DATA VISULIZATION PROJECT - NATURAL DISASTERS

Junnubala Gowtham(20181CSE0295)

Kalangi Sai Nikhil(20181CSE0307)

Karlapudi Praveen(20181CSE0317)

K Sai Rama Kaushik(20181CSE0318)

K Venkata Sai Abhishek(20181CSE0320)

TYPES OF NATURAL DISASTERS

```
#To mount the Colab and google drive
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mour
```

```
#Importing the required libraries
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
sns.set(color_codes=True)
```

```
#Read CSV
df = pd.read_csv("/content/drive/MyDrive/DV/NATURAL DISASTERS.csv")
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8824 entries, 0 to 8823
Data columns (total 19 columns):
```

```
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Year               8824 non-null   int64  
 1   Disaster Subgroup  8824 non-null   object 
 2   Disaster Type      8824 non-null   object 
 3   Disaster Subtype   8824 non-null   object 
 4   Country            8824 non-null   object 
 5   ISO                8824 non-null   object 
 6   Region             8824 non-null   object 
 7   Location           8535 non-null   object 
 8   Start Year         8823 non-null   float64
 9   Start Month        8769 non-null   float64
 10  Start Day          7503 non-null   float64
 11  End Year           8824 non-null   int64  
 12  End Month          8669 non-null   float64
 13  End Day            7544 non-null   float64
 14  Total Deaths       6350 non-null   float64
 15  No Injured         2339 non-null   float64
 16  No Homeless        1172 non-null   float64
 17  Total Affected     6997 non-null   float64
 18  Total Damages      2735 non-null   float64
dtypes: float64(10), int64(2), object(7)
memory usage: 1.3+ MB
None
```

```
#To Remove Unnamed Columns
df = df.loc[:, ~df.columns.str.contains('^Unnamed')]
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8824 entries, 0 to 8823
Data columns (total 19 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Year               8824 non-null   int64  
 1   Disaster Subgroup  8824 non-null   object 
 2   Disaster Type      8824 non-null   object 
 3   Disaster Subtype   8824 non-null   object 
 4   Country            8824 non-null   object 
 5   ISO                8824 non-null   object 
 6   Region             8824 non-null   object 
 7   Location           8535 non-null   object 
 8   Start Year         8823 non-null   float64
 9   Start Month        8769 non-null   float64
 10  Start Day          7503 non-null   float64
 11  End Year           8824 non-null   int64  
 12  End Month          8669 non-null   float64
 13  End Day            7544 non-null   float64
 14  Total Deaths       6350 non-null   float64
 15  No Injured         2339 non-null   float64
 16  No Homeless        1172 non-null   float64
 17  Total Affected     6997 non-null   float64
 18  Total Damages      2735 non-null   float64
dtypes: float64(10), int64(2), object(7)
```

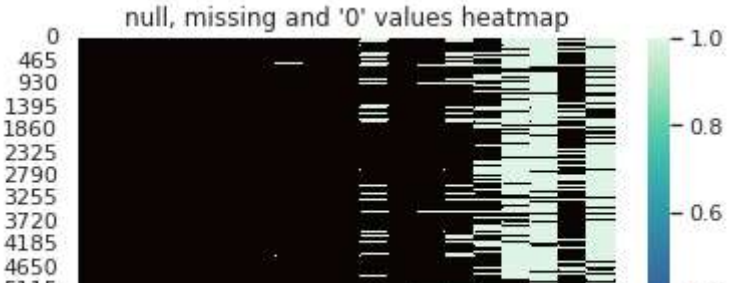
```
memory usage: 1.3+ MB
None
```

```
#Replace Null Values with 0
df.fillna(value = 0,inplace = True)
df.isnull().sum()
df.head()
```

	Year	Disaster Subgroup	Disaster Type	Disaster Subtype	Country	ISO	Region	Location	Start Year	Start Month
0	2000	Hydrological	Landslide	Landslide	Angola	AGO	Middle Africa	Chassuala village (Saurimo district, Lunda-Sul...	2000.0	10.0
1	2000	Biological	Epidemic	Epidemic	Angola	AGO	Middle Africa	Bie, Lunda Sul, Benguela	2000.0	8.0
2	2000	Hydrological	Flood	Flood	Angola	AGO	Middle Africa	Caxito village (Dande district	2000.0	3.0

HEAT MAP

```
plt.title('null, missing and \'0\' values heatmap')
sns.heatmap((df.isnull()) | (df == 0), cmap = 'mako')
plt.show()
print('\nDUPLICATE VALUE COUNT: ', df.duplicated().sum())
```



