In [1]:
```python
#importing the libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
sns.set(style="whitegrid")
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import RobustScaler
```

In [2]:
```python
#Loading the dataset
dataset = pd.read_csv('globalterrorismdb_0718dist.csv' ,encoding= 'ISO-8859-1')
print(f'Data Shape is {dataset.shape}')
dataset.head()
```

```
C:\Users\akhil\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3165: DtypeWarning: Columns (4,6,31,33,61,62,63,76,79,90,92,94,96,114,1
15,121) have mixed types.Specify dtype option on import or set low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
Data Shape is (181691, 135)
```

Out[2]:

	eventid	iyear	imonth	iday	approxdate	extended	resolution	country	country_txt	region	...	addnotes	scite1	scite2	scite3	dbsource	INT_LOG	INT_IDEO	INT_MISC
0	197000000001	1970	7	2	NaN	0	NaN	58	Dominican Republic	2	...	NaN	NaN	NaN	NaN	PGIS	0	0	0
1	197000000002	1970	0	0	NaN	0	NaN	130	Mexico	1	...	NaN	NaN	NaN	NaN	PGIS	0	1	1
2	197001000001	1970	1	0	NaN	0	NaN	160	Philippines	5	...	NaN	NaN	NaN	NaN	PGIS	-9	-9	1
3	197001000002	1970	1	0	NaN	0	NaN	78	Greece	8	...	NaN	NaN	NaN	NaN	PGIS	-9	-9	1
4	197001000003	1970	1	0	NaN	0	NaN	101	Japan	4	...	NaN	NaN	NaN	NaN	PGIS	-9	-9	1

5 rows × 135 columns

In [3]:
```python
#Function definitions:
def drop(feature) :
    global dataset
    dataset.drop([feature],axis=1, inplace=True)
    dataset.head()

def unique(feature) :
    global dataset
    print(f'Number of unique vaure are {len(list(dataset[feature].unique()))} which are : \n {list(dataset[feature].unique())}')

def unique_all(show_value = True) :
    global dataset
    for col in dataset.columns :
        print(f'Length of unique data for   {col}   is    {len(dataset[col].unique())} ')
        if show_value == True  :
            print(f'unique values ae {dataset[col].unique()}' )
            print('-----------------------------')
def drop_nulls(percentage = 0.3) :
    global dataset
    for col in dataset.columns :
        ratio =  dataset[col].isna().sum()/dataset.shape[0]
        if ratio >= percentage :
            dataset.drop([col],axis=1, inplace=True)
            print(f'Column {col} has been dropped since nulls percentage is {round(ratio *100)} %')

def count_nulls() :
    global dataset
    for col in dataset.columns :
        if not dataset[col].isna().sum() == 0 :  print(f'Column {col} has been number of nulls {dataset[col].isna().sum()}')


def fillna(feature , val = 'none') :
    global dataset
    dataset[feature].fillna(val, inplace=True)

def cplot(feature) :
    global data
    sns.countplot(x=feature, dataset=dataset,facecolor=(0, 0, 0, 0),linewidth=5,edgecolor=sns.color_palette("dark", 3))


def spie(series) :
    global dataset
    plt.pie(series.values,labels=list(series.index),autopct ='%1.2f%%',labeldistance = 1.1,explode = [0.05 for i in range(len(series.values))])
    plt.show()

def pie(feature) :
    global dataset
    plt.pie(dataset[feature].value_counts(),labels=list(dataset[feature].value_counts().index),autopct ='%1.2f%%' , labeldistance = 1.1,explode = [
    plt.show()

def make_xy(feature) :
    global dataset
    X = dataset.drop([feature], axis=1, inplace=False)
    y = dataset[feature]
    return X , y

def encoder(feature , new_feature, drop = True) :
    global dataset
    enc  = LabelEncoder()
    enc.fit(dataset[feature])
    dataset[new_feature] = enc.transform(dataset[feature])
    if drop == True :
        dataset.drop([feature],axis=1, inplace=True)

def max_counts(feature, number, return_rest = False) :
    global dataset
    counts = dataset[feature].value_counts()
    values_list = list(counts[:number].values)
    rest_value =  sum(counts.values) - sum (values_list)
    index_list = list(counts[:number].index)

    if return_rest :
        values_list.append(rest_value )
        index_list.append('rest items')

    result = pd.Series(values_list, index=index_list)
    if len(dataset[feature]) <= number :
        result = None
    return result

def remove_zero(feature , val = 0) :
    global dataset
    dataset = dataset[dataset[feature] != val]

def show_details() :
    global dataset
    for col in dataset.columns :
        print(f'for feature : {col}')
        print(f'Number of Nulls is   {dataset[col].isna().sum()}')
        print(f'Number of Unique values is   {len(dataset[col].unique())}')
        print(f'random Value {dataset[col][0]}')
        print(f'random Value {dataset[col][10]}')
        print(f'random Value {dataset[col][20]}')
        print('--------------------------')
```

In [4]:
```python
#DATA CLEANING
dataset.shape
```

Out[4]: (181691, 135)

