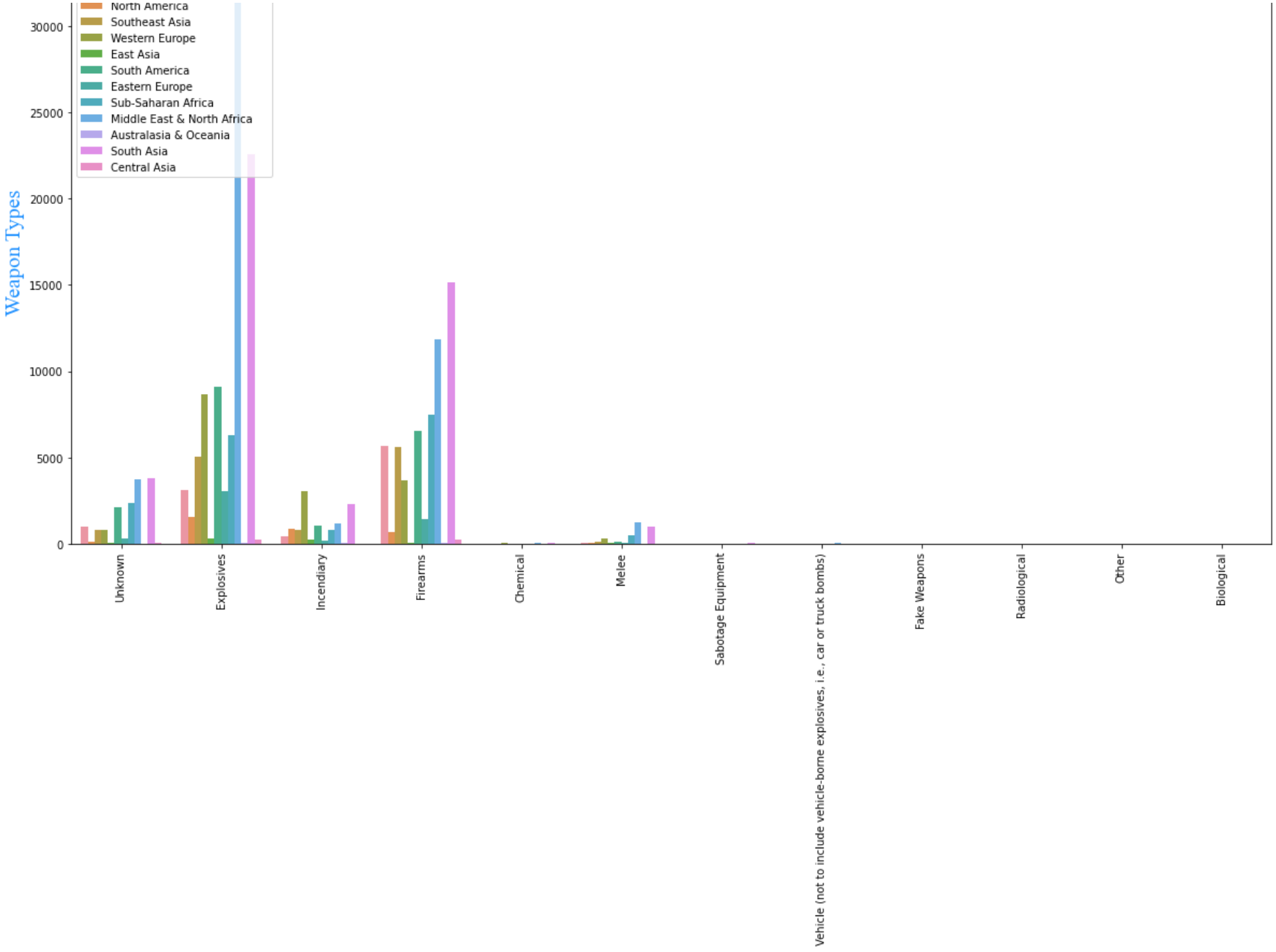
In [32]:
```
weapon_cross = pd.crosstab(trr["Weapon_type"], trr["Region"])
weapon_cross
```

Out[32]:

Region	Australasia & Oceania	Central America & Caribbean	Central Asia	East Asia	Eastern Europe	Middle East & North Africa	North America	South America	South Asia	Southeast Asia	Sub-Saharan Africa	Western Europe
Weapon_type												
Biological	0	0	0	2	0	1	24	1	2	0	3	2
Chemical	11	2	2	17	12	73	26	26	83	11	12	46
Explosives	80	3149	254	333	3089	32283	1557	9098	22568	5039	6319	8657
Fake Weapons	0	0	1	4	4	6	5	3	3	0	1	6
Firearms	74	5679	232	41	1461	11877	682	6525	15169	5634	7499	3651
Incendiary	74	435	15	252	186	1181	897	1077	2285	837	840	3056
Melee	10	65	14	82	90	1227	74	131	998	147	478	339
Other	1	0	0	3	4	23	18	6	16	4	19	20
Radiological	0	0	0	10	0	0	1	0	1	0	0	2
Sabotage Equipment	0	5	0	3	4	10	19	15	46	21	7	11
Unknown	31	1005	45	47	293	3724	138	2093	3788	792	2371	830
Vehicle (not to include vehicle-borne explosives, i.e., car or truck bombs)	1	4	0	8	1	69	15	3	15	0	1	19

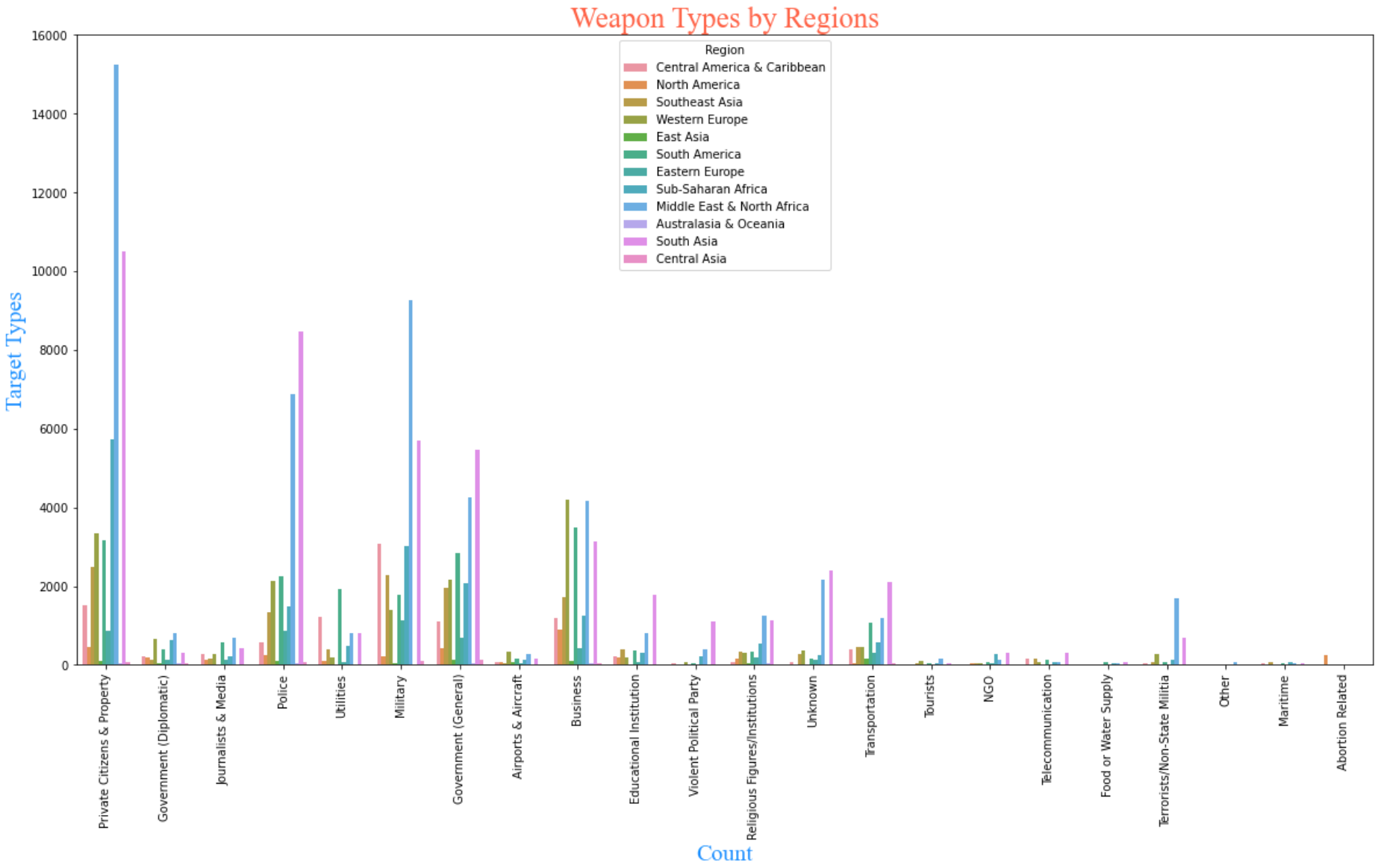
In [33]:

```
plt.figure(figsize=(20,10))
sns.countplot(x="Weapon_type", hue="Region", data=trr)
style1  = {'family': 'Times New Roman', 'color': 'Tomato', 'size': 25}
style2  = {'family': 'Times New Roman', 'color': 'DodgerBlue', 'size': 20}
plt.title("Weapon Types by Regions", fontdict = style1)
plt.ylabel("Weapon Types", fontdict = style2 )
plt.xlabel("Count", fontdict = style2)
plt.xticks(rotation=90)
plt.show()
```



Count

In [34]:
```
plt.figure(figsize=(20,10))
sns.countplot(x="Target_type", hue="Region", data=trr)
style1  = {'family': 'Times New Roman', 'color': 'Tomato', 'size': 25}
style2  = {'family': 'Times New Roman', 'color': 'DodgerBlue', 'size': 20}
plt.title("Weapon Types by Regions", fontdict = style1)
plt.ylabel("Target Types", fontdict = style2 )
plt.xlabel("Count", fontdict = style2)
plt.xticks(rotation=90)
plt.show()
```



- **In middle East region most attacks have been carried out on Private citizens followed by military.**
- **Most of the transportation and violent political party attacks have been witnessed by the South Asian Countries.**

- **The South Asian countries need to tighten up security measures in the police, military, Government, Business, Educational, and religious sectors.**

Evaluating

In [35]:
```
print("Country with the most attacks:",trr['Country'].value_counts().idxmax())
print("City with the most attacks:",trr['city'].value_counts().index[1])
print("Region with the most attacks:",trr['Region'].value_counts().idxmax())
print("Year with the most attacks:",trr['Year'].value_counts().idxmax())
print("Month with the most attacks:",trr['Month'].value_counts().idxmax())
print("Group with the most attacks:",trr['Group'].value_counts().index[1])
print("Most Attack Types:",trr['AttackType'].value_counts().idxmax())
print("Most Target Types:",trr['Target_type'].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
Most Target Types: Private Citizens & Property
```

CONCLUSION

- **There has been a gradual increase in Terror Activities since 2004.**
- **Highest number of terror activities occurred in the year 2014.**
- **After 2014 the terror activities started to decrease.**
- **Maximum Attacks have been in Central America, followed by Australasia in the year 2010.**
- **There have been very less terror casualties in: East Asia, North America.**
- **In East Asia the number of attacks gradually reduced to 0 after the year 1990.**
- **The Worst Attack took place in the US in the year 2001 with a total casualty of 9574.**
- **After the 9/11 attack the security measures in the US were escalated in such a way that no attacks took place until the year 2017.**
- **Kenya had the 2nd worst attack in the 1998 with 4224 Casualties, after that proper security measures were taken to avoid any such incident.**
- **Russia had its worst attack in the year 2004 with 1071 casualties, since then there has been no major attack.**

- **Mostly there is usage of Explosives and Firearms as Weapons.**
- **In middle East region most attacks have been carried out on Private citizens followed by military.**
- **Most of the transportation and violent political party attacks have been witnessed by the South Asian Countries.**
- **The South Asian countries need to tighten up security measures in the police, military, Government, Business, Educational, and religious sectors.**