

```
In [ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
import seaborn as sns
```

```
In [ ]: data = pd.read_csv('terror.csv', encoding='latin-1')
pd.set_option('display.max_columns',None)
data.head()
```

C:\Users\Hunain\AppData\Local\Temp\ipykernel_11892\3214474236.py:1: 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.

```
data = pd.read_csv('terror.csv', encoding='latin-1')
```

```
Out[ ]:      eventid  iyear  imonth  iday  approxdate  extended  resolution  country  country_txt  region
```

0	1970000000001	1970	7	2	NaN	0	NaN	58	Dominican Republic
1	1970000000002	1970	0	0	NaN	0	NaN	130	Mexico
2	1970010000001	1970	1	0	NaN	0	NaN	160	Philippines
3	1970010000002	1970	1	0	NaN	0	NaN	78	Greece
4	1970010000003	1970	1	0	NaN	0	NaN	101	Japan

```
In [ ]: data.rename(columns={'iyear':'Year','imonth':'Month','iday':'Day','country_txt':'Country',
                             'region_txt':'Region','attacktype1_txt':'AttackType','target1':
                             'Wounded','summary':'Summary','gname':'Group','target_type':
                             'Weapon_type','motive':'Motive'},inplace=True)
```

```
In [ ]: data = data[['Year','Month','Day','Country','Region','state','city','latitude','longitude',
                    'Summary','Group','Target_type','Weapon_type','Motive']]
```

```
In [ ]: data.shape
```

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

```
In [ ]: data.describe()
```

```
Out[ ]:
```

	Year	Month	Day	latitude	longitude	Killed
count	181691.000000	181691.000000	181691.000000	177135.000000	1.771340e+05	171378.000000
mean	2002.638997	6.467277	15.505644	23.498343	-4.586957e+02	2.403272

	Year	Month	Day	latitude	longitude	Killed
std	13.259430	3.388303	8.814045	18.569242	2.047790e+05	11.545741
min	1970.000000	0.000000	0.000000	-53.154613	-8.618590e+07	0.000000
25%	1991.000000	4.000000	8.000000	11.510046	4.545640e+00	0.000000
50%	2009.000000	6.000000	15.000000	31.467463	4.324651e+01	0.000000
75%	2014.000000	9.000000	23.000000	34.685087	6.871033e+01	2.000000
max	2017.000000	12.000000	31.000000	74.633553	1.793667e+02	1570.000000

```
In [ ]: data.columns.to_list()
```

```
Out[ ]: ['Year',
        'Month',
        'Day',
        'Country',
        'Region',
        'state',
        'city',
        'latitude',
        'longitude',
        'AttackType',
        'Killed',
        'Wounded',
        'Target',
        'Summary',
        'Group',
        'Target_type',
        'Weapon_type',
        'Motive']
```

```
In [ ]: data.dtypes
```

```
Out[ ]: Year          int64
        Month         int64
        Day           int64
        Country       object
        Region        object
        state         object
        city          object
        latitude      float64
        longitude     float64
        AttackType    object
        Killed        float64
        Wounded       float64
        Target        object
        Summary       object
        Group         object
        Target_type   object
        Weapon_type   object
        Motive        object
        dtype: object
```

```
In [ ]: data.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   Region    181691 non-null object
5   state     181270 non-null object
6   city      181256 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    181053 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

```

```
In [ ]: data.isnull().sum()
```

```

Out[ ]: Year      0
Month      0
Day        0
Country    0
Region     0
state      421
city       435
latitude   4556
longitude  4557
AttackType  0
Killed     10313
Wounded    16311
Target      638
Summary    66129
Group       0
Target_type 0
Weapon_type 0
Motive     131130
dtype: int64

```

Dropping Null Values

```
In [ ]: data = data.dropna()
```

```
In [ ]: data.shape
```

```
Out[ ]: (46554, 18)
```

Dropping Duplicate Values

```
In [ ]: data.drop_duplicates(inplace=True)
```

```
In [ ]: data.shape
```

```
Out[ ]: (45965, 18)
```

we have to remove two more columns holding majority null values from the data set as we dont need them in our requirement

```
In [ ]: data.isnull().sum()
```

```
Out[ ]: Year          0
        Month         0
        Day           0
        Country       0
        Region        0
        state         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
```