In [5]:
```python
drop_nulls()
```

```
Column approxdate has been dropped since nulls percentage is 95 %
Column resolution has been dropped since nulls percentage is 99 %
Column location has been dropped since nulls percentage is 69 %
Column summary has been dropped since nulls percentage is 36 %
Column alternative has been dropped since nulls percentage is 84 %
Column alternative_txt has been dropped since nulls percentage is 84 %
Column attacktype2 has been dropped since nulls percentage is 97 %
Column attacktype2_txt has been dropped since nulls percentage is 97 %
Column attacktype3 has been dropped since nulls percentage is 100 %
Column attacktype3_txt has been dropped since nulls percentage is 100 %
Column targtype2 has been dropped since nulls percentage is 94 %
Column targtype2_txt has been dropped since nulls percentage is 94 %
Column targsubtype2 has been dropped since nulls percentage is 94 %
Column targsubtype2_txt has been dropped since nulls percentage is 94 %
Column corp2 has been dropped since nulls percentage is 94 %
Column target2 has been dropped since nulls percentage is 94 %
Column natlty2 has been dropped since nulls percentage is 94 %
Column natlty2_txt has been dropped since nulls percentage is 94 %
Column targtype3 has been dropped since nulls percentage is 99 %
Column targtype3_txt has been dropped since nulls percentage is 99 %
Column targsubtype3 has been dropped since nulls percentage is 99 %
Column targsubtype3_txt has been dropped since nulls percentage is 99 %
Column corp3 has been dropped since nulls percentage is 99 %
Column target3 has been dropped since nulls percentage is 99 %
Column natlty3 has been dropped since nulls percentage is 99 %
Column natlty3_txt has been dropped since nulls percentage is 99 %
Column gsubname has been dropped since nulls percentage is 97 %
Column gname2 has been dropped since nulls percentage is 99 %
Column gsubname2 has been dropped since nulls percentage is 100 %
Column gname3 has been dropped since nulls percentage is 100 %
Column gsubname3 has been dropped since nulls percentage is 100 %
Column motive has been dropped since nulls percentage is 72 %
Column guncertain2 has been dropped since nulls percentage is 99 %
Column guncertain3 has been dropped since nulls percentage is 100 %
Column nperps has been dropped since nulls percentage is 39 %
Column nperpcap has been dropped since nulls percentage is 38 %
Column claimed has been dropped since nulls percentage is 36 %
Column claimmode has been dropped since nulls percentage is 89 %
Column claimmode_txt has been dropped since nulls percentage is 89 %
Column claim2 has been dropped since nulls percentage is 99 %
Column claimmode2 has been dropped since nulls percentage is 100 %
Column claimmode2_txt has been dropped since nulls percentage is 100 %
Column claim3 has been dropped since nulls percentage is 100 %
Column claimmode3 has been dropped since nulls percentage is 100 %
Column claimmode3_txt has been dropped since nulls percentage is 100 %
Column compclaim has been dropped since nulls percentage is 97 %
Column weaptype2 has been dropped since nulls percentage is 93 %
Column weaptype2_txt has been dropped since nulls percentage is 93 %
Column weapsubtype2 has been dropped since nulls percentage is 94 %
Column weapsubtype2_txt has been dropped since nulls percentage is 94 %
Column weaptype3 has been dropped since nulls percentage is 99 %
Column weaptype3_txt has been dropped since nulls percentage is 99 %
Column weapsubtype3 has been dropped since nulls percentage is 99 %
Column weapsubtype3_txt has been dropped since nulls percentage is 99 %
Column weaptype4 has been dropped since nulls percentage is 100 %
Column weaptype4_txt has been dropped since nulls percentage is 100 %
Column weapsubtype4 has been dropped since nulls percentage is 100 %
Column weapsubtype4_txt has been dropped since nulls percentage is 100 %
Column weapdetail has been dropped since nulls percentage is 37 %
Column nkillus has been dropped since nulls percentage is 35 %
Column nkillter has been dropped since nulls percentage is 37 %
Column nwoundus has been dropped since nulls percentage is 36 %
Column nwoundte has been dropped since nulls percentage is 38 %
Column propextent has been dropped since nulls percentage is 65 %
Column propextent_txt has been dropped since nulls percentage is 65 %
Column propvalue has been dropped since nulls percentage is 79 %
Column propcomment has been dropped since nulls percentage is 68 %
Column nhostkid has been dropped since nulls percentage is 93 %
Column nhostkidus has been dropped since nulls percentage is 93 %
Column nhours has been dropped since nulls percentage is 98 %
Column ndays has been dropped since nulls percentage is 96 %
Column divert has been dropped since nulls percentage is 100 %
Column kidhijcountry has been dropped since nulls percentage is 98 %
Column ransom has been dropped since nulls percentage is 57 %
Column ransomamt has been dropped since nulls percentage is 99 %
Column ransomamtus has been dropped since nulls percentage is 100 %
Column ransompaid has been dropped since nulls percentage is 100 %
Column ransompaidus has been dropped since nulls percentage is 100 %
Column ransomnote has been dropped since nulls percentage is 100 %
Column hostkidoutcome has been dropped since nulls percentage is 94 %
Column hostkidoutcome_txt has been dropped since nulls percentage is 94 %
Column nreleased has been dropped since nulls percentage is 94 %
Column addnotes has been dropped since nulls percentage is 84 %
Column scite1 has been dropped since nulls percentage is 36 %
Column scite2 has been dropped since nulls percentage is 58 %
Column scite3 has been dropped since nulls percentage is 76 %
Column related has been dropped since nulls percentage is 86 %
```

