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

```
from google.colab import drive
drive.mount('/content/drive/')
```

```
Drive already mounted at /content/drive/; to attempt to forcibly remount, call drive.mount("/content/drive/", force_remount=Tr
```

```
path = '/content/drive/MyDrive'
```

```
import pandas as pd

# Update the path based on your Google Drive structure
path = '/content/drive/MyDrive/Colab Notebooks/'

data = pd.read_csv(path + 'Global Terrrism Dataset .csv', encoding='Latin-1')
data.head()
```

```
<ipython-input-4-444f6c77698e>:6: DtypeWarning: Columns (4,6,31,33,61,62,63,76,79,
  data = pd.read_csv(path + 'Global Terrrism Dataset .csv', encoding='Latin-1')
```

	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	co
0	197000000001	1970	7	2	NaN	0	NaN	58	
1	197000000002	1970	0	0	NaN	0	NaN	130	
2	197001000001	1970	1	0	NaN	0	NaN	160	
3	197001000002	1970	1	0	NaN	0	NaN	78	
4	197001000003	1970	1	0	NaN	0	NaN	101	

5 rows × 135 columns

```
data.head()
```

	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	co
0	197000000001	1970	7	2	NaN	0	NaN	58	
1	197000000002	1970	0	0	NaN	0	NaN	130	
2	197001000001	1970	1	0	NaN	0	NaN	160	
3	197001000002	1970	1	0	NaN	0	NaN	78	
4	197001000003	1970	1	0	NaN	0	NaN	101	

5 rows × 135 columns

```
data.columns
```

```
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)
```

```
data.rename(columns={'iyear':'Year','imonth':'Month','iday':'Day','country_txt':'Country','provstate':'state','region_txt':'Region'
```

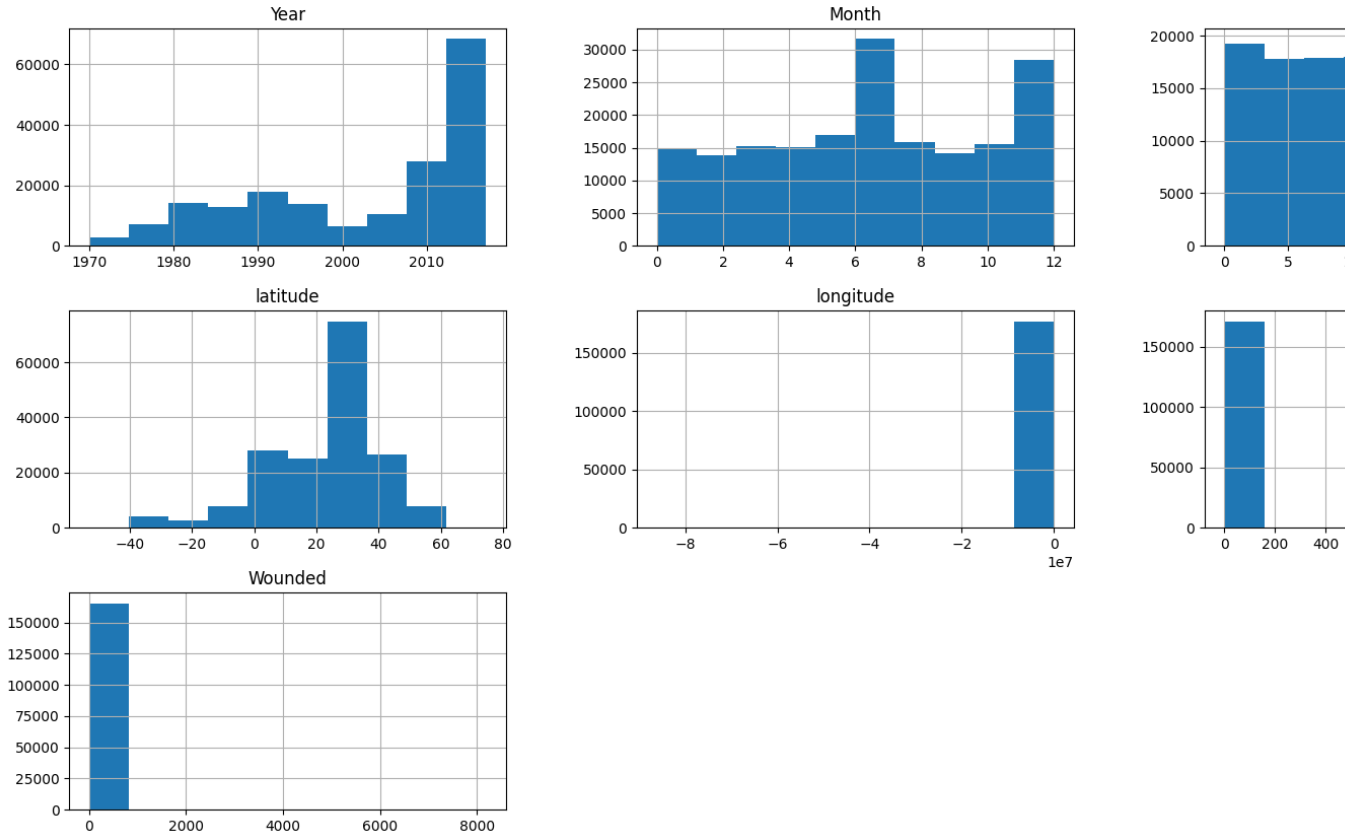
```
data=data[['Year','Month','Day','Country','state','Region','city','latitude','longitude','AttackType','Killed','Wounded','Target','
```

