In [18]:

```
import calendar
calendar.month_name[1]
```

Out[18]:

```
'January'
```

In [19]:

```
webPageUrl =requests.get('https://en.wikipedia.org/wiki/List_of_terrorist_incidents')
allUrlList = soup(webPageUrl.text,'html.parser')
allUrl = allUrlList.find_all('div',attrs={'class':'div-col columns column-width'})
listOfTerroristIncidentsWrtTime=pd.DataFrame()
for each in allUrl:
    j= each.find_all('a',attrs={'href':True})
    for elem in j:
        listOfTerroristIncidentsWrtTime = pd.concat([listOfTerroristIncidentsWrtTime,
                                                     pd.Series('https://en.wikipedia.org/'+elem['href'])],axis=0)
    break
listOfTerroristIncidentsWrtTime.columns=['Year_of_Terrorist_attack']
listOfTerroristIncidentsWrtTime.reset_index(drop=True,inplace=True)
```

In [20]:

```
listOfTerroristIncidentsWrtTime.sort_index(axis=0,ascending=False,inplace=True)
listOfTerroristIncidentsWrtTime.head()
```

Out[20]:

	Year_of_Terrorist_attack
100	https://en.wikipedia.org//wiki/List_of_terrorist_incidents_in_April_2019
99	https://en.wikipedia.org//wiki/List_of_terrorist_incidents_in_March_2019
98	https://en.wikipedia.org//wiki/List_of_terrorist_incidents_in_February_2019
97	https://en.wikipedia.org//wiki/List_of_terrorist_incidents_in_January_2019
96	https://en.wikipedia.org//wiki/List_of_terrorist_incidents_in_December_2018

In [21]:

```
def func(x):
    newX=''
    if str(x).find('(') >0:
        #print('ssss: ',x)
        newX= str(x).replace('(s)','s')
        #print('newX: ',newX)
    elif str(x) == 'Perpetrator':
        newX = 'Perpetrators'
    elif str(x) == 'Deaths':
        newX = 'Dead'
    elif str(x) == 'Injuries':
        newX = 'Injured'
    elif str(x) == 'Dates':
        newX = 'Date'
    else:
        newX = x
    return newX
```

In [22]:

```python
def tagNames(tags,each):
    j=pd.DataFrame()
    k=pd.DataFrame()
    h= requests.get(each)
    weburl= soup(h.text,'html.parser')
    headersList = pd.Series()
    for each in tags:
        thList=each.find_all('th')
    for elem in thList:
        headersList = headersList.append(pd.Series(elem.text.strip()))
    headersList = headersList.unique()
    headersList = pd.Series(headersList).apply(lambda x : func(x))
    headersList = headersList.unique()
    for each in tags:
        for elem in each.find_all('tr'):
            for each1 in elem.find_all('td'):
                j = pd.concat([j,pd.Series(each1.text.strip())],axis=1)
                if j.shape[1] == headersList.size:
                    break
            while j.shape[1] <  headersList.size:
                j = pd.concat([j,pd.Series(' ')],axis=1)
            k =pd.concat([k,j],axis=0)
            j=pd.DataFrame()
    k.columns=[headersList]
    k = k.iloc[1:,:]
    return(k)

```

In [23]:

```python
mergedDataFrame=pd.DataFrame()
for each in listOfTerroristIncidentsWrtTime.Year_of_Terrorist_attack:
        #print(each)
        h= requests.get(each)
        weburl= soup(h.text,'html.parser')
        tags = weburl.find_all('table',attrs={'class':'wikitable sortable'})
        tags1 = weburl.find_all('table',attrs={'class':'wikitable'})
        if len(tags) == 0:
            tags = tags1
        allTagNames= tagNames(tags,each)
        mergedDataFrame = pd.concat([mergedDataFrame,allTagNames],axis=0,ignore_index=True)
#display(mergedDataFrame)


```

In [24]:

```python
newDTFrame = mergedDataFrame.reset_index(drop=True)
newDTFrame = mergedDataFrame[['Date','Type','Dead','Injured','Location','Details','Description','Location and des
                         'Perpetrators','Part of','Non-state','State']]