In [6]:
```python
#NUMERICAL VALUES
dataset.shape
```

Out[6]: (181691, 48)

In [7]:
```python
drop('eventid')
drop('country')
drop('region')
drop('attacktype1')
drop('targtype1')
drop('targsubtype1')
drop('natlty1')
drop('weaptype1')
drop('weapsubtype1')
```

In [8]:
```python
dataset.head()
```

Out[8]:

	iyear	imonth	iday	extended	country_txt	region_txt	provstate	city	latitude	longitude	...	weapsubtype1_txt	nkill	nwound	property	ishostkid	dbsource	INT_LOG
0	1970	7	2	0	Dominican Republic	Central America & Caribbean	NaN	Santo Domingo	18.456792	-69.951164	...	NaN	1.0	0.0	0	0.0	PGIS	0
1	1970	0	0	0	Mexico	North America	Federal	Mexico city	19.371887	-99.086624	...	NaN	0.0	0.0	0	1.0	PGIS	0
2	1970	1	0	0	Philippines	Southeast Asia	Tarlac	Unknown	15.478598	120.599741	...	NaN	1.0	0.0	0	0.0	PGIS	-9
3	1970	1	0	0	Greece	Western Europe	Attica	Athens	37.997490	23.762728	...	Unknown Explosive Type	NaN	NaN	1	0.0	PGIS	-9
4	1970	1	0	0	Japan	East Asia	Fukouka	Fukouka	33.580412	130.396361	...	NaN	NaN	NaN	1	0.0	PGIS	-9

5 rows × 39 columns

In [9]:
```python
unique_all(False)
```

```
Length of unique data for   iyear   is    47
Length of unique data for   imonth   is    13
Length of unique data for   iday   is    32
Length of unique data for   extended   is    2
Length of unique data for   country_txt   is    205
Length of unique data for   region_txt   is    12
Length of unique data for   provstate   is    2856
Length of unique data for   city   is    36675
Length of unique data for   latitude   is    48323
Length of unique data for   longitude   is    48040
Length of unique data for   specificity   is    6
Length of unique data for   vicinity   is    3
Length of unique data for   crit1   is    2
Length of unique data for   crit2   is    2
Length of unique data for   crit3   is    2
Length of unique data for   doubtterr   is    4
Length of unique data for   multiple   is    3
Length of unique data for   success   is    2
Length of unique data for   suicide   is    2
Length of unique data for   attacktype1_txt   is    9
Length of unique data for   targtype1_txt   is    22
Length of unique data for   targsubtype1_txt   is    113
Length of unique data for   corp1   is    33239
Length of unique data for   target1   is    86007
Length of unique data for   natlty1_txt   is    216
Length of unique data for   gname   is    3537
Length of unique data for   guncertain1   is    3
Length of unique data for   individual   is    2
Length of unique data for   weaptype1_txt   is    12
Length of unique data for   weapsubtype1_txt   is    31
Length of unique data for   nkill   is    206
Length of unique data for   nwound   is    239
Length of unique data for   property   is    3
Length of unique data for   ishostkid   is    4
Length of unique data for   dbsource   is    26
Length of unique data for   INT_LOG   is    3
Length of unique data for   INT_IDEO   is    3
Length of unique data for   INT_MISC   is    3
Length of unique data for   INT_ANY   is    3
```

In [10]:
```python
count_nulls()
```

```
Column provstate has been number of nulls 421
Column city has been number of nulls 434
Column latitude has been number of nulls 4556
Column longitude has been number of nulls 4557
Column specificity has been number of nulls 6
Column doubtterr has been number of nulls 1
Column multiple has been number of nulls 1
Column targsubtype1_txt has been number of nulls 10373
Column corp1 has been number of nulls 42550
Column target1 has been number of nulls 636
Column natlty1_txt has been number of nulls 1559
Column guncertain1 has been number of nulls 380
Column weapsubtype1_txt has been number of nulls 20768
Column nkill has been number of nulls 10313
Column nwound has been number of nulls 16311
Column ishostkid has been number of nulls 178
```

In [11]:
```python
unique('iyear')
```

```
Number of unique vaure are 47 which are :
 [1970, 1971, 1972, 1973, 1974, 1975, 1976, 1977, 1978, 1979, 1980, 1981, 1986, 1982, 1983, 1984, 1985, 1987, 1988, 1989, 1990, 1991, 1992, 1994, 1
995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017]
```

In [12]:
```python
unique('imonth')
```

```
Number of unique vaure are 13 which are :
 [7, 0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12]
```

In [13]:
```python
dataset.shape[0]
```

Out[13]: 181691

In [14]:
```python
remove_zero('imonth')
dataset.shape[0]
```

Out[14]: 181671

In [15]:
```python
unique('imonth')
```

```
Number of unique vaure are 12 which are :
 [7, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12]
```