Exploratory Data Analysis

```
```

```
data.hist(figsize=(20,10))  # This represents  the distribution of data  on each series in the DataFrame.
```

```
array([[<Axes: title={'center': 'Year'}>,
        <Axes: title={'center': 'Month'}>,
        <Axes: title={'center': 'Day'}>],
       [<Axes: title={'center': 'latitude'}>,
        <Axes: title={'center': 'longitude'}>,
        <Axes: title={'center': 'Killed'}>],
       [<Axes: title={'center': 'Wounded'}>, <Axes: >, <Axes: >]],
      dtype=object)
```



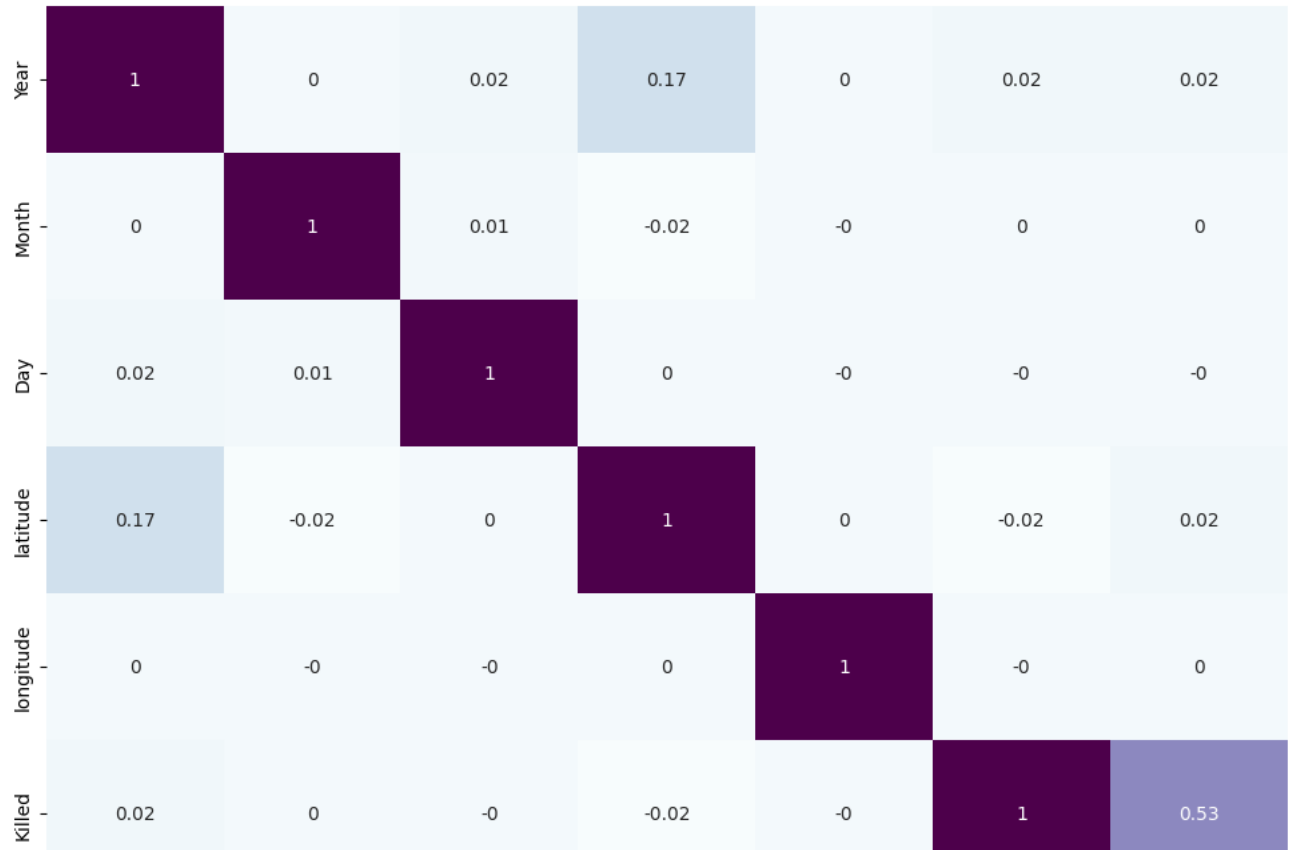

```
data.describe()
```

	Year	Month	Day	latitude	longitude	Killed	Wounded
count	181691.000000	181691.000000	181691.000000	177135.000000	1.771340e+05	171378.000000	165380.000000
mean	2002.638997	6.467277	15.505644	23.498343	-4.586957e+02	2.403272	3.167668
std	13.259430	3.388303	8.814045	18.569242	2.047790e+05	11.545741	35.949392
min	1970.000000	0.000000	0.000000	-53.154613	-8.618590e+07	0.000000	0.000000
25%	1991.000000	4.000000	8.000000	11.510046	4.545640e+00	0.000000	0.000000
50%	2009.000000	6.000000	15.000000	31.467463	4.324651e+01	0.000000	0.000000
75%	2014.000000	9.000000	23.000000	34.685087	6.871033e+01	2.000000	2.000000
max	2017.000000	12.000000	31.000000	74.633553	1.793667e+02	1570.000000	8191.000000