Neccessary Information

Highest Time Attacked

```
In [ ]: data['city'].value_counts()
```

```
Out[ ]: city
        Baghdad      3590
        Unknown      1269
        Mosul        876
        Karachi      712
        Kirkuk       524
        ...
        Belagua       1
        Varani        1
        Baskale       1
        Sitapur       1
        Spinetoli     1
        Name: count, Length: 13082, dtype: int64
```

```
In [ ]: data['Region'].value_counts()
```

```
Out[ ]: Region
        South Asia          15749
        Middle East & North Africa  13984
        Southeast Asia       4609
        Sub-Saharan Africa    3518
        Western Europe       2366
        Eastern Europe       2092
        South America        1619
        North America        1531
        Central Asia         224
        East Asia            151
        Central America & Caribbean  72
        Australasia & Oceania    50
        Name: count, dtype: int64
```

In []:

```
data.head()
```

Out[]:

	Year	Month	Day	Country	Region	state	city	latitude	longitude	Attac
5	1970	1	1	United States	North America	Illinois	Cairo	37.005105	-89.176269	Armed A
8	1970	1	2	United States	North America	Wisconsin	Madison	43.076592	-89.412488	Facility/Infrastr
9	1970	1	3	United States	North America	Wisconsin	Madison	43.072950	-89.386694	Facility/Infrastr
11	1970	1	6	United States	North America	Colorado	Denver	39.758968	-104.876305	Facility/Infrastr
14	1970	1	9	United States	North America	Puerto Rico	Rio Piedras	18.386932	-66.061127	Facility/Infrastr



Highest Attack Information

In []:

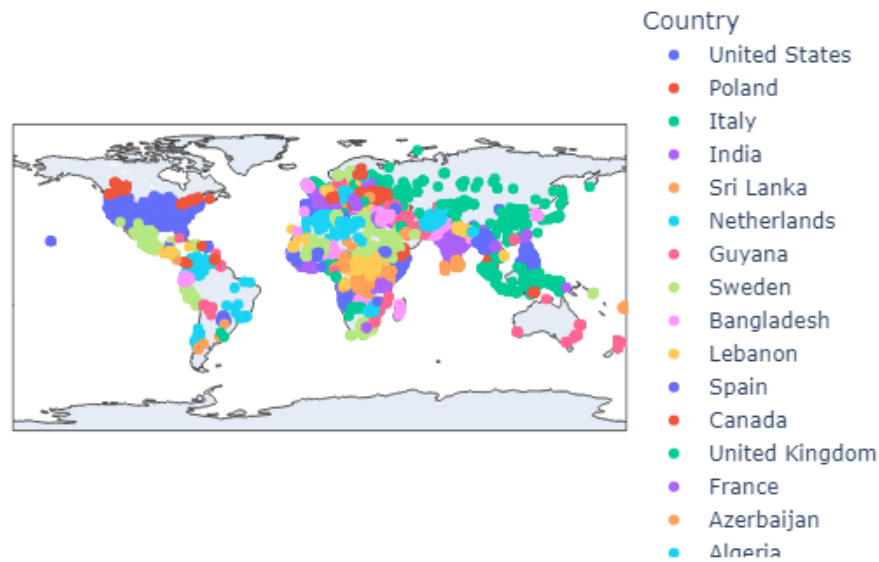
```
print("Country      : ", data['Country'].value_counts().idxmax(), " : ", data['Cou
print("City        : ", data['city'].value_counts().idxmax(), " : ", data['city'].val
print("Region      : ", data['Region'].value_counts().idxmax(), " : ", data['Region'].v
print("State       : ", data['state'].value_counts().idxmax(), " : ", data['state'].v
print("Attack Type : ", data['AttackType'].value_counts().idxmax(), " : ", data['Atta
print("Target      : ", data['Target'].value_counts().idxmax(), " : ", data['Target']
print("Target Type : ", data['Target_type'].value_counts().idxmax(), " : ", data['Tar
print("Weapon Type : ", data['Weapon_type'].value_counts().idxmax(), " : ", data['Wea
print("Year        : ", data['Year'].value_counts().idxmax(), " : ", data['Year'].val
```

```
Country      : Iraq      : 8658
City         : Baghdad   : 3590
Region       : South Asia : 15749
State        : Baghdad   : 3588
Attack Type  : Bombing/Explosion : 23193
Target       : Civilians  : 2188
Target Type  : Private Citizens & Property : 13721
Weapon Type  : Explosives : 24337
Year         : 2011      : 4846
```