In [16]:
```python
unique('iday')
remove_zero('iday')
unique('iday')
dataset.shape[0]
```

```
Number of unique vaure are 32 which are :
 [2, 0, 1, 3, 6, 8, 9, 10, 11, 12, 13, 14, 15, 19, 20, 21, 22, 25, 26, 27, 28, 30, 31, 4, 7, 16, 17, 18, 23, 24, 5, 29]
Number of unique vaure are 31 which are :
 [2, 1, 3, 6, 8, 9, 10, 11, 12, 13, 14, 15, 19, 20, 21, 22, 25, 26, 27, 28, 30, 31, 4, 7, 16, 17, 18, 23, 24, 5, 29]
```

Out[16]: 180800

In [17]:
```python
unique('extended')
```

```
Number of unique vaure are 2 which are :
 [0, 1]
```

In [18]:
```python
dataset['latitude'].isna().sum()
fillna('latitude',0)
remove_zero('latitude')
dataset['latitude'].isna().sum()
dataset.shape[0]
```

Out[18]: 176331

In [19]:
```python
fillna('longitude',0)
remove_zero('longitude')
dataset['longitude'].isna().sum()
dataset.shape[0]
```

Out[19]: 176330

In [20]:
```python
#CATEGORICAL VALUES
dataset['specificity'].isna().sum()
```

Out[20]: 0

In [21]:
```python
dataset['vicinity'].isna().sum()
```

Out[21]: 0

In [22]:
```python
dataset['doubtterr'].isna().sum()
```

Out[22]: 1

In [23]:
```python
unique('doubtterr')
```

```
Number of unique vaure are 4 which are :
 [0.0, 1.0, -9.0, nan]
```

In [24]:
```python
fillna('doubtterr',33)
remove_zero('doubtterr',33)
dataset['doubtterr'].isna().sum()
```

Out[24]: 0

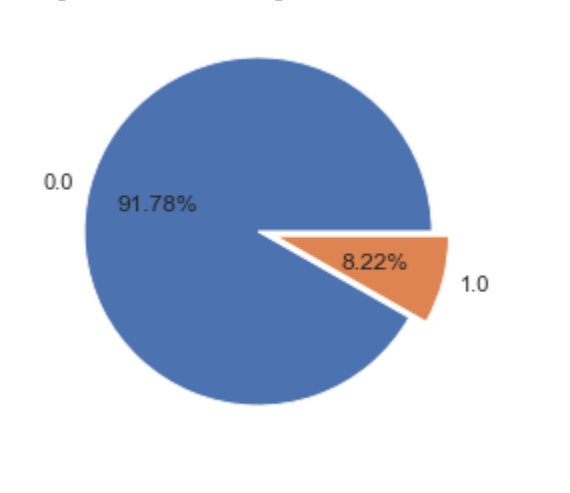
In [25]:
```python
dataset['multiple'].isna().sum()
unique('multiple')
fillna('multiple',33)
remove_zero('multiple',33)
dataset['multiple'].isna().sum()
```

```
Number of unique vaure are 3 which are :
 [0.0, 1.0, nan]
```

Out[25]: 0

In [26]:
```python
dataset['guncertain1'].isna().sum()
unique('guncertain1')
fillna('guncertain1',33)
remove_zero('guncertain1',33)
dataset['guncertain1'].isna().sum()
pie('guncertain1')
```

```
Number of unique vaure are 3 which are :
 [0.0, 1.0, nan]
```



In [27]:
```python
#counting the victims
unique('nkill')
```

```
Number of unique vaure are 202 which are :
 [1.0, 0.0, nan, 7.0, 47.0, 2.0, 5.0, 3.0, 4.0, 25.0, 15.0, 8.0, 26.0, 81.0, 6.0, 9.0, 16.0, 30.0, 31.0, 12.0, 21.0, 14.0, 88.0, 11.0, 10.0, 27.0, 18.0, 22.0, 19.0, 92.0, 13.0, 73.0, 100.0, 42.0, 17.0, 98.0, 422.0, 48.0, 34.0, 54.0, 50.0, 20.0, 41.0, 37.0, 28.0, 40.0, 32.0, 85.0, 23.0, 300.0, 35.0, 60.0, 24.0, 58.0, 87.0, 45.0, 38.0, 29.0, 36.0, 74.0, 83.0, 90.0, 70.0, 66.0, 80.0, 67.0, 51.0, 39.0, 114.0, 124.0, 76.0, 33.0, 75.0, 57.0, 62.0, 46.0, 56.0, 63.0, 120.0, 102.0, 79.0, 52.0, 77.0, 49.0, 111.0, 165.0, 241.0, 108.0, 132.0, 65.0, 43.0, 228.0, 110.0, 180.0, 250.0, 93.0, 200.0, 130.0, 59.0, 94.0, 146.0, 329.0, 44.0, 97.0, 240.0, 227.0, 126.0, 106.0, 388.0, 68.0, 270.0, 84.0, 53.0, 82.0, 171.0, 107.0, 55.0, 112.0, 72.0, 96.0, 140.0, 61.0, 105.0, 115.0, 1180.0, 170.0, 168.0, 121.0, 375.0, 64.0, 91.0, 123.0, 135.0, 256.0, 109.0, 271.0, 206.0, 104.0, 320.0, 224.0, 118.0, 150.0, 129.0, 95.0, 1384.0, 1383.0, 190.0, 119.0, 78.0, 101.0, 71.0, 116.0, 518.0, 344.0, 160.0, 188.0, 103.0, 205.0, 145.0, 153.0, 127.0, 69.0, 141.0, 134.0, 89.0, 400.0, 86.0, 184.0, 210.0, 142.0, 212.0, 287.0, 315.0, 151.0, 670.0, 1570.0, 310.0, 298.0, 953.0, 517.0, 201.0, 122.0, 158.0, 117.0, 144.0, 208.0, 152.0, 230.0, 280.0, 174.0, 143.0, 383.0, 283.0, 154.0, 284.0, 433.0, 266.0, 133.0, 163.0, 128.0, 588.0, 311.0]
```