SWARM PLOT


```
# groupby()- To group df based on Year
y=df.groupby('Year')
y
# Iterate through he Year column and their corresponding df
for Year,Year_df in y:
  print(Year) # Prints Key Value - Year
  print(Year_df) # Prints df corresponding to the key as value
```

```
1900
      Year Disaster Subgroup  ... Total Affected Total Damages
8016  1900       Geophysical  ...            0.0           0.0

[1 rows x 19 columns]
1902
      Year Disaster Subgroup  ... Total Affected Total Damages
6732  1902       Geophysical  ...        17540.0           0.0
6733  1902       Geophysical  ...            0.0           0.0

[2 rows x 19 columns]
1904
      Year Disaster Subgroup  ... Total Affected Total Damages
7347  1904       Geophysical  ...         1620.0           0.0
8019  1904       Geophysical  ...          208.0           0.0
8020  1904       Geophysical  ...         2141.0           0.0

[3 rows x 19 columns]
1905
      Year Disaster Subgroup  ... Total Affected Total Damages
6734  1905       Geophysical  ...            0.0       23800.0
6735  1905       Geophysical  ...            0.0           0.0
7348  1905       Geophysical  ...          177.0           0.0

[3 rows x 19 columns]
1906
      Year Disaster Subgroup  ... Total Affected Total Damages
6736  1906       Geophysical  ...        10000.0           0.0
8021  1906       Geophysical  ...         5466.0           0.0

[2 rows x 19 columns]
1909
```

```
      Year Disaster Subgroup  ... Total Affected Total Damages
6737  1909       Geophysical  ...            0.0           0.0
7349  1909       Geophysical  ...         2928.0           0.0
7350  1909       Geophysical  ...            0.0           0.0
7351  1909       Geophysical  ...            0.0           0.0
8022  1909       Geophysical  ...          417.0           0.0

[5 rows x 19 columns]
1910
      Year Disaster Subgroup  ... Total Affected Total Damages
6738  1910       Geophysical  ...            0.0           0.0
8023  1910       Geophysical  ...            0.0           0.0

[2 rows x 19 columns]
1911
      Year Disaster Subgroup  ... Total Affected Total Damages
7352  1911       Geophysical  ...            0.0           0.0
7353  1911       Geophysical  ...         1266.0           0.0
7354  1911       Geophysical  ...            0.0           0.0
7355  1911       Geophysical  ...           22.0           0.0
7356  1911       Geophysical  ...            0.0           0.0

[5 rows x 19 columns]
1913
      Year Disaster Subgroup  ... Total Affected Total Damages
```

```
# To access a speciic df with key value 2021
gg=y.get_group(2021)
gg
```

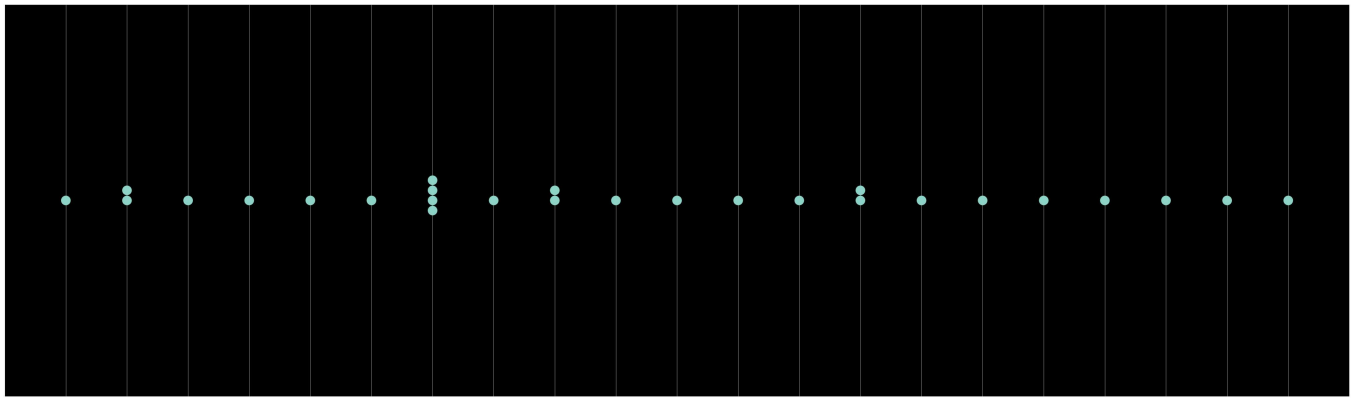
	Year	Disaster Subgroup	Disaster Type	Disaster Subtype	Country	ISO	Region	Loc
8384	2021	Meteorological	Storm	Storm	Chile	CHL	South America	Sa Metro OÃ¢â‚¬â„¢
8401	2021	Meteorological	Storm	Storm	Spain	ESP	Southern Europe	Madrid, M Za
8419	2021	Geophysical	Earthquake	Earthquake	Indonesia	IDN	South-Eastern Asia	Mamuju, M Polewali M districts
8428	2021	Hydrological	Flood	Flood	Indonesia	IDN	South-Eastern Asia	Kalim Sulawesi Maluku
8449	2021	Hydrological	Flood	Flood	Paraguay	PRY	South America	Distrito C Paragu C
8464	2021	Hydrological	Flood	Flood	Bolivia (Plurinational State of)	BOL	South America	Suc (Chuc Depa
8472	2021	Meteorological	Storm	Storm	Japan	JPN	Eastern Asia	Hokuriku, T Hc
8474	2021	Meteorological	Storm	Storm	Taiwan (Province of China)	TWN	Eastern Asia	Taipei, New Taoyuan Keelung

```
# Display countries wrt year 2021
yc=gg[['Year','Country']]
yc
```

	Year	Country
8384	2021	Chile
8401	2021	Spain
8419	2021	Indonesia
8428	2021	Indonesia
8449	2021	Paraguay
8464	2021	Bolivia (Plurinational State of)
8472	2021	Japan
8474	2021	Taiwan (Province of China)
8483	2021	India
8489	2021	Mozambique
8508	2021	Malaysia
8520	2021	Australia
8532	2021	Bolivia (Plurinational State of)
8540	2021	China
8566	2021	Fiji
8601	2021	Indonesia
8612	2021	Indonesia
8621	2021	Iran (Islamic Republic of)

```
#Width=100 and Height=30
plt.figure(figsize=(100,30))
plt.style.use('dark_background')
sns.swarmplot(x ='Country', data = yc,s=40).set_title('Seaborn Plot')
```

```
Text(0.5, 1.0, 'Seaborn Plot')
```