Visualization of Data

```
In [ ]: fig = px.scatter_geo(data, lat='latitude', lon='longitude', color='Country',
                           , hover_name='city',
                           title='Terrorist Attack in the World by City & Country')
fig.show("png")
```

Terrorist Attack in the World by City & Country

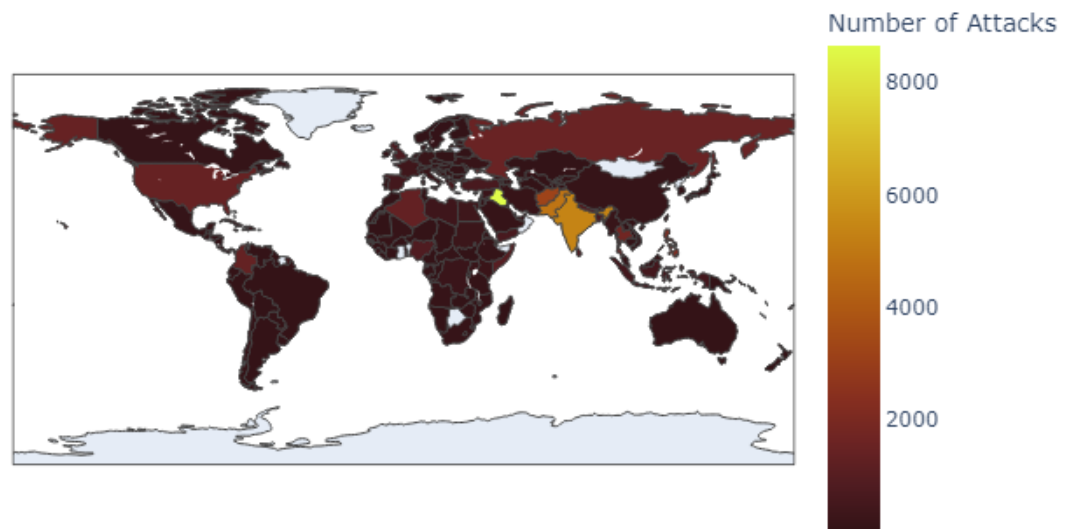


```
In [ ]: country_counts = data['Country'].value_counts().reset_index()
country_counts.columns = ['Country', 'Attack Count']

fig = px.choropleth(country_counts,
                    locations='Country',
                    locationmode='country names',
                    color='Attack Count',
                    title='No. Terrorist Attacks in the world by Country',
                    labels={'Attack Count': 'Number of Attacks'},
                    hover_name='Country',
                    color_continuous_scale='solar')

fig.show("png")
```