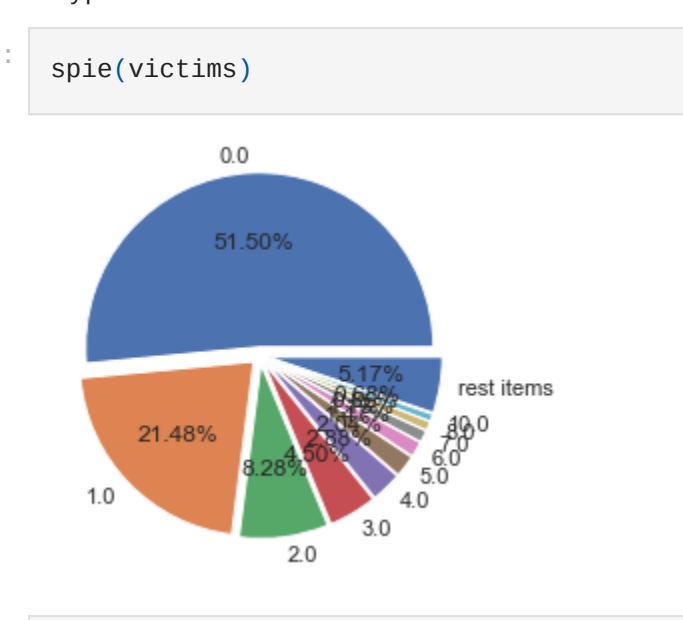
In [28]:
```python
#we cannot use 0 since it's exist indeed, let's choose  999999
fillna('nkill',999999)
remove_zero('nkill',999999)
dataset['nkill'].isna().sum()
```

Out[28]: 0

In [29]:
```python
#victims distribution
victims = max_counts('nkill',10, True)
victims
```

```
Out[29]:
0.0           85666
1.0           35735
2.0           13780
3.0            7477
4.0            4784
5.0            3401
6.0            2439
7.0            1876
8.0            1432
10.0           1134
rest items     8605
dtype: int64
```