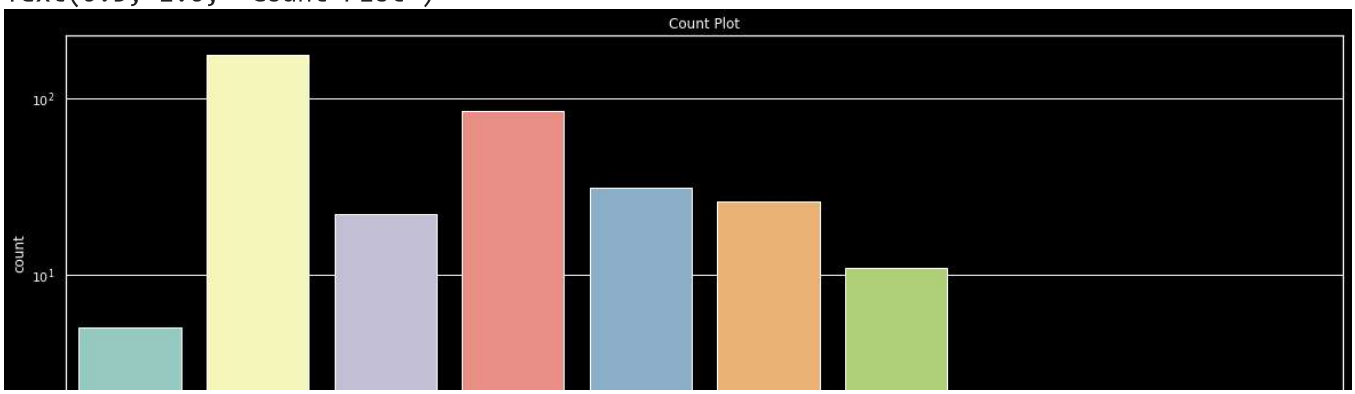
COUNT PLOT

```
plt.figure(figsize=(20,9))
plt.yscale('log') # Set Y axis scale to log
plt.style.use('dark_background')
# x,hue - inputs
sns.countplot(x = 'Disaster Subgroup', hue='Disaster Type' ,data = df).set_title('Count Plot'
plt.legend(loc="upper right")
# In df,extract all row values of India
i=df[df['Country'] == 'India']
i
```

	Year	Disaster Subgroup	Disaster Type	Disaster Subtype	Country	ISO	Region	Location	St Y
118	2000	Climatological	Drought	Drought	India	IND	Southern Asia	New Delhi city (Delhi province), Gujarat, Raja...	200
132	2000	Hydrological	Flood	Flood	India	IND	Southern Asia	Yingkiang,Tuting villages (Upper Siang (70051)...	200
215	2000	Hydrological	Flood	Flood	India	IND	Southern Asia	East Siang, 70041, West Siang, 70054, Upper Si...	200
216	2000	Hydrological	Flood	Flood	India	IND	Southern Asia	Adilabad, Cuddapah, East Godavari, Guntur, Hyd...	200
217	2000	Hydrological	Flood	Flood	India	IND	Southern Asia	Katihar, Sitamarhi districts (Bihar province),...	200
...	...	...	...	...	...	...	...	...	
8615	2020	Meteorological	Storm	Storm	India	IND	Southern Asia	0	20
								Raigad, Pune	

```
# Extract Disaster Type and Total Affected of India
DT = i[['Disaster Type','Total Affected']]
DT
plt.figure(figsize=(20,7))
plt.style.use('dark_background')
plt.yscale('log')
sns.countplot(x='Disaster Type',data=DT).set_title('Count Plot')
```

```
Text(0.5, 1.0, 'Count Plot')
```



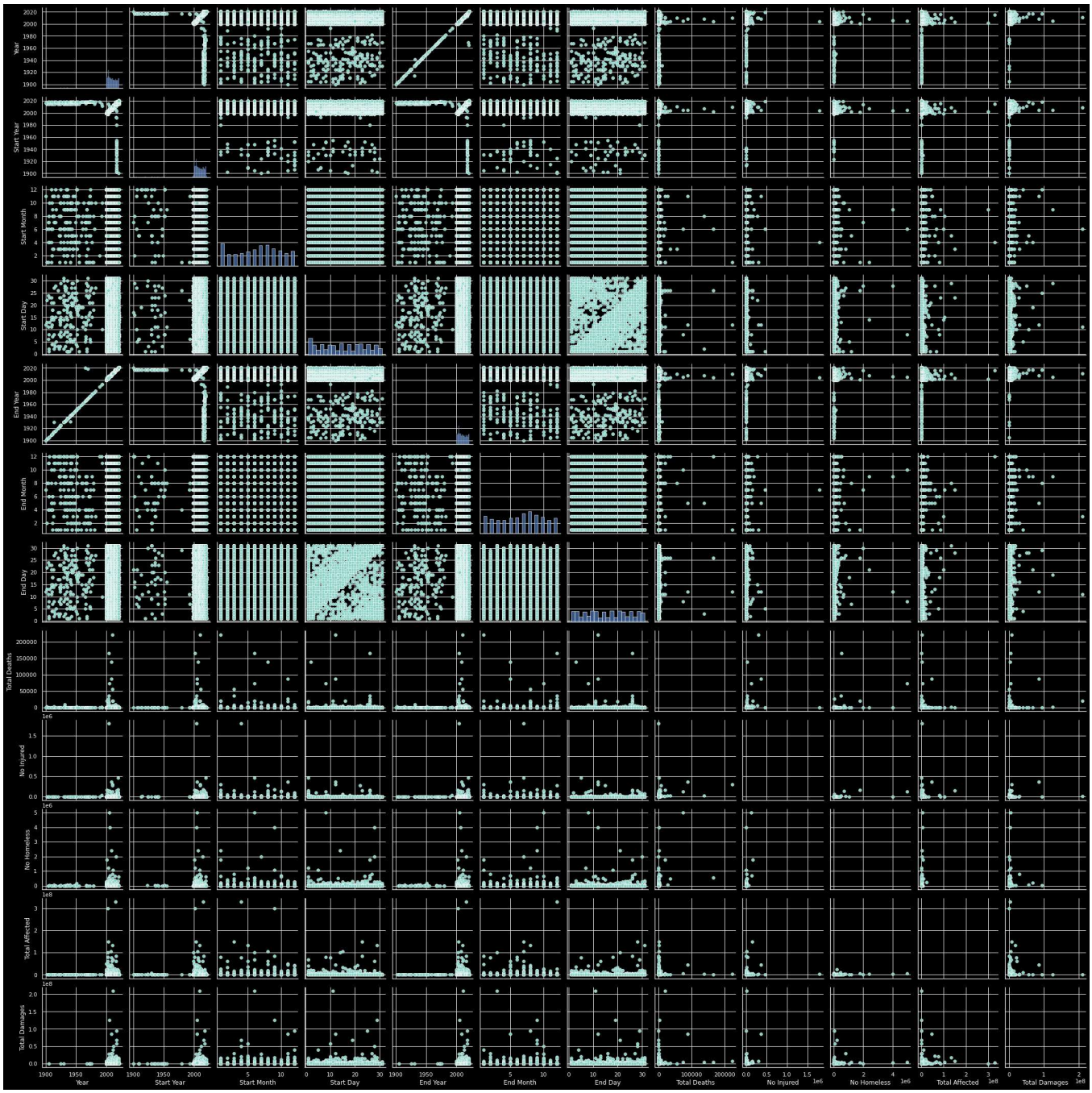
STRIP PLOT

```
# Sort Total Affected column in Descending order and print first 10 values
TA=df.sort_values(by='Total Affected',ascending=False).head(10)
# Prints first 10 Countries with Highest Total Affected values
print(TA['Country'])
# Prints first 10 Highest Total Affected values
14
print(TA['Total Affected'])
plt.figure(figsize=(20,6))
# Sort Total Affected column in Descending order and print first 40 values
TA=df.sort_values(by='Total Affected',ascending=False).head(40)
t=TA['Total Affected']
c=TA['Country']
plt.style.use('dark_background')
# x axis - Countries , y axis - Total Affected Count
sns.stripplot(x=c,y=t,hue='Country',data=TA,jitter=True,s=9)
```

```
6492                            India
1154                            India
1567                            China
4297                            China
3484                            China
746                             China
7005    United States of America (the)
664                             China
3725                            China
4448                            China
Name: Country, dtype: object
6492    330000000.0
1154    300000000.0
1567    150146000.0
4297    134000000.0
3484    105004000.0
746     100000000.0
7005     85000012.0
664      80035257.0
3725     77000000.0
4448     67900000.0
Name: Total Affected, dtype: float64
<matplotlib.axes._subplots.AxesSubplot at 0x7fd84ee6e050>
```