No. Terrorist Attacks in the world by Country



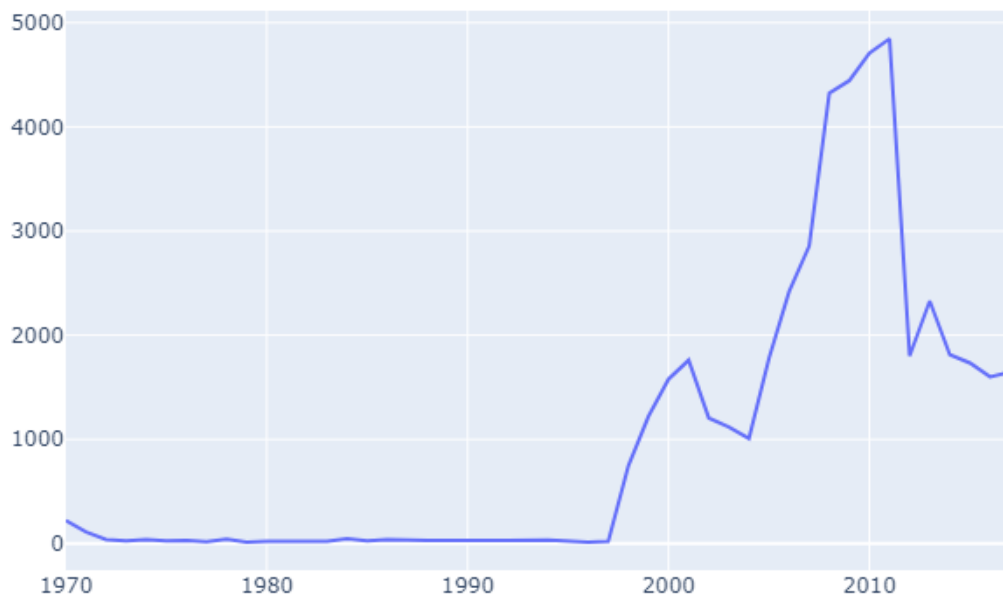
```
In [ ]: # country_counts_for_year = data['Country'].value_counts().reset_index()
# country_counts_for_year.columns = ['Country', 'Attack Count']

# sns.lineplot(data, x=data['Year'], y=data['Country'].value_counts())

yearly_counts = data['Year'].value_counts().sort_index()

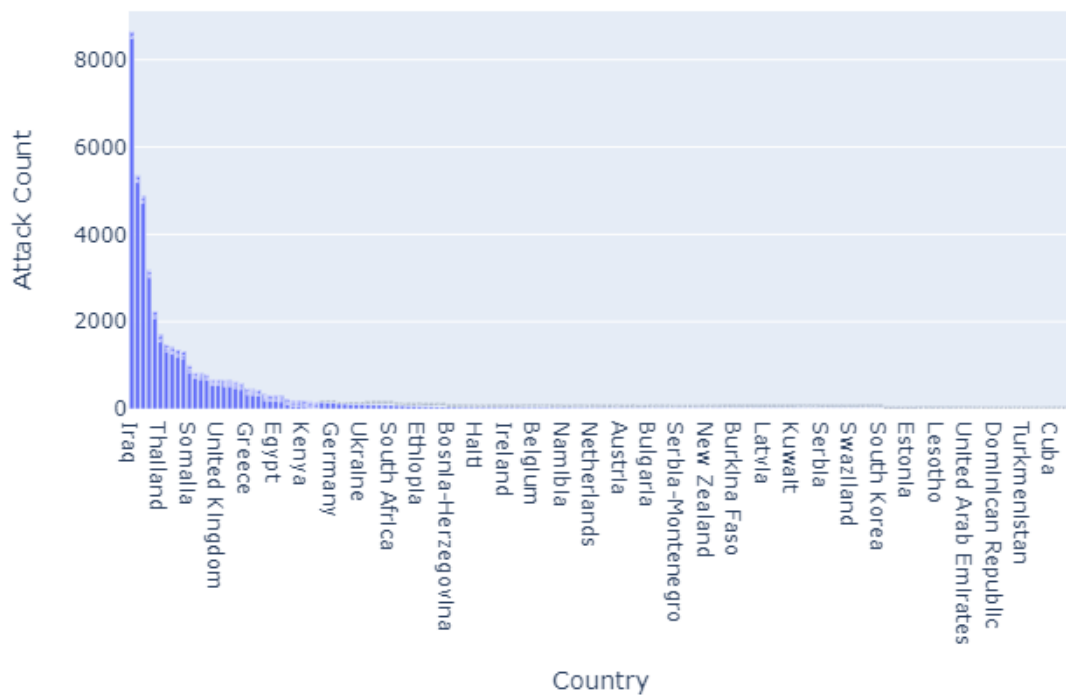
fig = go.Figure([go.Scatter(x=yearly_counts.index, y=yearly_counts.values)])
fig.update_layout(title = 'Attacks over Time')
fig.show("png")
```

Attacks over Time



```
In [ ]: # df = px.data.gapminder().query("continent == 'Europe' and year == 2007 and pop > 2
fig = px.bar(country_counts, y='Attack Count', x='Country', text_auto='.2s',
            title="Default: various text sizes, positions and angles")
fig.show("png")
```

Default: various text sizes, positions and angles



```
In [ ]: region_attack_counts = data.groupby(['Region', 'AttackType']).size().reset_index()
region_attack_counts.columns = ['Region', 'AttackType', 'Attack Count']
```



```
fig = px.bar(region_attack_counts, y='Attack Count', x='Region', text_auto='.2s',
             title="No. of Attacks on Regions with Attack Types", hover_name='AttackType')
fig.update_traces(textfont_size=12, textangle=0, textposition="outside", cliponaxis=False)
fig.show("png")
```

No. of Attacks on Regions with Attack Types