In [30]:
```python
spie(victims)
```

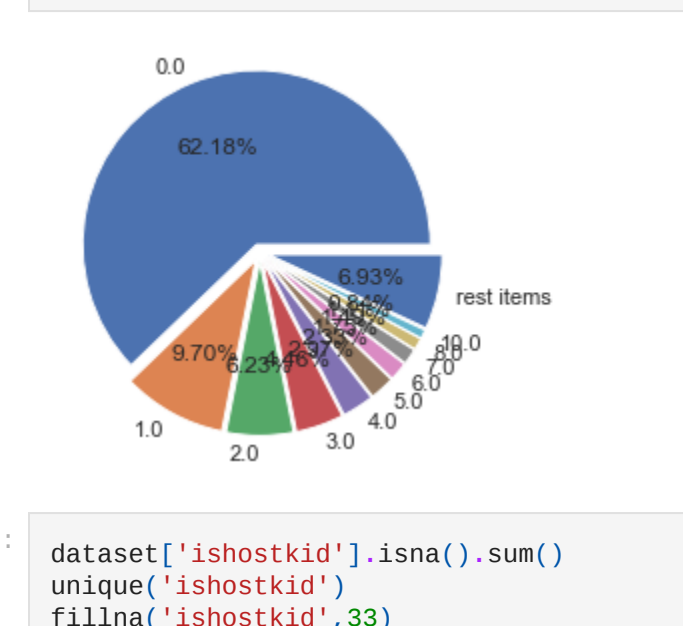


In [31]:
```python
dataset['nwound'].isna().sum()
unique('nwound')
fillna('nwound',999999)
remove_zero('nwound',999999)
dataset['nwound'].isna().sum()
```

```
Number of unique vaure are 239 which are :
 [0.0, 1.0, 2.0, nan, 7.0, 9.0, 17.0, 5.0, 3.0, 10.0, 12.0, 20.0, 4.0, 11.0, 27.0, 13.0, 19.0, 130.0, 56.0, 24.0, 6.0, 72.0, 238.0, 55.0, 15.0, 8.0, 18.0, 50.0, 34.0, 14.0, 70.0, 102.0, 41.0, 48.0, 36.0, 25.0, 81.0, 54.0, 53.0, 100.0, 45.0, 40.0, 62.0, 74.0, 22.0, 31.0, 16.0, 30.0, 285.0, 23.0, 42.0, 43.0, 76.0, 44.0, 52.0, 46.0, 49.0, 160.0, 37.0, 21.0, 60.0, 26.0, 33.0, 28.0, 39.0, 38.0, 35.0, 32.0, 78.0, 80.0, 47.0, 188.0, 215.0, 85.0, 75.0, 161.0, 90.0, 300.0, 95.0, 135.0, 61.0, 138.0, 71.0, 700.0, 66.0, 136.0, 120.0, 217.0, 319.0, 133.0, 68.0, 73.0, 29.0, 751.0, 200.0, 112.0, 250.0, 82.0, 132.0, 230.0, 107.0, 114.0, 84.0, 140.0, 150.0, 51.0, 58.0, 64.0, 295.0, 106.0, 125.0, 69.0, 59.0, 57.0, 122.0, 109.0, 88.0, 115.0, 176.0, 220.0, 97.0, 65.0, 800.0, 173.0, 92.0, 500.0, 236.0, 98.0, 286.0, 141.0, 5500.0, 650.0, 671.0, 86.0, 104.0, 83.0, 101.0, 1272.0, 105.0, 386.0, 391.0, 110.0, 149.0, 89.0, 91.0, 119.0, 77.0, 118.0, 67.0, 170.0, 194.0, 192.0, 227.0, 260.0, 4000.0, 180.0, 184.0, 99.0, 94.0, 137.0, 8190.0, 8191.0, 171.0, 121.0, 103.0, 162.0, 197.0, 151.0, 224.0, 134.0, 233.0, 167.0, 450.0, 216.0, 145.0, 63.0, 727.0, 182.0, 159.0, 108.0, 128.0, 340.0, 163.0, 542.0, 87.0, 155.0, 96.0, 817.0, 257.0, 148.0, 123.0, 246.0, 127.0, 347.0, 178.0, 168.0, 207.0, 183.0, 117.0, 750.0, 147.0, 1001.0, 93.0, 154.0, 235.0, 158.0, 211.0, 116.0, 276.0, 552.0, 360.0, 113.0, 157.0, 175.0, 301.0, 131.0, 400.0, 370.0, 8.5, 169.0, 201.0, 111.0, 143.0, 153.0, 124.0, 79.0, 270.0, 152.0, 296.0, 245.0, 1500.0, 351.0, 600.0, 433.0, 177.0, 491.0, 405.0, 851.0, 316.0]
```

Out[31]: 0

In [32]:
```python
wounded = max_counts('nwound',10, True)
wounded
spie(wounded)
```



In [33]:
```python
dataset['ishostkid'].isna().sum()
unique('ishostkid')
fillna('ishostkid',33)
remove_zero('ishostkid',33)
dataset['ishostkid'].isna().sum()
```

```
Number of unique vaure are 4 which are :
 [0.0, 1.0, nan, -9.0]
```

Out[33]: 0

In [34]:
```python
dataset.shape
```

Out[34]: (159766, 39)

In [35]:
```python
count_nulls()
```

```
Column provstate has been number of nulls 377
Column city has been number of nulls 422
Column targsubtype1_txt has been number of nulls 9478
Column corp1 has been number of nulls 34712
Column target1 has been number of nulls 525
Column natlty1_txt has been number of nulls 1445
Column weapsubtype1_txt has been number of nulls 16078
```

In [36]:
```python
##since we have hundreds of nulls in these 7 features , let's fill them with 'other' to make them ready for label encoder
fillna('provstate','other')
fillna('city','other')
fillna('targsubtype1_txt','other')
fillna('corp1','other')
fillna('target1','other')
fillna('natlty1_txt','other')
fillna('weapsubtype1_txt','other')
```

In [37]:
```python
#Now we can label encode all categorical features & drop original features
count_nulls()
```

In [38]:
```python
encoder('provstate','provstate_code',True)
encoder('city','city_code',True)
encoder('targsubtype1_txt','targsubtype_code',True)
encoder('corp1','corp_code',True)
encoder('target1','target_code',True)
encoder('natlty1_txt','natlty_code',True)
encoder('weapsubtype1_txt','weapsubtype_code',True)

encoder('country_txt','country_code',True)
encoder('region_txt','region_code',True)
encoder('attacktype1_txt','attacktype_code',True)
encoder('targtype1_txt','targtype_code',True)
encoder('gname','gname_code',True)
encoder('weaptype1_txt','weaptype_code',True)
encoder('dbsource','dbsource_code',True)
```

In [39]:
```python
dataset.head()
```

Out[39]:

	iyear	imonth	iday	extended	latitude	longitude	specificity	vicinity	crit1	crit2	...	target_code	natlty_code	weapsubtype_code	country_code	region_code	attacktype_code
0	1970	7	2	0	18.456792	-69.951164	1.0	0	1	1	...	35441	49	30	45	1	1
5	1970	1	1	0	37.005105	-89.176269	1.0	0	1	1	...	13988	197	27	186	6	0
6	1970	1	2	0	-34.891151	-56.187214	1.0	0	1	1	...	35251	198	1	187	7	1
7	1970	1	2	0	37.791927	-122.225906	1.0	0	1	1	...	23515	197	26	186	6	2
8	1970	1	2	0	43.076592	-89.412488	1.0	0	1	1	...	50337	197	12	186	6	3