Correlation Analysis

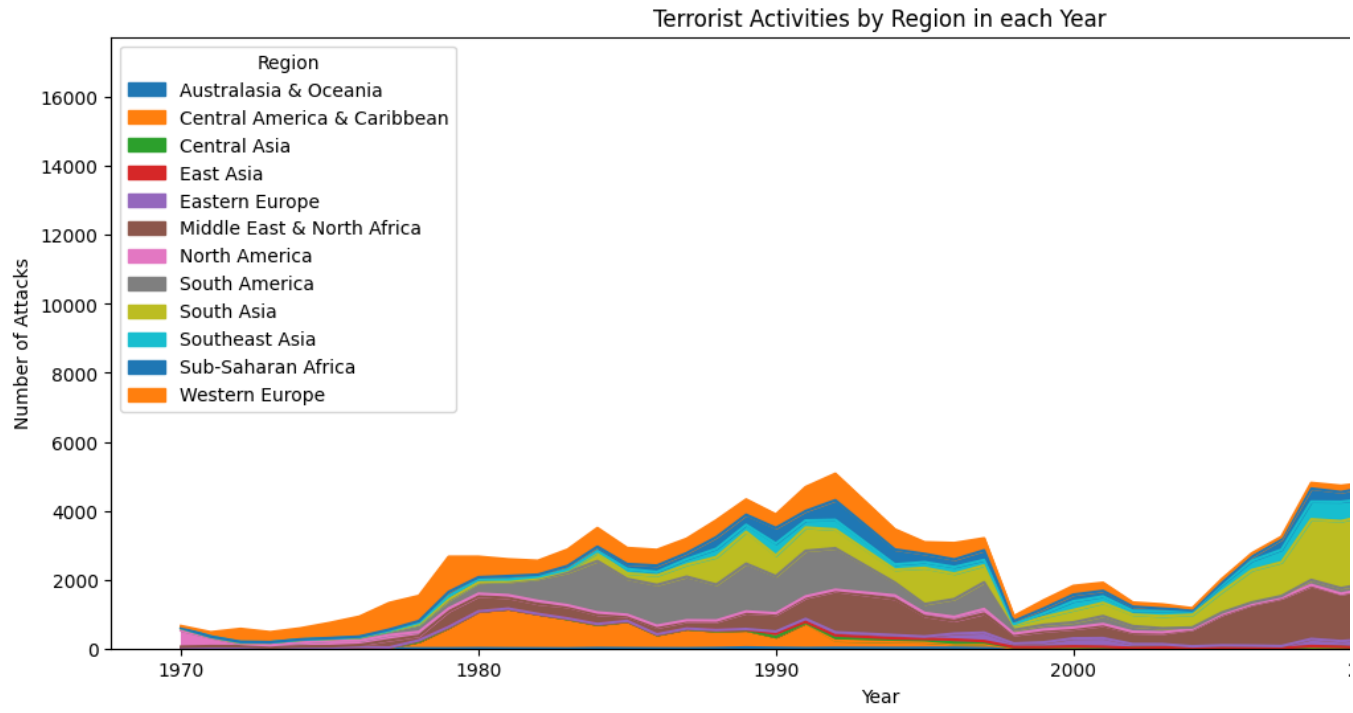
```
plt.figure(figsize=(15,10))
#This shows how much related is one parameter to the other in the dataset.
sns.heatmap(np.round(data.corr(),2),annot=True, cmap='BuPu')
```

```
<ipython-input-11-f1f67925da92>:3: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a futu
  sns.heatmap(np.round(data.corr(),2),annot=True, cmap='BuPu')
<Axes: >
```



Terrorist Activities by Region in each Year through Area Plot

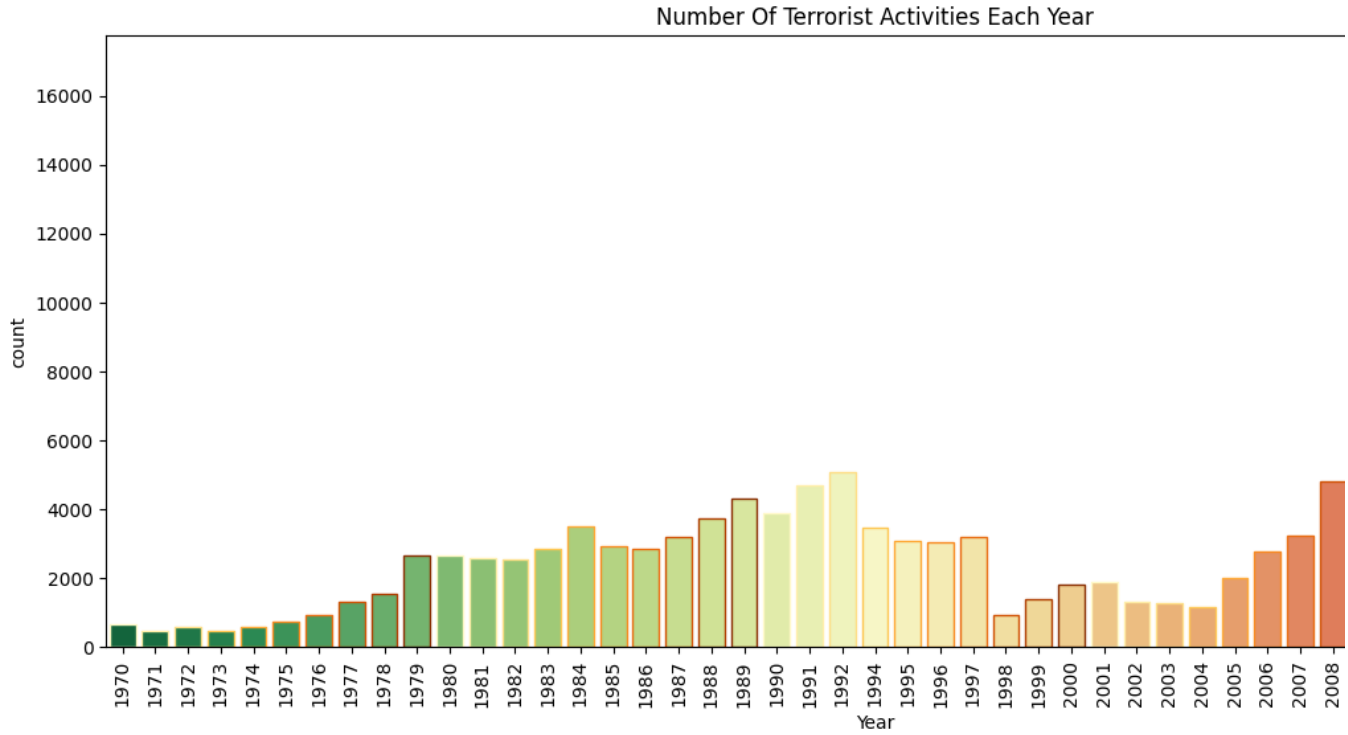
```
pd.crosstab(data.Year, data.Region).plot(kind='area',figsize=(15,6))
plt.title('Terrorist Activities by Region in each Year')
plt.ylabel('Number of Attacks')
plt.show()
```