PAIR PLOT

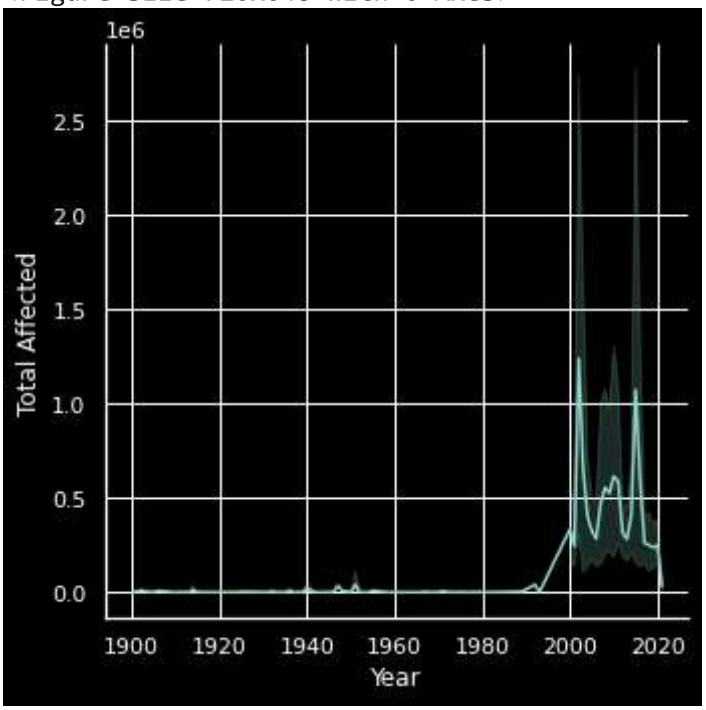

```
sns.pairplot(data=df)
plt.show()
```


Year
Start Year
Start Month
Start Day
End Year
End Month
End Day
Total Deaths
No Injured
No Homeless
Total Affected
Total Damages

RED PLOT

```
plt.figure(figsize=(10,9))
sns.relplot(x=df['Year'], y=df['Total Affected'], kind='line');
```

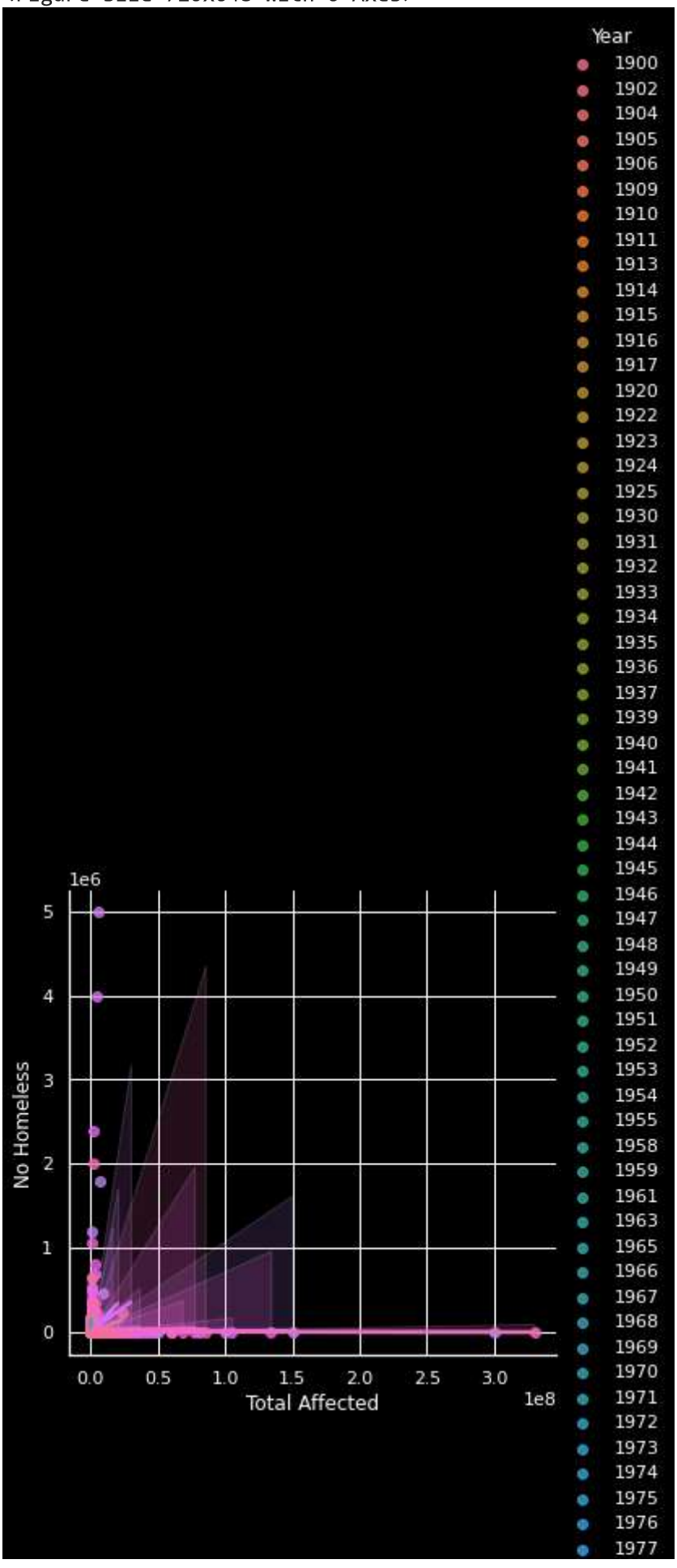
```
<Figure size 720x648 with 0 Axes>
```



LM PLOT

```
plt.figure(figsize=(10,9))
sns.lmplot(x='Total Affected', y='No Homeless',hue='Year', data= df, fit_reg= True)
plt.show()
```

```
<Figure size 720x648 with 0 Axes>
```

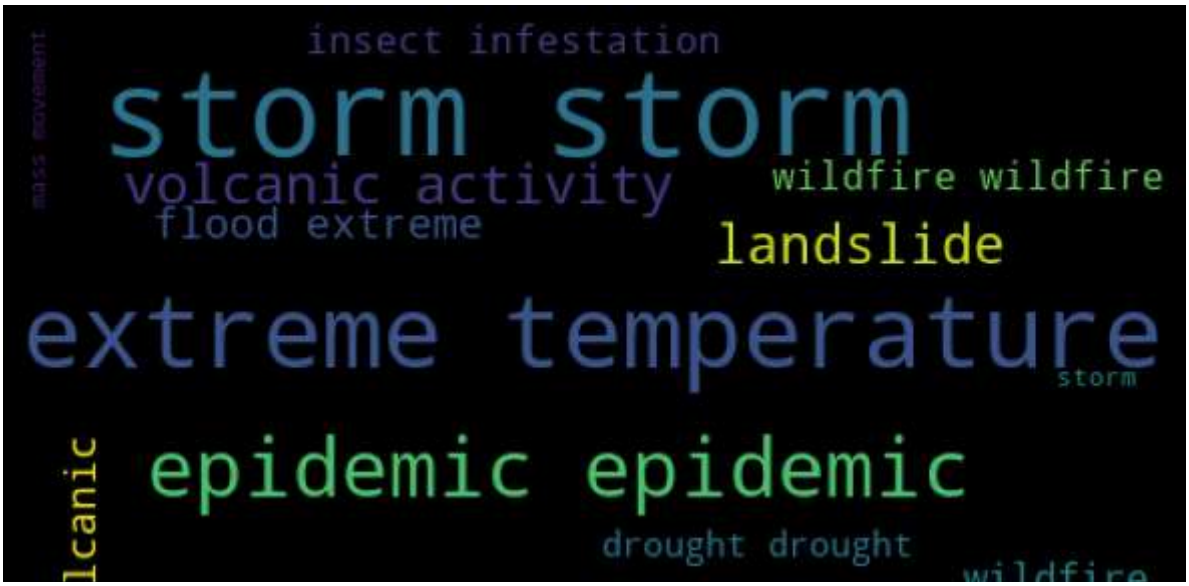


VIOLIN PLOT

```
plt.figure(figsize=(15,8))
sns.violinplot(data=df, x='Year', y='No Homeless', linewidth=1,)
plt.show()
```

WORD CLOUD

```
from PIL import Image
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS
import pandas as pd
df=pd.read_csv("/content/drive/MyDrive/DV/NATURAL DISASTERS.csv")
comment_words = " "
stopwords = set(STOPWORDS)
#Iterating disaster subtype values
for val in df['Disaster Subtype'][0:3000]:
  val = str(val)
  tokens = val.split()
  #converting to lowercase
  for i in range(len(tokens)):
    tokens[i] = tokens[i].lower()
    comment_words += " ".join(tokens)+" "
#generate wordcloud
wc = WordCloud(width = 500, height = 500, background_color ='black', stopwords = stopwords, m
plt.figure(figsize = (8, 8), facecolor = None)

#display wordcloud
plt.imshow(wc)
plt.axis('off')
plt.tight_layout(pad = 0)
plt.show()
```



WAFFLE CHART

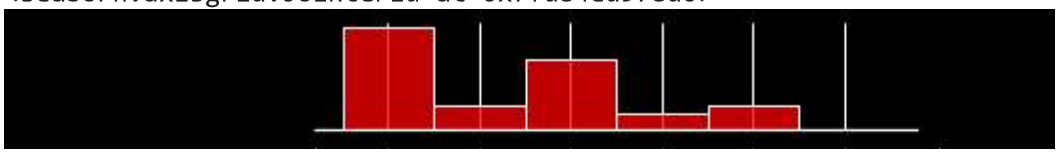

```
!pip install pywaffle
from pywaffle import Waffle
data=pd.read_csv("/content/drive/MyDrive/DV/NATURAL DISASTERS.csv")
data['Disaster Type'].value_counts()
dis_data={'Disaster_Type':['Insect infestation','Mass movement','Glacial lake outburst','Impa
#converting data to dataframe
dataframe = pd.DataFrame(dis_data)
plt.style.use('dark_background')
#plotting waffle chart with five rows of cells
fig = plt.figure(FigureClass = Waffle,values = dataframe.Count,rows=5,labels = list(dataframe
```

```
Collecting pywaffle
  Downloading pywaffle-0.6.4-py2.py3-none-any.whl (565 kB)
     |████████████████████████████████| 565 kB 7.5 MB/s
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: numpy>=1.11 in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/li
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from
Installing collected packages: pywaffle
Successfully installed pywaffle-0.6.4
```

JOINT PLOT

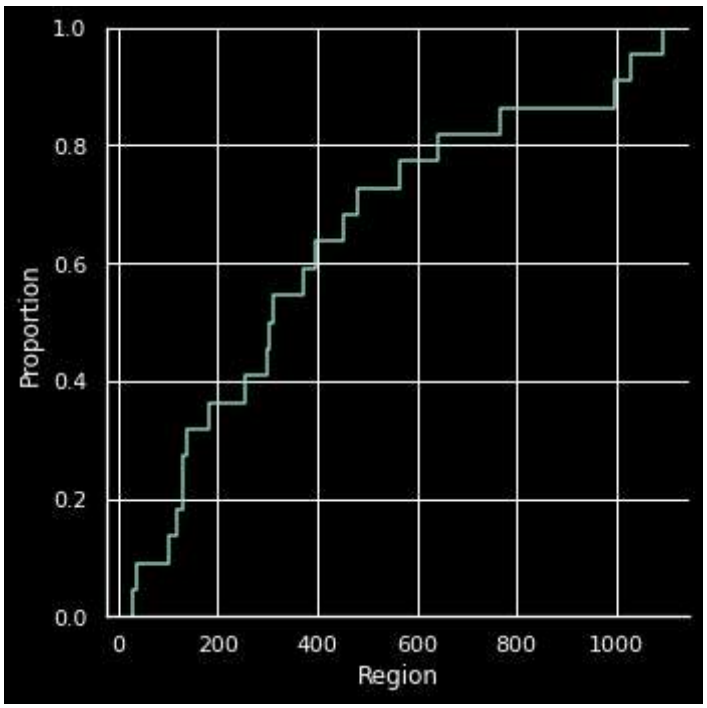
```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
data=pd.read_csv("/content/drive/MyDrive/DV/NATURAL DISASTERS.csv")
#renaming column names
data=data.replace(['Hydrological'],'Hydro')
data=data.replace(['Biological'],'Bio')
data=data.replace(['Meteorological'],'Meteoro')
data=data.replace(['Climatological'],'Climat')
data=data.replace(['Geophysical'],'Geo')
data=data.replace(['Extra-terrestrial'],'Extra')
#plotting jointplot
sns.jointplot(x = 'Disaster Subgroup',y = 'Disaster Type',color='red',data = data)
```

```
<seaborn.axisgrid.JointGrid at 0x7fd84ed978d0>
```



DIS PLOT

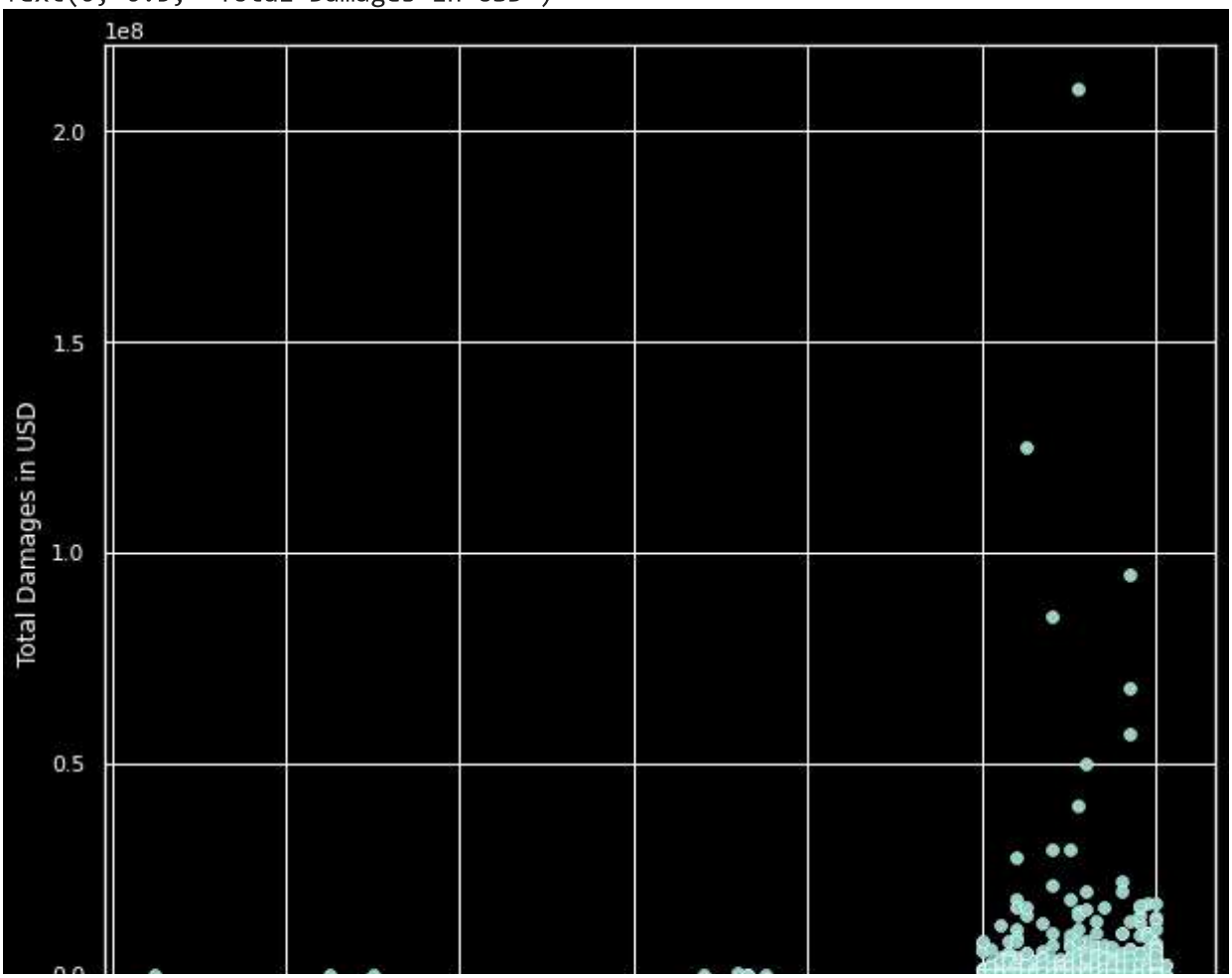
```
import seaborn as sns
#empirical cumulative distribution function plot
#plotting displot
sns.displot(x =data['Region'].value_counts(),kind='ecdf')
plt.show()
```



SCATTER PLOT

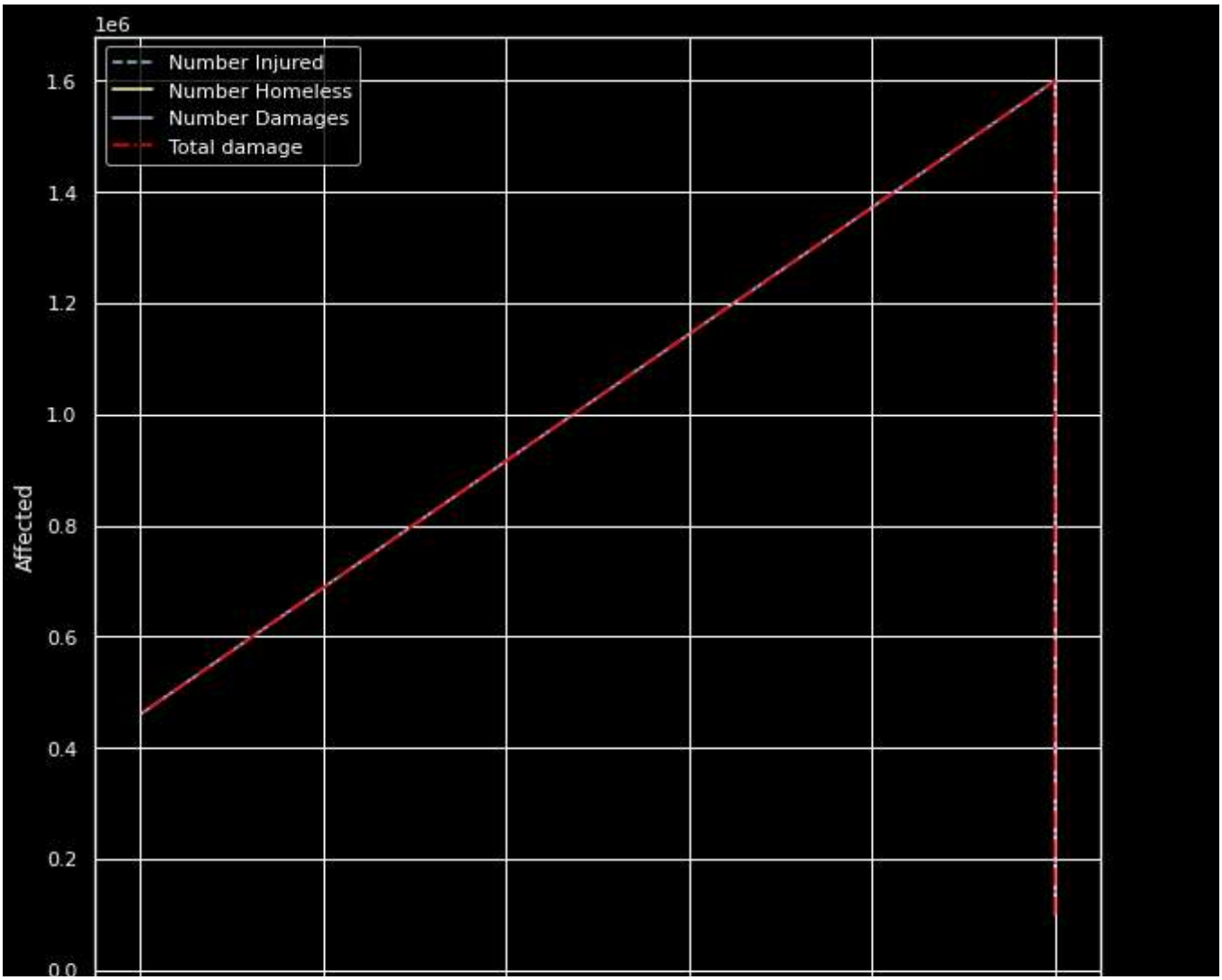
```
#selecting data to make dataframe sn
plt.figure(figsize=(10,9))
df=pd.read_csv("/content/drive/MyDrive/DV/NATURAL DISASTERS.csv")

sn=df['Total Damages'].iloc[100:9000:500]
sns.scatterplot(x=df['Year'],y=df['Total Damages'],data=sn)
plt.ylabel('Total Damages in USD')
```

```
Text(0, 0.5, 'Total Damages in USD')
```

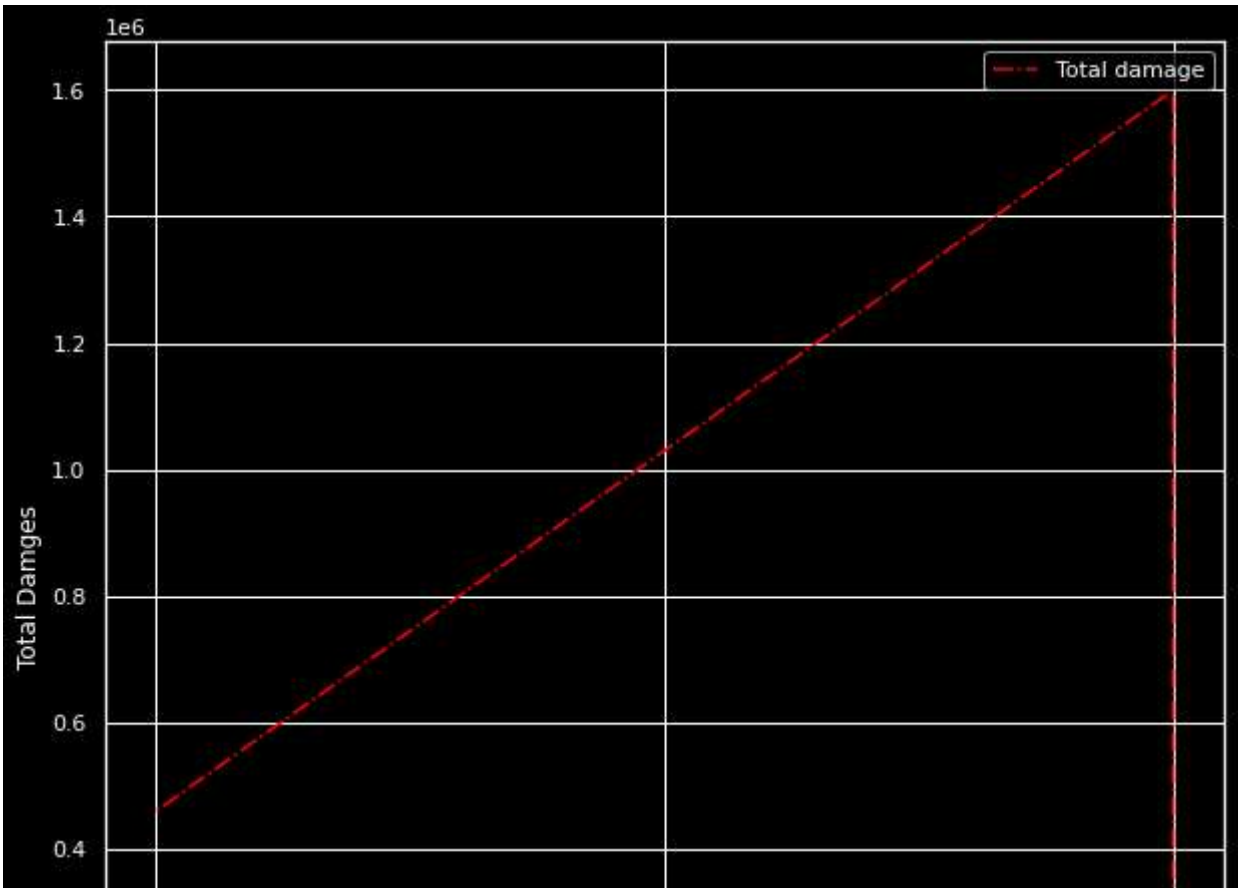


LINE PLOT

```
#setting up size of the figure
plt.figure(figsize=[10,10])
state=df['Country'].iloc[1000:9000:1000] #selecting data to make dataframe state
ar=df['No Injured'].iloc[1000:9000:1000] #selecting data to make dataframe ar
disr=df['No Homeless'].iloc[1000:9000:1000] #selecting data to make dataframe disr
dr=df['Total Damages'].iloc[1000:9000:1000] #selecting data to make dataframe dr
plt.plot(state,ar,linestyle = "--",label='Number Injured') #plotting for the data in datafram
plt.plot(state,disr,label='Number Homeless') #plotting for the data in dataframe disr
plt.plot(state,dr,label='Number Damages') #plotting for the data in dataframe dr
plt.plot(state,dr,linestyle = "-.",label='Total damage',color='red')
plt.xlabel('Disaster')
plt.ylabel('Affected')
plt.legend()
plt.show()
```




```
#setting up size of the figure
plt.figure(figsize=[10,10])

#selecting data to make from dataframe state
state=df['Disaster Subtype'].iloc[1000:9000:1000]
#selecting data to make from dataframe dr
dr=df['Total Damages'].iloc[1000:9000:1000]
plt.plot(state,dr,linestyle = "-.",label='Total damage',color='red')
plt.xlabel('Disaster')
plt.ylabel('Total Damges')
plt.legend()
plt.show()
```



HISTOGRAM

```
#setting up size of the figure
plt.figure(figsize=[10,5])
max_occured=df.sort_values(by='Total Damages',ascending=False)
md=max_occured.head(100)
#selecting data from dataframe md
a=md['Disaster Type']
plt.hist(a,bins=10,edgecolor='Black',color='Cyan',histtype='stepfilled')
plt.title('Histogram')
plt.ylabel('Most frequently occurred calamity')
plt.show()
```