5 rows × 39 columns

In [40]:
```python
#Now we can see that our data has no null values.
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 159766 entries, 0 to 181690
Data columns (total 39 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   iyear             159766 non-null  int64
 1   imonth            159766 non-null  int64
 2   iday              159766 non-null  int64
 3   extended          159766 non-null  int64
 4   latitude          159766 non-null  float64
 5   longitude         159766 non-null  float64
 6   specificity       159766 non-null  float64
 7   vicinity          159766 non-null  int64
 8   crit1             159766 non-null  int64
 9   crit2             159766 non-null  int64
 10  crit3             159766 non-null  int64
 11  doubtterr         159766 non-null  float64
 12  multiple          159766 non-null  float64
 13  success           159766 non-null  int64
 14  suicide           159766 non-null  int64
 15  guncertain1       159766 non-null  float64
 16  individual        159766 non-null  int64
 17  nkill             159766 non-null  float64
 18  nwound            159766 non-null  float64
 19  property          159766 non-null  int64
 20  ishostkid         159766 non-null  float64
 21  INT_LOG           159766 non-null  int64
 22  INT_IDEO          159766 non-null  int64
 23  INT_MISC          159766 non-null  int64
 24  INT_ANY           159766 non-null  int64
 25  provstate_code    159766 non-null  int32
 26  city_code         159766 non-null  int32
 27  targsubtype_code  159766 non-null  int32
 28  corp_code         159766 non-null  int32
 29  target_code       159766 non-null  int32
 30  natlty_code       159766 non-null  int32
 31  weapsubtype_code  159766 non-null  int32
 32  country_code      159766 non-null  int32
 33  region_code       159766 non-null  int32
 34  attacktype_code   159766 non-null  int32
 35  targtype_code     159766 non-null  int32
 36  gname_code        159766 non-null  int32
 37  weaptype_code     159766 non-null  int32
 38  dbsource_code     159766 non-null  int32
dtypes: float64(9), int32(14), int64(16)
memory usage: 40.2 MB
```

In [41]:
```python
#DATA SPLITTING
X , y = make_xy('success')
```

In [42]:
```python
X.shape
```

Out[42]: (159766, 38)