Number of Terrorist Activities each Year

```
plt.subplots(figsize=(15, 6))
sns.countplot(x='Year', data=data, palette='RdYlGn_r', edgecolor=sns.color_palette("YlOrBr", 10))
plt.xticks(rotation=90)
```

```
plt.title('Number Of Terrorist Activities Each Year')
plt.show()
```



Since 2014 and 2015 has a large number of Terrorist activities. **Number of attack were there in 1970 & 2017 and Also find percentage the attacks have increased. **

```
Year=data.Year.value_counts().to_dict()
rate=((Year[2017]-Year[1970])/Year[2017])*100
print(Year[1970],'attacks happened in 1970 &',Year[2017],'attacks happened in 2017')
print('So the number of attacks from 1970 has increased by',np.round(rate,0),'% till 2017')
```

```
651 attacks happened in 1970 & 10900 attacks happened in 2017
So the number of attacks from 1970 has increased by 94.0 % till 2017
```

```
rate = ((Year[2017] - Year[1970]) / Year[2017]) * 100
```

```
print(Year.keys())
```

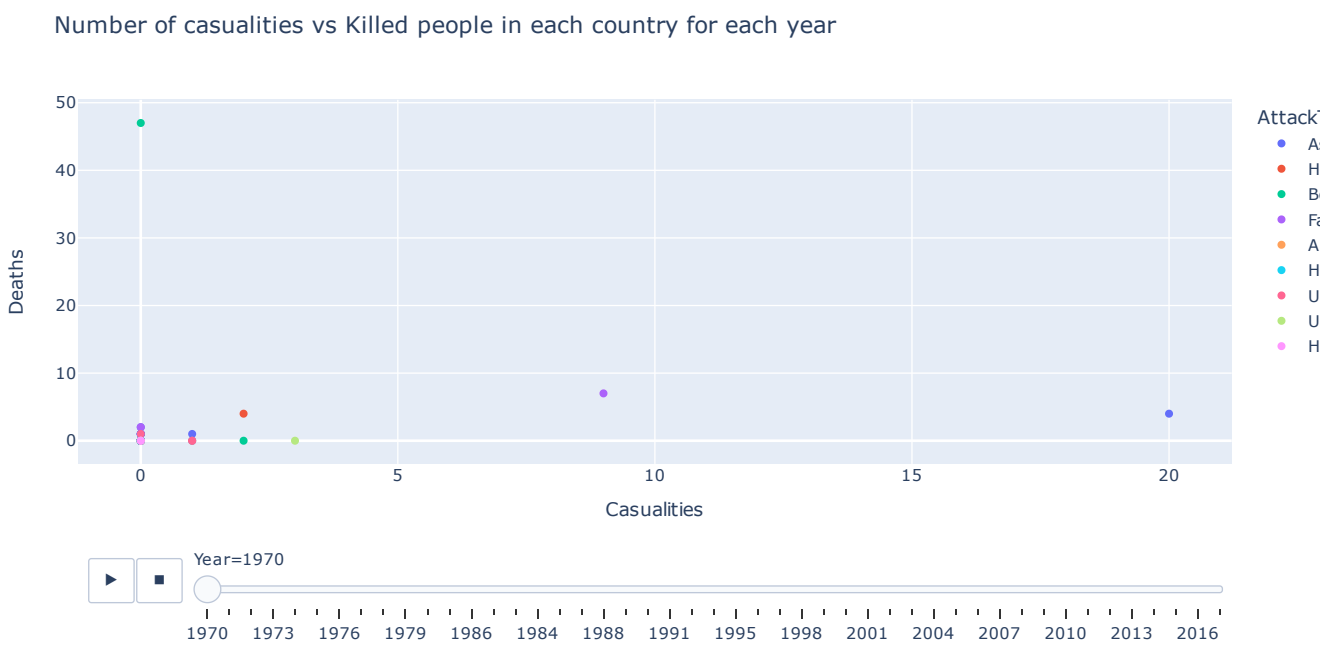
```
dict_keys([2014, 2015, 2016, 2013, 2017, 2012, 2011, 1992, 2010, 2008, 2009, 1991, 1989, 1990, 1988, 1984, 1994, 2007, 1997, 1
```

```
# Assuming 'Year' is the correct column name
Year = data['Year'].value_counts().to_dict()

# Check the keys in the dictionary
print(Year.keys())

# Calculate the rate of increase
rate = ((Year[2017] - Year[1970]) / Year[2017]) * 100

# Print the results
print(Year[1970], 'attacks happened in 1970 &', Year[2017], 'attacks happened in 2017')
print('So the number of attacks from 1970 has increased by', np.round(rate, 0), '% till 2017')
```

```
dict_keys([2014, 2015, 2016, 2013, 2017, 2012, 2011, 1992, 2010, 2008, 2009, 1991, 1989, 1990, 1988, 1984, 1994, 2007, 1997, 1
651 attacks happened in 1970 & 10900 attacks happened in 2017
So the number of attacks from 1970 has increased by 94.0 % till 2017
```

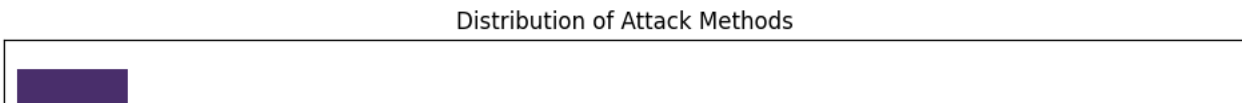
The number of casualities corresponding to the killed people in each country for each year.

```
px.scatter(data,data.Wounded,data.Killed,hover_name='Country',animation_frame='Year',animation_group='Country',color='AttackType',
           range_color=[0,1],labels={'Killed':'Deaths','Wounded':'Casualities'},
           title='Number of casualities vs Killed people in each country for each year')
```



Methods of attack

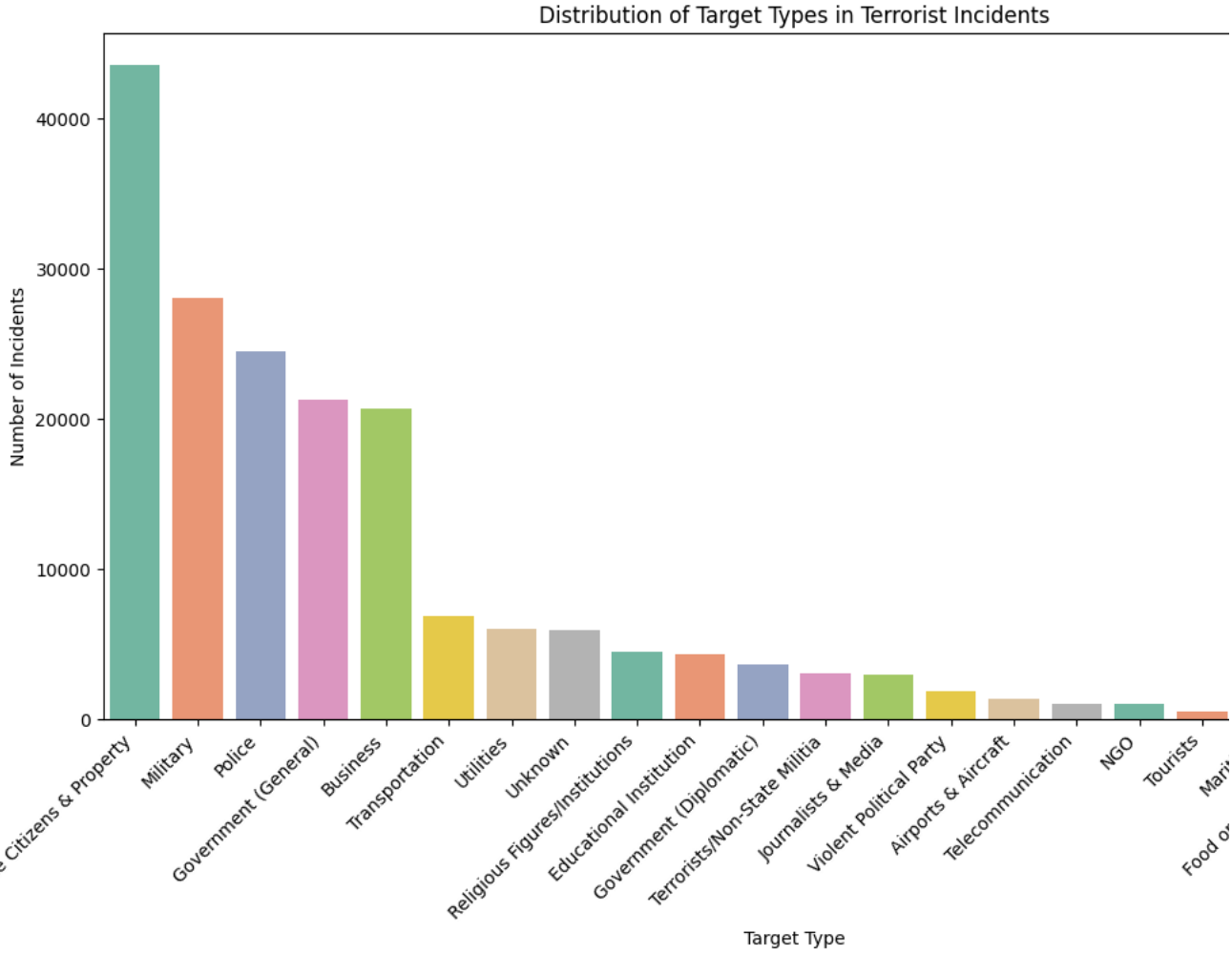
```
# Count the occurrences of each attack method
attack_methods = data['AttackType'].value_counts()

# Plotting the results
plt.figure(figsize=(12, 6))
sns.barplot(x=attack_methods.index, y=attack_methods.values, palette='viridis')
plt.xticks(rotation=45, ha='right')  # Rotate x-axis labels for better readability
plt.title('Distribution of Attack Methods')
plt.xlabel('Attack Method')
plt.ylabel('Number of Incidents')
plt.show()
```



Since from the above chart it is clear that Bombing/Explosion method was mostly used.

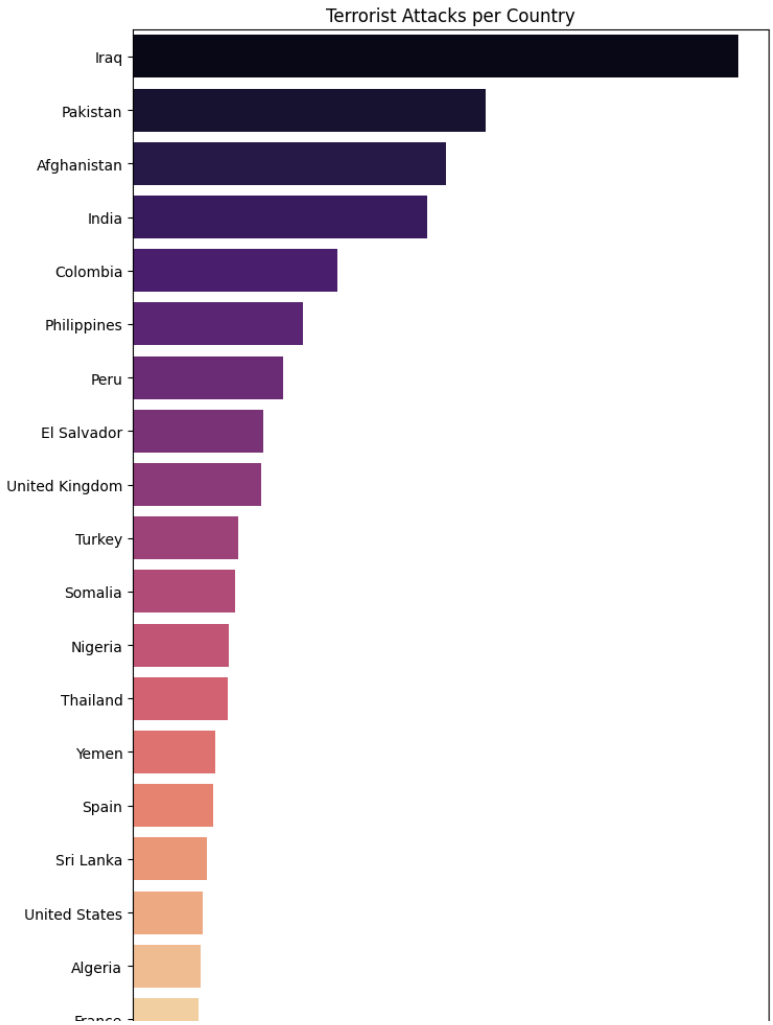
Type of target

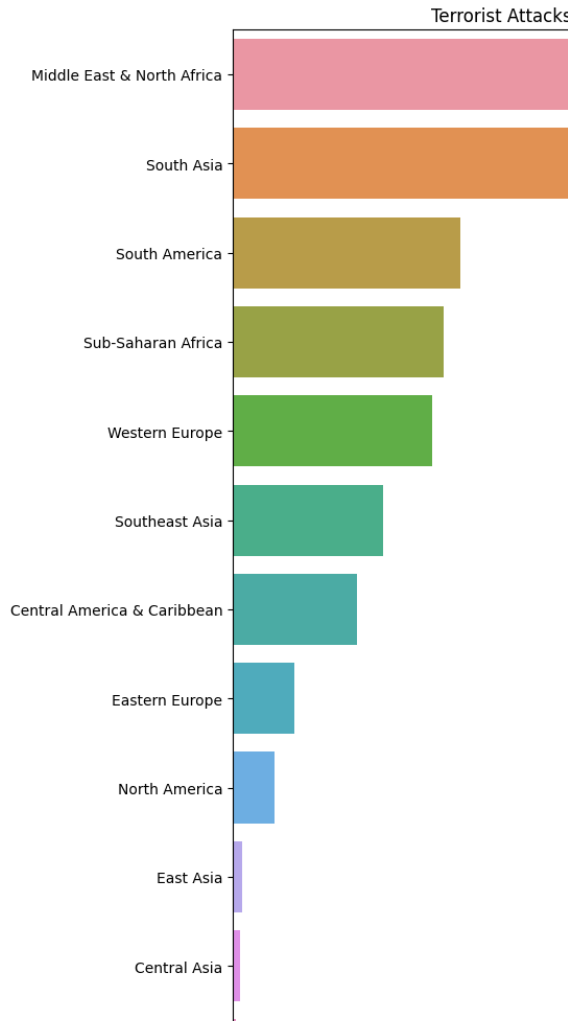
```
# Count the occurrences of each target type
target_types = data['Target_type'].value_counts()

# Plotting the results
plt.figure(figsize=(14, 7))
sns.barplot(x=target_types.index, y=target_types.values, palette='Set2')
plt.xticks(rotation=45, ha='right')  # Rotate x-axis labels for better readability
plt.title('Distribution of Target Types in Terrorist Incidents')
plt.xlabel('Target Type')
plt.ylabel('Number of Incidents')
plt.show()
```



Total number of terrorist attack in each country and regions using barplot

```
fig,axes = plt.subplots(figsize=(16,11),nrows=1,ncols=2)
sns.barplot(x = data['Country'].value_counts()[:20].values, y = data['Country'].value_counts()[:20].index,
            ax=axes[0],palette = 'magma');
axes[0].set_title('Terrorist Attacks per Country')
sns.barplot(x=data['Region'].value_counts().values,y=data['Region'].value_counts().index,
            ax=axes[1])
axes[1].set_title('Terrorist Attacks per Region')
fig.tight_layout()
plt.show()
```





The total number of attacks in each country using Globe

We will first create a new dataframe which contains the total count of attacks for each country in order to use it as a colour bar.

```
terr=data.groupby(['Country'],as_index=False).count()
```

```
fig=px.choropleth(terr,locations='Country',locationmode='country names',
                  color='Year',hover_name='Country',projection='orthographic',
                  title='Total number of attacks (1970-2017)',labels={'Year':'Attacks'})
fig.show()
```

Country suffered the maximum attacks(state the number) and country suffered minimum attacks(state the number)

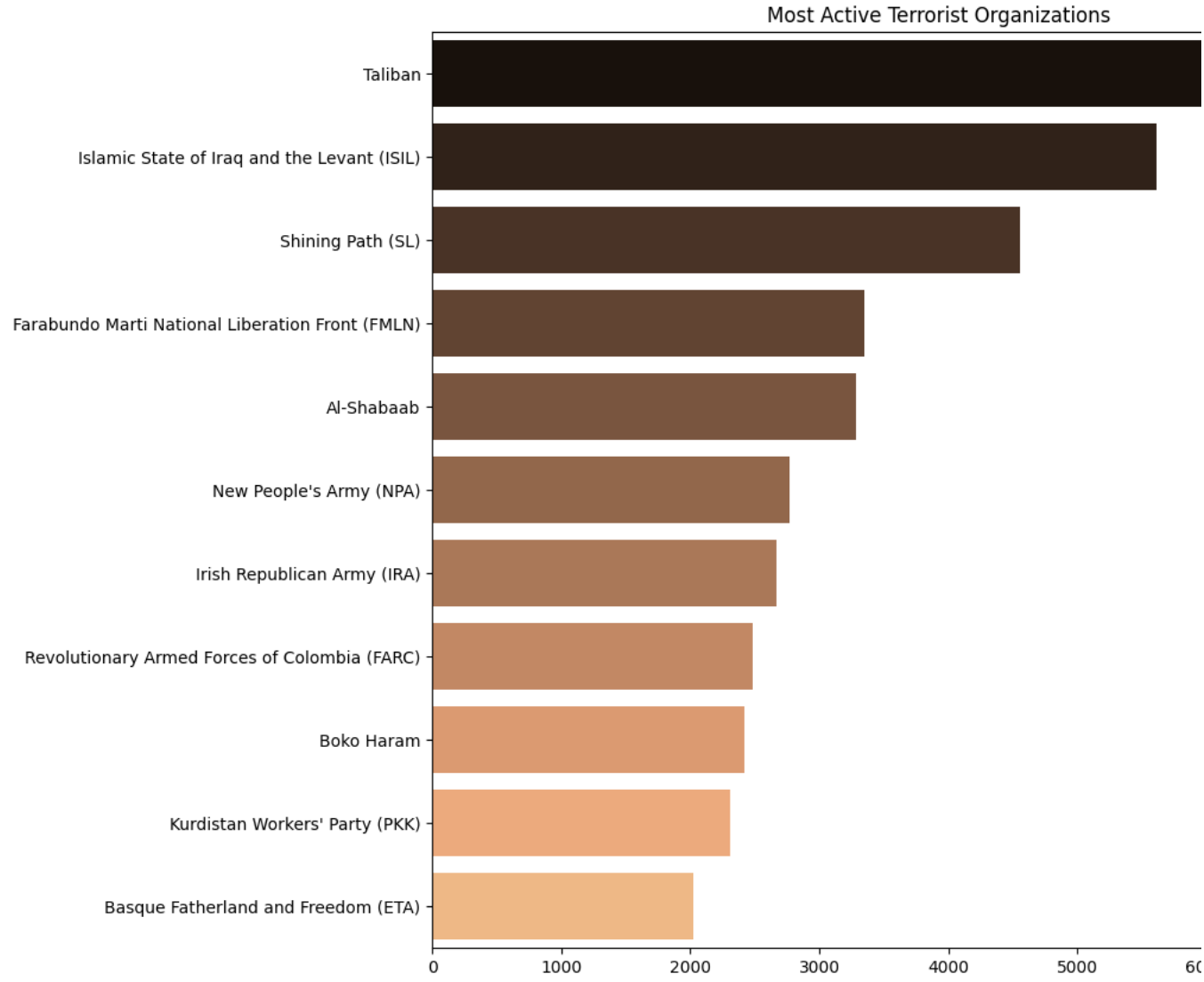
```
max_count=terr['Year'].max()
max_id=terr['Year'].idxmax()
max_name=terr['Country'][max_id]
min_count=terr['Year'].min()
min_id=terr['Year'].idxmin()
min_name=terr['Country'][min_id]
```

```
print(max_name,'has suffered the maximum number of terror attacks of',max_count)
print(min_name,'has suffered the minimum number of terror attacks of',min_count)
```

```
Iraq has suffered the maximum number of terror attacks of 24636
Andorra has suffered the minimum number of terror attacks of 1
```

Now let us check out which terrorist organizations have carried out their operations in each country.

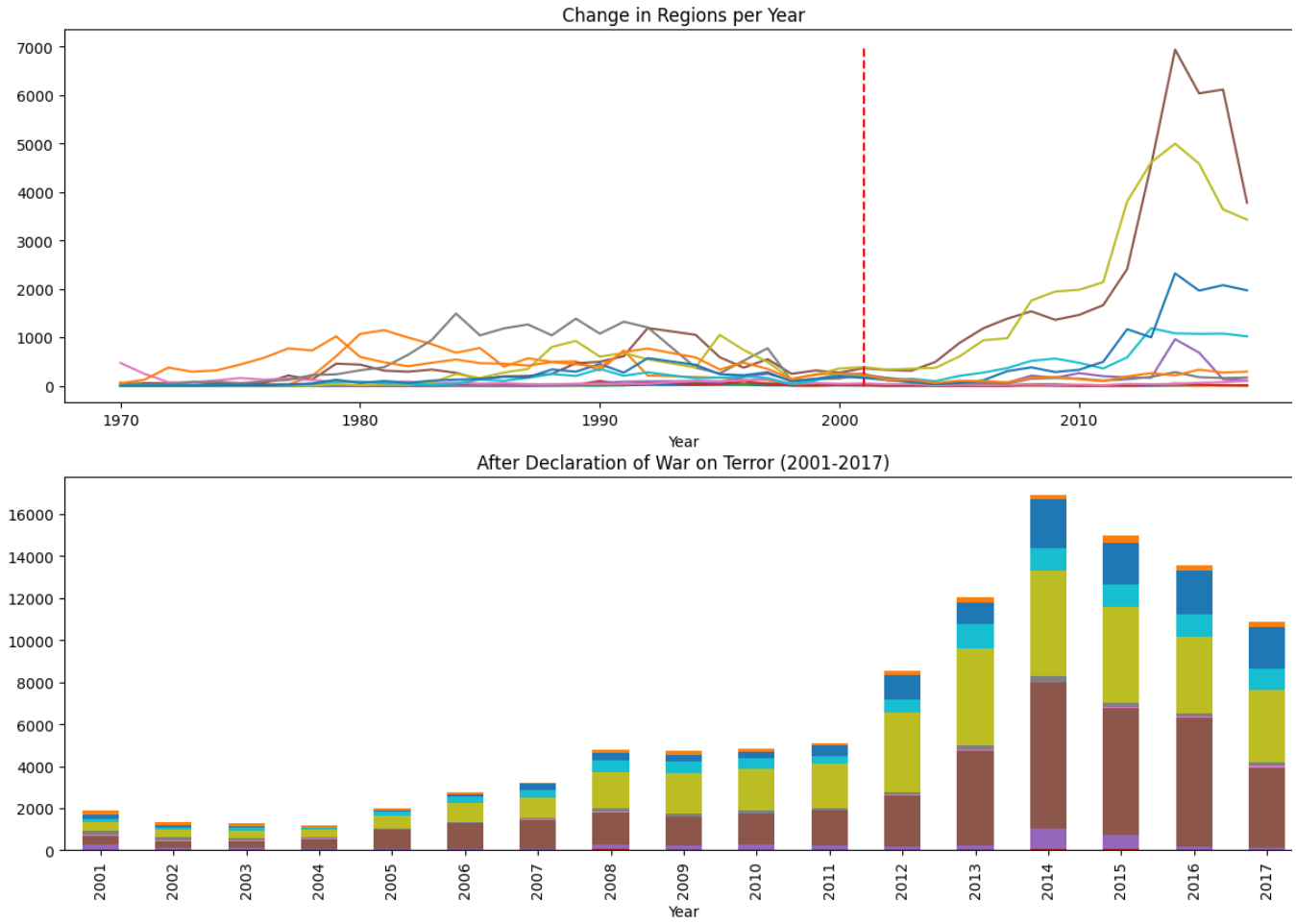
```
plt.subplots(figsize=(11,10))
sns.barplot(y=data['Group'].value_counts()[1:12].index,x=data['Group'].value_counts()[1:12].values,
            palette='copper')
plt.title('Most Active Terrorist Organizations')
plt.show()
```



Now we will look closer at trend Before and after the War on Terror

```
data_after = data[data['Year']>=2001]
fig,ax = plt.subplots(figsize=(15,10),nrows=2,ncols=1)
ax[0] = pd.crosstab(data.Year,data.Region).plot(ax=ax[0])
ax[0].set_title('Change in Regions per Year')
ax[0].legend(loc='center left',bbox_to_anchor = (1,0.5))
ax[0].vlines(x=2001,ymin=0,ymax=7000,colors='red',linestyles='--')
```

```
pd.crosstab(data_after.Year,data_after.Region).plot.bar(stacked=True,ax=ax[1])
ax[1].set_title('After Declaration of War on Terror (2001-2017)')
ax[1].legend(loc='center left',bbox_to_anchor = (1,0.5))
plt.show()
```



From the first plot, it is very noticeable that the terrorism landscape before and after the War on Terror is vastly different. Before 2001, the regions were much closer together in terms of activity, eventually all dropping to a minimum in 2000. After 2001, the Middle East and South Asia have dictated the rise in terrorism numbers, with a significant increase in Sub-Saharan Africa as well

Insights derived from the above EDA: Attacks has increased but number of people killed manier times as attack happened. Iraq has the most attacks. The Middle East and North Africa Regions has most taregeted. Maximum number of attacks are from Bombing/Explosions. There are maximum number of attacks in Private citizens and Property. Taliban and ISIL has a most active organisation.