```

In [25]:

```python
mergedDataFrame.columns.values
```

Out[25]:

```
array([('Date',), ('Dead',), ('Description',), ('Details',), ('Injured',),
       ('Location',), ('Location and description',), ('Non-state',),
       ('Part of',), ('Perpetrators',), ('State',), ('Type',)],
      dtype=object)
```

In [36]:

```python
TerrorismDTFrame = newDTFrame
```

In [37]:

```python
def func(x):
    if str(x) == ' ':
        x=np.nan
    return x
TerrorismDTFrame = TerrorismDTFrame.applymap(lambda x : func(x))
```

In [38]:
```
TerrorismDTFrame
```

Out[38]:

	Date	Type	Dead	Injured	Location	Details	Description	Location and description	Perpetrators
0	1	Bombing	0	2	Diyala Governorate, Iraq	Two Iraqi soldiers were wounded when an explosive device targeted their vehicle in the Sheikh Ta...	NaN	NaN	Islamic State (suspected
1	1	Arson, looting	0	0	Chibok, Nigeria	Boko Haram militants set on fire several houses and looted a village in the area of Chibok.[2]	NaN	NaN	Boko Haran
						Five people including 3			

In [39]:
```
TerrorismDTFrame = TerrorismDTFrame.dropna(how='all')
```

In [40]:
```
TerrorismDTFrame = TerrorismDTFrame.drop(columns=['Non-state','State'])
TerrorismDTFrame= TerrorismDTFrame.reset_index(drop=True)
```

```
C:\Users\supratik chanda\Anaconda3\lib\site-packages\pandas\core\generic.py:3111: PerformanceWarning: dropping on a n
on-lexsorted multi-index without a level parameter may impact performance.
  obj = obj._drop_axis(labels, axis, level=level, errors=errors)
```

In [33]:
```
TerrorismDTFrame.to_csv('C:\\Users\\supratik chanda\\Documents\\All Docs\\TerrorismNew.csv',index=False,header=Tru
```

In [336]:
```
mergedDTframe = pd.read_csv('C:\\Users\\supratik chanda\\Documents\\All Docs\\TerrorismNew.csv')
```

In [337]:
```
mergedDTframe.shape
```

Out[337]:
```
(10466, 10)
```

In [338]:
```
mergedDTframe[mergedDTframe['Location and description'].notna()].shape
```

Out[338]:
```
(813, 10)
```

In [339]:
```
import pycountry
countryList=[]
for each in pycountry.countries:
    countryList.append(each.name)

```

In [340]:
```
countryList.append('Iran')
countryList.append('Russia')
countryList.append('Syria')
countryList.append('Gaza City')
countryList.append('Bolivia')
countryList.append('Bosnia')
countryList.append('West Bank')
countryList.append('Rafah')
countryList.append('Ivory Coast')
countryList.append('Kedumim')
countryList.append('Transnistria')
if 'Baghdad' in countryList:
    print('yes')
countryList[0:5]
```

Out[340]:
```
['Aruba', 'Afghanistan', 'Angola', 'Anguilla', 'Åland Islands']
```

In [341]:
```
# Creating a panda series Location New with all the names of the country taken from the column : Location and desc
```

In [342]:

```
LocationNew=pd.Series()
j=0
i=0
import re
for elem in mergedDTframe['Location and description']:
    if str(elem) == 'nan':
        LocationNew = LocationNew.append(pd.Series(np.nan))
    if str(elem) != 'nan':
        tempStr =' '.join(str(elem).split(' ')[0:15])
        flag=False
        for each in countryList:
            if re.search(each,tempStr) != None:
                #print('now: ')
                flag=True
                j+=1
                LocationNew = LocationNew.append(pd.Series((re.search(each,tempStr).group())))
                #print(tempStr,'\n---------')
                break
        if flag == False:
            print(tempStr)
print(j)
```

```
813
```

In [343]:

```
print(LocationNew.shape)

```

```
(10466,)
```

In [344]:

```
mergedDTframe[mergedDTframe['Location and description'].notna()].shape
```

Out[344]:

```
(813, 10)
```

In [345]:

```
mergedDTframe[mergedDTframe['Location and description'].isna()].shape
```

Out[345]:

```
(9653, 10)
```

In [346]:

```
#Creating a new dataframe: LocationNewDataFrame, that contains all the values of LocationNew panda series
```

In [347]:

```
LocationNewDataFrame = pd.DataFrame(LocationNew)
```

In [348]:

```
LocationNewDataFrame.reset_index(drop=True,inplace=True)
LocationNewDataFrame.head()
```

Out[348]:

	0
0	NaN
1	NaN
2	NaN
3	NaN
4	NaN

In [349]:

```
# Merging the new dataframe: LocationNewDataFrame to the old dataframe : mergedDTframe. Renaming the new column as
# LocationNew
```

In [350]:

```
mergedDTframe = pd.concat([mergedDTframe,LocationNewDataFrame],axis=1)
mergedDTframe = mergedDTframe.rename(columns={0:'LocationNew'})
```

In [351]:

```
# Checking whether there is any such row where Column : mergedDTframe and Location have NaN value
```

In [352]:

```
mergedDTframe[(mergedDTframe.LocationNew.isna()) & (mergedDTframe.Location.isna())]
```

Out[352]:

Date	Dead	Description	Details	Injured	Location	Location and description	Part of	Perpetrators	Type	LocationNew

In [353]:

```
# dropping all the rows having NaN in all columns and reseting  the index from 0
```

In [354]:

```
mergedDTframe=mergedDTframe.dropna(how='all',axis=0)
```

In [355]:

```python
mergedDTframe.reset_index(drop=True,inplace=True)
mergedDTframe.head()
```

Out[355]:

	Date	Dead	Description	Details	Injured	Location	Location and description	Part of	Perpetrators	Type	LocationNew
0	1	0	NaN	Two Iraqi soldiers were wounded when an explosive device targeted their vehicle in the Sheikh Ta...	2	Diyala Governorate, Iraq	NaN	Iraqi insurgency	Islamic State (suspected)	Bombing	NaN
1	1	0	NaN	Boko Haram militants set on fire several houses and looted a village in the area of Chibok.[2]	0	Chibok, Nigeria	NaN	Boko Haram insurgency	Boko Haram	Arson, looting	NaN
2	1	5	NaN	Five people including 3 civilian militia members were executed and filmed on video by the Islami...	0	Borno State, Nigeria	NaN	Boko Haram insurgency	Boko Haram	Execution	NaN
3	1	8	NaN	Eight policemen were killed in a Taliban attack on a checkpoint in the Sholgara District of the ...	5	Sholgara district, Afghanistan	NaN	War in Afghanistan	Taliban	Shooting	NaN
4	2	1	NaN	A French military doctor was killed and two others were injured when an explosive device detonat...	2	Gourma-Rharous Cercle, Mali	NaN	Northern Mali Conflict	Militants (suspected)	Bombing	NaN

In [356]:

```python
# Creating a new column : LocationMerged whose values are the merged values of columns: Location and LocationNew
```

In [357]:

```python
mergedDTframe['LocationMerged'] = mergedDTframe['Location'].combine_first(mergedDTframe['LocationNew'])
```

In [358]:

```python
mergedDTframe.drop(columns=['LocationNew','Location'],inplace=True)
```

In [359]:

```python
# Checking any row present where LocationMerged not = np.nan
```

In [360]:

```python
mergedDTframe[(mergedDTframe.LocationMerged.isna())].shape[0]
```

Out[360]:

```
0
```

In [361]:

```python
mergedDTframe.head()
```

Out[361]:

	Date	Dead	Description	Details	Injured	Location and description	Part of	Perpetrators	Type	LocationMerged
0	1	0	NaN	Two Iraqi soldiers were wounded when an explosive device targeted their vehicle in the Sheikh Ta...	2	NaN	Iraqi insurgency	Islamic State (suspected)	Bombing	Diyala Governorate, Iraq
1	1	0	NaN	Boko Haram militants set on fire several houses and looted a village in the area of Chibok.[2]	0	NaN	Boko Haram insurgency	Boko Haram	Arson, looting	Chibok, Nigeria
2	1	5	NaN	Five people including 3 civilian militia members were executed and filmed on video by the Islami...	0	NaN	Boko Haram insurgency	Boko Haram	Execution	Borno State, Nigeria
3	1	8	NaN	Eight policemen were killed in a Taliban attack on a checkpoint in the Sholgara District of the ...	5	NaN	War in Afghanistan	Taliban	Shooting	Sholgara district, Afghanistan
4	2	1	NaN	A French military doctor was killed and two others were injured when an explosive device detonat...	2	NaN	Northern Mali Conflict	Militants (suspected)	Bombing	Gourma-Rharous Cercle, Mali

In [362]:

```python
mergedDTframe['tempDTDescFrame'] = mergedDTframe['Description'].combine_first(mergedDTframe['Details'])
```

In [363]:

```python
mergedDTframe['FinalDescFrame'] = mergedDTframe['tempDTDescFrame'].combine_first(mergedDTframe['Location and desc
```

In [364]:

```
mergedDTframe = mergedDTframe.drop(columns=['Description','Details','Location and description'])
mergedDTframe = mergedDTframe.drop(columns=['tempDTDescFrame'])
mergedDTframe.head()
```

Out[364]:

	Date	Dead	Injured	Part of	Perpetrators	Type	LocationMerged	FinalDescFrame
0	1	0	2	Iraqi insurgency	Islamic State (suspected)	Bombing	Diyala Governorate, Iraq	Two Iraqi soldiers were wounded when an explosive device targeted their vehicle in the Sheikh Ta...
1	1	0	0	Boko Haram insurgency	Boko Haram	Arson, looting	Chibok, Nigeria	Boko Haram militants set on fire several houses and looted a village in the area of Chibok.[2]
2	1	5	0	Boko Haram insurgency	Boko Haram	Execution	Borno State, Nigeria	Five people including 3 civilian militia members were executed and filmed on video by the Islami...
3	1	8	5	War in Afghanistan	Taliban	Shooting	Sholgara district, Afghanistan	Eight policemen were killed in a Taliban attack on a checkpoint in the Sholgara District of the ...
4	2	1	2	Northern Mali Conflict	Militants (suspected)	Bombing	Gourma-Rharous Cercle, Mali	A French military doctor was killed and two others were injured when an explosive device detonat...

In [365]:

```
mergedDTframe.FinalDescFrame.isna().sum()
```

Out[365]:

```
116
```

In [366]:

```
mergedDTframe = mergedDTframe[mergedDTframe.FinalDescFrame.notna()]
```

In [367]:

```
mergedDTframe.FinalDescFrame.isna().sum()
```

Out[367]:

```
0
```

In [368]:

```
# Creating a new dataframe:"Description", with only description extracted from column: "FinalDescFrame"
```

In [369]:

```
def ExtractDescOnly(x):
    DescStr=''
    flag = False
    for each in countryList:
            if each in str(x):
                flag = True
                splittedStr = str(x).split(each,1)[1]
                temp = re.sub('\[.*\]','',splittedStr)
                Desc = re.sub('[^a-zA-Z0-9\s]','',temp)
                DescStr = Desc
    if flag == False:
        DescStr = x
    return(DescStr)

```

In [370]:

```
mergedDTframe['Description'] = mergedDTframe['FinalDescFrame'].apply(lambda x: ExtractDescOnly(x))
```

In [371]:

```
# Checking the shape of new mergedDTframe['Description']
```

In [372]:

```
mergedDTframe[mergedDTframe['Description'].notna()].shape
```

Out[372]:

```
(10350, 9)
```

In [373]:

```
mergedDTframe[mergedDTframe['Description'].isna()].shape
```

Out[373]:

```
(0, 9)
```

In [374]:

```
mergedDTframe = mergedDTframe.drop(columns=['Part of','FinalDescFrame'])
mergedDTframe.head(3)
```

Out[374]:

	Date	Dead	Injured	Perpetrators	Type	LocationMerged	Description
0	1	0	2	Islamic State (suspected)	Bombing	Diyala Governorate, Iraq	i soldiers were wounded when an explosive device targeted their vehicle in the Sheikh Tami villa...
1	1	0	0	Boko Haram	Arson, looting	Chibok, Nigeria	Boko Haram militants set on fire several houses and looted a village in the area of Chibok.[2]
2	1	5	0	Boko Haram	Execution	Borno State, Nigeria	Five people including 3 civilian militia members were executed and filmed on video by the Islami...

In [375]:

```python
listOfCountry = ['Tanzania','Burma','Gaza','Laos','South Korea','Vietnam','Bangkok','Kosovo','Jerusalem','Aghanis
for each in listOfCountry:
    countryList.append(each)
def CountryExtract(loc):
    for each in countryList:
            if re.search(each,loc) != None:
                return each
```

In [376]:

```python
mergedDTframe['Location'] = mergedDTframe['LocationMerged'].apply(lambda x : CountryExtract(x))
```

In [377]:

```python
mergedDTframe = mergedDTframe[mergedDTframe.Location.notna()]
```

In [391]:

```python
mergedDTframe.Description.isnull().sum()
```

Out[391]:

```
0
```

In [389]:

```python
finalDTframe = mergedDTframe.drop(columns=['LocationMerged'],axis=1).reset_index(drop=True)
finalDTframe = finalDTframe.applymap(lambda x: re.sub('\[.*\]|\(.*\)|\+|~|','',str(x)))
finalDTframe.Type = finalDTframe.Type.apply(lambda x: str(x).split(',')[0])
finalDTframe.head(10)
```

Out[389]:

	Date	Dead	Injured	Perpetrators	Type	Description	Location
0	1	0	2	Islamic State	Bombing	i soldiers were wounded when an explosive device targeted their vehicle in the Sheikh Tami villa...	Iraq
1	1	0	0	Boko Haram	Arson	Boko Haram militants set on fire several houses and looted a village in the area of Chibok.	Niger
2	1	5	0	Boko Haram	Execution	Five people including 3 civilian militia members were executed and filmed on video by the Islami...	Niger
3	1	8	5	Taliban	Shooting	Eight policemen were killed in a Taliban attack on a checkpoint in the Sholgara District of the ...	Afghanistan
4	2	1	2	Militants	Bombing	A French military doctor was killed and two others were injured when an explosive device detonat...	Mali
5	2	2	1	Unknown	Bombing	i soldiers were killed and a number of others were injured when their vehicle struck a roadside ...	Yemen
6	3	1	3	Islamic State	Shooting	A policeman was killed and 3 others were left wounded in an attack by militants in the Jalawla d...	Iraq
7	3	62	0	Militants	Shootings		Burkina Faso
8	3	1	0	Militants	Shooting	In Shalipora village of Kulgam District, Abdul Majeed, a panch, was shot and injured by suspecte...	India
9	3	30	22	Taliban	Shooting	Some early reports also said that air support also caused heavy casualties to the Taliban durin...	Afghanistan

In [380]:

```python
finalDTframe.to_csv('C:\\Users\\supratik chanda\\Documents\\All Docs\\CleansedTerrorismData.csv',index=False)
```

In [381]:

```python
cleansedDTframe = pd.read_csv('C:\\Users\\supratik chanda\\Documents\\All Docs\\CleansedTerrorismData.csv')

```

In [382]:

```python
# Removing all the non-digit characters from the dead and injured columns
```

In [383]:

```python
def func(x):
    each=re.sub('[a-zA-Z]','',str(x)).strip()
    elem = re.sub('[^0-9]','',str(each)).strip()
    return elem


```

In [384]:

```python
cleansedDTframe[['Injured','Dead']]= cleansedDTframe[['Injured','Dead']].applymap(lambda x : func(x))
```

In [385]:

```python
cleansedDTframe.Injured.shape
```

Out[385]:

```
(10293,)
```

In [386]:

```python
cleansedDTframe.Dead.shape
```

Out[386]:

```
(10293,)
```

In [387]:

```
cleansedDTframe.Perpetrators.isna().sum()
cleansedDTframe = cleansedDTframe[cleansedDTframe.Perpetrators.notna()]
```

In [321]:

```
cleansedDTframe.shape
```

Out[321]:

```
(9089, 7)
```

In [390]:

```
cleansedDTframe.Description.isnull().sum()
```

Out[390]:

```
575
```

In [406]:

```
cleansedDTframe = cleansedDTframe[cleansedDTframe.Description.notnull()]
```

In [407]:

```
cleansedDTframe.to_csv('C:\\Users\\supratik chanda\\Documents\\All Docs\\FinalisedTerrorDataFrame.csv',index=Fals
```

In [ ]:

```

```