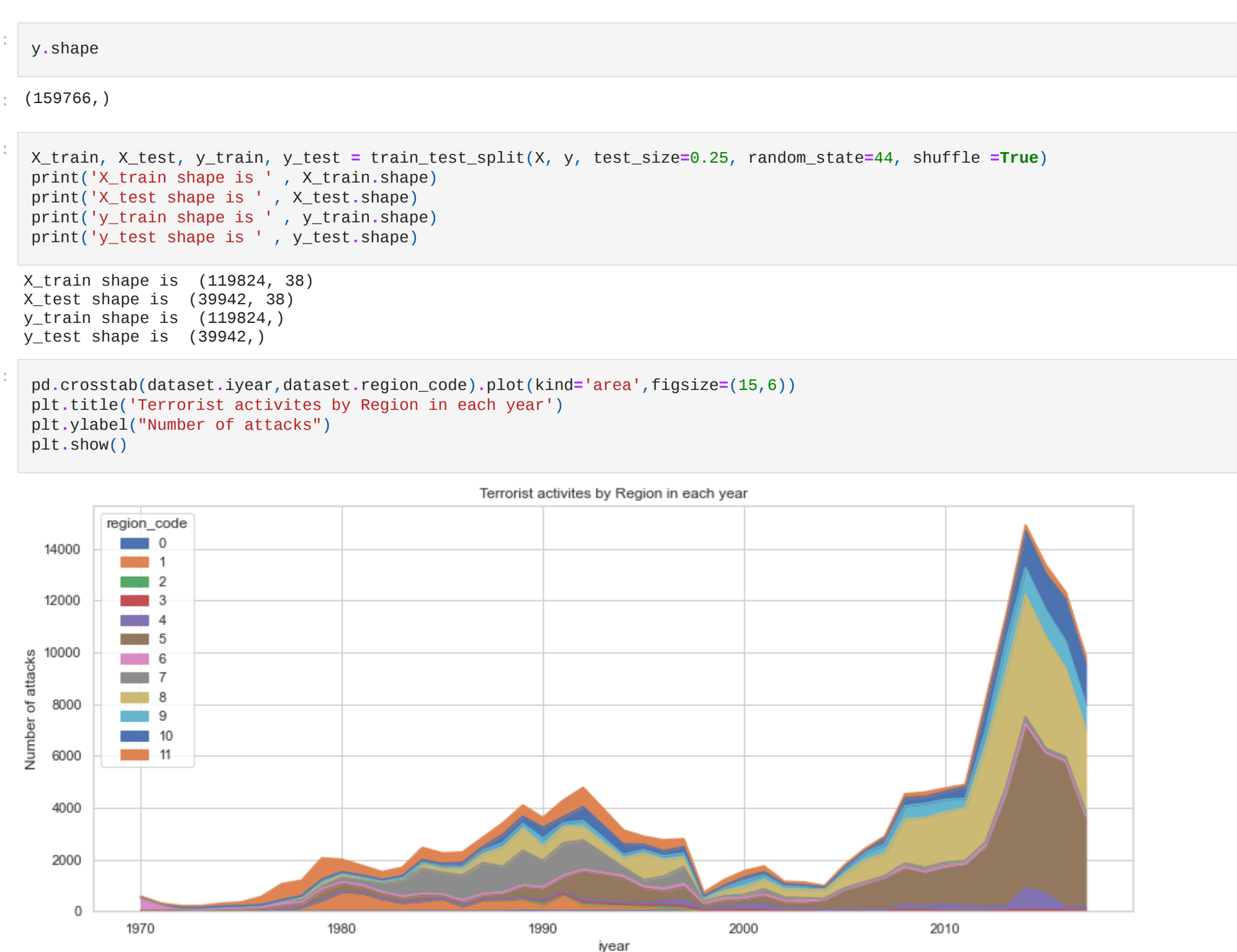
In [43]:
```python
y.shape
```

Out[43]: (159766,)

In [44]:
```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=44, shuffle =True)
print('X_train shape is ' , X_train.shape)
print('X_test shape is ' , X_test.shape)
print('y_train shape is ' , y_train.shape)
print('y_test shape is ' , y_test.shape)
```

```
X_train shape is  (119824, 38)
X_test shape is  (39942, 38)
y_train shape is  (119824,)
y_test shape is  (39942,)
```

In [45]:
```python
pd.crosstab(dataset.iyear,dataset.region_code).plot(kind='area',figsize=(15,6))
plt.title('Terrorist activites by Region in each year')
plt.ylabel("Number of attacks")
plt.show()
```



In [46]:
```python
from sklearn.ensemble import GradientBoostingClassifier

GBCModel = GradientBoostingClassifier(n_estimators=100,max_depth=3,random_state=33)
GBCModel.fit(X_train, y_train)
```

Out[46]: GradientBoostingClassifier(random_state=33)

In [47]:
```python
print('GBCModel Train Score is : ' , GBCModel.score(X_train, y_train))
print('GBCModel Test Score is : ' , GBCModel.score(X_test, y_test))
```

```
GBCModel Train Score is :  0.9472392842836159
GBCModel Test Score is :  0.9455460417605528
```

In [48]:
```python
y_pred = GBCModel.predict(X_test)
y_pred_prob = GBCModel.predict_proba(X_test)
print('Predicted Value for GBCModel is : ' , y_pred[:10])
print('Prediction Probabilities Value for GBCModel is : ' , y_pred_prob[:10])
```

```
Predicted Value for GBCModel is :  [0 1 1 1 1 1 1 1 1 0]
Prediction Probabilities Value for GBCModel is :  [[0.53389237 0.46610763]
 [0.01341194 0.98658806]
 [0.013064   0.986936  ]
 [0.04913287 0.95086713]
 [0.03383446 0.96616554]
 [0.26060571 0.73939429]
 [0.07396851 0.92603149]
 [0.04155095 0.95844905]
 [0.05149775 0.94850225]
 [0.89478621 0.10521379]]
```

In [49]:
```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=0)
```

In [50]:
```python
def cross_validation_split(dataset, n_folds):
    dataset_split = list()
    dataset_copy = list(dataset)
    fold_size = int(len(dataset) / n_folds)
    for _ in range(n_folds):
        fold = list()
        while len(fold) < fold_size:
            index = randrange(len(dataset_copy))
            fold.append(dataset_copy.pop(index))
        dataset_split.append(fold)
    return dataset_split
```

In [51]:
```python
def accuracy_metric(actual, predicted):
    correct = 0
    for i in range(len(actual)):
        if actual[i] == predicted[i]:
            correct += 1
    return correct / float(len(actual)) * 100.0
```

In [52]:
```python
dataset.shape
Train, Test = train_test_split(dataset, test_size=0.2, random_state = 123)
Train.shape
```

Out[52]: (127812, 39)

In [53]:
```python
round(Train.shape[0]/dataset.shape[0],1)
```

Out[53]: 0.8

In [54]:
```python
Train.index[1:10]
```

```
Out[54]: Int64Index([152986, 100156, 104426, 178894, 95929, 121558, 107347, 37311,
            27647],
           dtype='int64')
```

In [55]:
```python
DT1 = DecisionTreeClassifier(random_state=100, min_samples_leaf=50)
DT1_Model = DT1.fit(X_train, y_train)
Test_Pred_DT = DT1_Model.predict(X_test)
```

In [56]:
```python
Confusion_Mat_DT = confusion_matrix(y_test, Test_Pred_DT)
Confusion_Mat_DT
```

```
Out[56]: array([[ 6583,  2912],
       [ 1279, 69109]], dtype=int64)
```

In [57]:
```python
print(accuracy_score(y_test, Test_Pred_DT))
print(classification_report(y_test, Test_Pred_DT))
```

```
0.9475357710651828
              precision    recall  f1-score   support

           0       0.84      0.69      0.76      9495
           1       0.96      0.98      0.97     70388

    accuracy                           0.95     79883
   macro avg       0.90      0.84      0.86     79883
weighted avg       0.95      0.95      0.95     79883
```

In [ ]:

In [ ]:

In [ ]: