

Cause Of Death

Problem Statement:

This contains the problem statement and business goal

Content

In this Dataset, we have Historical Data of different cause of deaths for all ages around the World. The key features of this Dataset are: Meningitis, Alzheimer's Disease and Other Dementias, Parkinson's Disease, Nutritional Deficiencies, Malaria, Drowning, Interpersonal Violence, Maternal Disorders, HIV/AIDS, Drug Use Disorders, Tuberculosis, Cardiovascular Diseases, Lower Respiratory Infections, Neonatal Disorders, Alcohol Use Disorders, Self-harm, Exposure to Forces of Nature, Diarrheal Diseases, Environmental Heat and Cold Exposure, Neoplasms, Conflict and Terrorism, Diabetes Mellitus, Chronic Kidney Disease, Poisonings, Protein-Energy Malnutrition, Road Injuries, Chronic Respiratory Diseases, Cirrhosis and Other Chronic Liver Diseases, Digestive Diseases, Fire, Heat, and Hot Substances, Acute Hepatitis.

```
In [2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from plotly.offline import init_notebook_mode
init_notebook_mode(connected=True)

import warnings
warnings.filterwarnings('ignore')
```

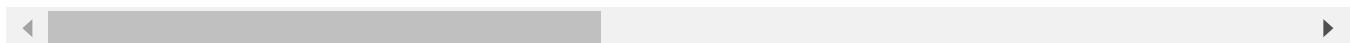
```
In [3]: df=pd.read_csv(r"C:\Users\Kundan Patil\DS0522\INTERNSHIP\project\cause_of_deaths da
```

```
In [4]: df
```

Out[4]:

	Country/Territory	Code	Year	Meningitis	Alzheimer's Disease and Other Dementias	Parkinson's Disease	Nutritional Deficiencies	Malaria
0	Afghanistan	AFG	1990	2159	1116	371	2087	93
1	Afghanistan	AFG	1991	2218	1136	374	2153	189
2	Afghanistan	AFG	1992	2475	1162	378	2441	239
3	Afghanistan	AFG	1993	2812	1187	384	2837	108
4	Afghanistan	AFG	1994	3027	1211	391	3081	211
...
6115	Zimbabwe	ZWE	2015	1439	754	215	3019	2518
6116	Zimbabwe	ZWE	2016	1457	767	219	3056	2050
6117	Zimbabwe	ZWE	2017	1460	781	223	2990	2116
6118	Zimbabwe	ZWE	2018	1450	795	227	2918	2088
6119	Zimbabwe	ZWE	2019	1450	812	232	2884	2068

6120 rows × 34 columns



In [5]: df.shape

Out[5]: (6120, 34)

In [6]: df.dtypes

```
Out[6]:    Country/Territory          object
           Code                      object
           Year                     int64
           Meningitis                int64
           Alzheimer's Disease and Other Dementias   int64
           Parkinson's Disease        int64
           Nutritional Deficiencies  int64
           Malaria                   int64
           Drowning                  int64
           Interpersonal Violence    int64
           Maternal Disorders        int64
           HIV/AIDS                 int64
           Drug Use Disorders        int64
           Tuberculosis              int64
           Cardiovascular Diseases   int64
           Lower Respiratory Infections  int64
           Neonatal Disorders        int64
           Alcohol Use Disorders     int64
           Self-harm                 int64
           Exposure to Forces of Nature  int64
           Diarrheal Diseases        int64
           Environmental Heat and Cold Exposure  int64
           Neoplasms                 int64
           Conflict and Terrorism    int64
           Diabetes Mellitus          int64
           Chronic Kidney Disease    int64
           Poisonings                int64
           Protein-Energy Malnutrition  int64
           Road Injuries              int64
           Chronic Respiratory Diseases  int64
           Cirrhosis and Other Chronic Liver Diseases  int64
           Digestive Diseases         int64
           Fire, Heat, and Hot Substances   int64
           Acute Hepatitis             int64
           dtype: object
```

In [7]: `df.describe()`

	Year	Meningitis	Alzheimer's Disease and Other Dementias	Parkinson's Disease	Nutritional Deficiencies	Malaria	
count	6120.000000	6120.000000	6120.000000	6120.000000	6120.000000	6120.000000	6
mean	2004.500000	1719.701307	4864.189379	1173.169118	2253.600000	4140.960131	1
std	8.656149	6672.006930	18220.659072	4616.156238	10483.633601	18427.753137	8
min	1990.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	1997.000000	15.000000	90.000000	27.000000	9.000000	0.000000	
50%	2004.500000	109.000000	666.500000	164.000000	119.000000	0.000000	
75%	2012.000000	847.250000	2456.250000	609.250000	1167.250000	393.000000	
max	2019.000000	98358.000000	320715.000000	76990.000000	268223.000000	280604.000000	153

8 rows × 32 columns

In [8]: `df.isnull().sum()`

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
Out[8]:    Country/Territory      0
           Code                  0
           Year                 0
           Meningitis            0
           Alzheimer's Disease and Other Dementias 0
           Parkinson's Disease   0
           Nutritional Deficiencies 0
           Malaria                0
           Drowning                0
           Interpersonal Violence 0
           Maternal Disorders     0
           HIV/AIDS               0
           Drug Use Disorders     0
           Tuberculosis           0
           Cardiovascular Diseases 0
           Lower Respiratory Infections 0
           Neonatal Disorders     0
           Alcohol Use Disorders   0
           Self-harm               0
           Exposure to Forces of Nature 0
           Diarrheal Diseases      0
           Environmental Heat and Cold Exposure 0
           Neoplasms               0
           Conflict and Terrorism   0
           Diabetes Mellitus        0
           Chronic Kidney Disease   0
           Poisonings              0
           Protein-Energy Malnutrition 0
           Road Injuries            0
           Chronic Respiratory Diseases 0
           Cirrhosis and Other Chronic Liver Diseases 0
           Digestive Diseases       0
           Fire, Heat, and Hot Substances 0
           Acute Hepatitis          0
           dtype: int64
```

In [9]: `df.columns`

```
Out[9]: Index(['Country/Territory', 'Code', 'Year', 'Meningitis',
       'Alzheimer's Disease and Other Dementias', 'Parkinson's Disease',
       'Nutritional Deficiencies', 'Malaria', 'Drowning',
       'Interpersonal Violence', 'Maternal Disorders', 'HIV/AIDS',
       'Drug Use Disorders', 'Tuberculosis', 'Cardiovascular Diseases',
       'Lower Respiratory Infections', 'Neonatal Disorders',
       'Alcohol Use Disorders', 'Self-harm', 'Exposure to Forces of Nature',
       'Diarrheal Diseases', 'Environmental Heat and Cold Exposure',
       'Neoplasms', 'Conflict and Terrorism', 'Diabetes Mellitus',
       'Chronic Kidney Disease', 'Poisonings', 'Protein-Energy Malnutrition',
       'Road Injuries', 'Chronic Respiratory Diseases',
       'Cirrhosis and Other Chronic Liver Diseases', 'Digestive Diseases',
       'Fire, Heat, and Hot Substances', 'Acute Hepatitis'],
      dtype='object')
```

In [10]: `df.duplicated(keep='last')`

```
Out[10]: 0      False
         1      False
         2      False
         3      False
         4      False
         ...
        6115    False
        6116    False
        6117    False
        6118    False
        6119    False
Length: 6120, dtype: bool
```

```
In [11]: #great we don't have null and duplicate values in our data set
```

Country/Territory

```
In [12]: df['Country/Territory'].describe()
```

```
Out[12]: count      6120
unique      204
top       Afghanistan
freq       30
Name: Country/Territory, dtype: object
```

```
In [13]: # Country/Territory contain nominal data in text formate
```

```
In [14]: #checking unique of variable
print(df['Country/Territory'].unique())
#counting the uniques
print(df['Country/Territory'].value_counts())
```

```
[ 'Afghanistan' 'Albania' 'Algeria' 'American Samoa' 'Andorra' 'Angola'
 'Antigua and Barbuda' 'Argentina' 'Armenia' 'Australia' 'Austria'
 'Azerbaijan' 'Bahamas' 'Bahrain' 'Bangladesh' 'Barbados' 'Belarus'
 'Belgium' 'Belize' 'Benin' 'Bermuda' 'Bhutan' 'Bolivia'
 'Bosnia and Herzegovina' 'Botswana' 'Brazil' 'Brunei' 'Bulgaria'
 'Burkina Faso' 'Burundi' 'Cambodia' 'Cameroon' 'Canada' 'Cape Verde'
 'Central African Republic' 'Chad' 'Chile' 'China' 'Colombia' 'Comoros'
 'Congo' 'Cook Islands' 'Costa Rica' "Cote d'Ivoire" 'Croatia' 'Cuba'
 'Cyprus' 'Czechia' 'Democratic Republic of Congo' 'Denmark' 'Djibouti'
 'Dominica' 'Dominican Republic' 'Ecuador' 'Egypt' 'El Salvador'
 'Equatorial Guinea' 'Eritrea' 'Estonia' 'Eswatini' 'Ethiopia' 'Fiji'
 'Finland' 'France' 'Gabon' 'Gambia' 'Georgia' 'Germany' 'Ghana' 'Greece'
 'Greenland' 'Grenada' 'Guam' 'Guatemala' 'Guinea' 'Guinea-Bissau'
 'Guyana' 'Haiti' 'Honduras' 'Hungary' 'Iceland' 'India' 'Indonesia'
 'Iran' 'Iraq' 'Ireland' 'Israel' 'Italy' 'Jamaica' 'Japan' 'Jordan'
 'Kazakhstan' 'Kenya' 'Kiribati' 'Kuwait' 'Kyrgyzstan' 'Laos' 'Latvia'
 'Lebanon' 'Lesotho' 'Liberia' 'Libya' 'Lithuania' 'Luxembourg'
 'Madagascar' 'Malawi' 'Malaysia' 'Maldives' 'Mali' 'Malta'
 'Marshall Islands' 'Mauritania' 'Mauritius' 'Mexico' 'Micronesia'
 'Moldova' 'Monaco' 'Mongolia' 'Montenegro' 'Morocco' 'Mozambique'
 'Myanmar' 'Namibia' 'Nauru' 'Nepal' 'Netherlands' 'New Zealand'
 'Nicaragua' 'Niger' 'Nigeria' 'Niue' 'North Korea' 'North Macedonia'
 'Northern Mariana Islands' 'Norway' 'Oman' 'Pakistan' 'Palau' 'Palestine'
 'Panama' 'Papua New Guinea' 'Paraguay' 'Peru' 'Philippines' 'Poland'
 'Portugal' 'Puerto Rico' 'Qatar' 'Romania' 'Russia' 'Rwanda'
 'Saint Kitts and Nevis' 'Saint Lucia' 'Saint Vincent and the Grenadines'
 'Samoa' 'San Marino' 'Sao Tome and Principe' 'Saudi Arabia' 'Senegal'
 'Serbia' 'Seychelles' 'Sierra Leone' 'Singapore' 'Slovakia' 'Slovenia'
 'Solomon Islands' 'Somalia' 'South Africa' 'South Korea' 'South Sudan'
 'Spain' 'Sri Lanka' 'Sudan' 'Suriname' 'Sweden' 'Switzerland' 'Syria'
 'Taiwan' 'Tajikistan' 'Tanzania' 'Thailand' 'Timor' 'Togo' 'Tokelau'
 'Tonga' 'Trinidad and Tobago' 'Tunisia' 'Turkey' 'Turkmenistan' 'Tuvalu'
 'Uganda' 'Ukraine' 'United Arab Emirates' 'United Kingdom'
 'United States' 'United States Virgin Islands' 'Uruguay' 'Uzbekistan'
 'Vanuatu' 'Venezuela' 'Vietnam' 'Yemen' 'Zambia' 'Zimbabwe']
```

Afghanistan 30

Papua New Guinea 30

Niue 30

North Korea 30

North Macedonia 30

..

Greenland 30

Grenada 30

Guam 30

Guatemala 30

Zimbabwe 30

Name: Country/Territory, Length: 204, dtype: int64

In []:

Year

In [15]: df['Year'].describe()

```
Out[15]: count    6120.000000
          mean     2004.500000
          std      8.656149
          min     1990.000000
          25%    1997.000000
          50%    2004.500000
          75%    2012.000000
          max     2019.000000
          Name: Year, dtype: float64
```

```
In [16]: #checking unique of variable
print(df['Year'].unique())
#counting the uniques
print(df['Year'].value_counts())
```

```
[1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003
 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017
 2018 2019]
1990    204
1991    204
2018    204
2017    204
2016    204
2015    204
2014    204
2013    204
2012    204
2011    204
2010    204
2009    204
2008    204
2007    204
2006    204
2005    204
2004    204
2003    204
2002    204
2001    204
2000    204
1999    204
1998    204
1997    204
1996    204
1995    204
1994    204
1993    204
1992    204
2019    204
          Name: Year, dtype: int64
```

```
In [17]: # In the year column contains ordinal data
          # it is equally distributed
          # we have data from year 1990 to 2019 = 30 year of death records we have
```

04. Meningitis - No. of People died from Meningitis

```
In [18]: # No.of People died from Meningitis
```

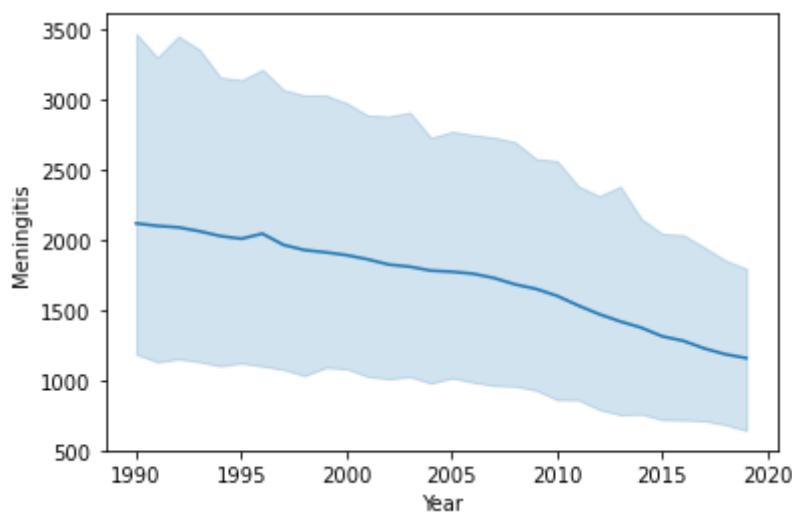
```
In [19]: df['Meningitis'].describe()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
Out[19]: count    6120.000000
          mean     1719.701307
          std      6672.006930
          min      0.000000
          25%     15.000000
          50%     109.000000
          75%     847.250000
          max     98358.000000
          Name: Meningitis, dtype: float64
```

```
In [20]: sns.lineplot(data=df, x="Year", y="Meningitis")
```

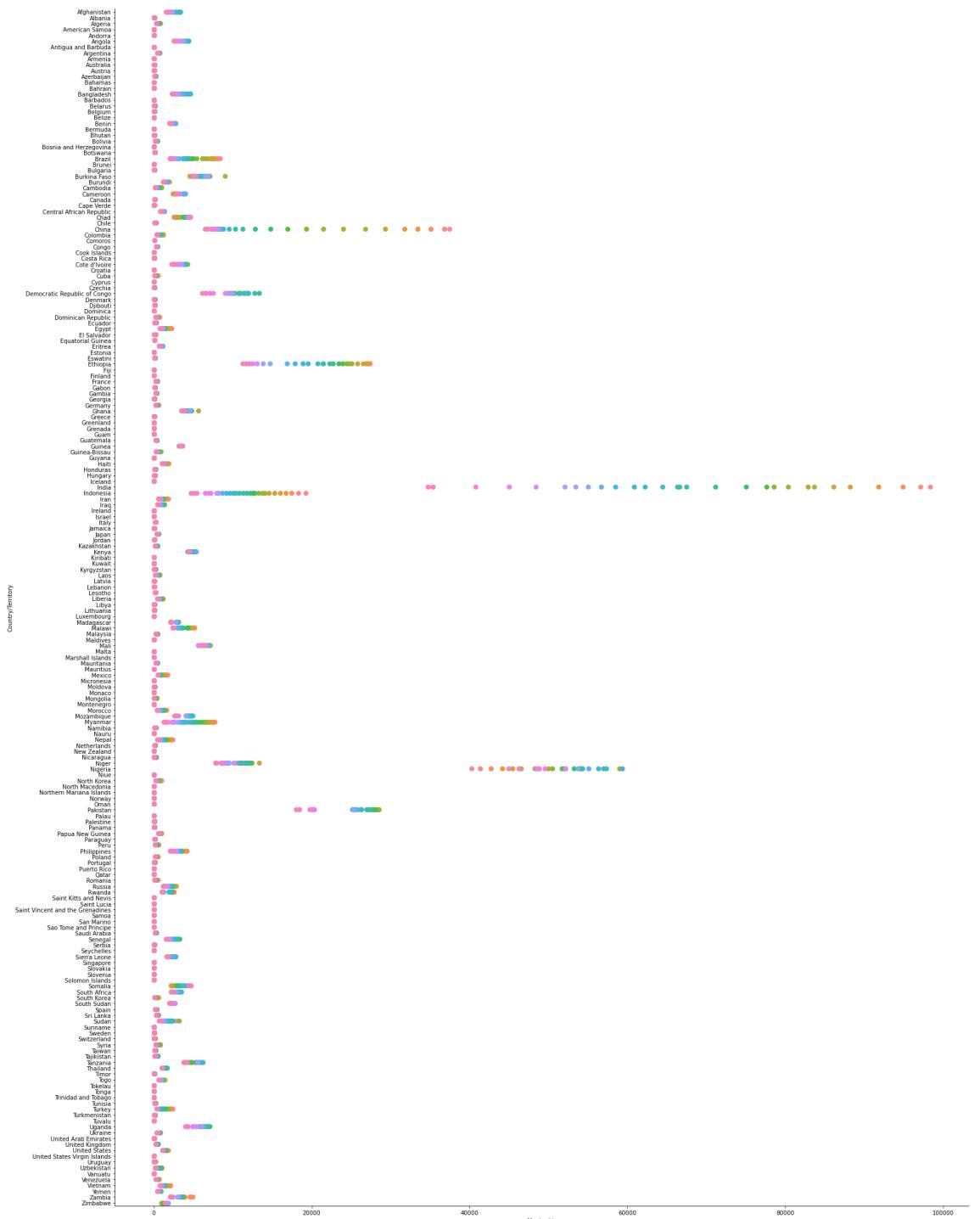
```
Out[20]: <AxesSubplot:xlabel='Year', ylabel='Meningitis'>
```



```
In [21]: # No.of People died from Meningitis is following down trend pattern for overall population
```

```
In [22]: sns.factorplot(x='Meningitis',y='Country/Territory',hue='Year',data=df,size=30,aspect=1)
```

```
Out[22]: <seaborn.axisgrid.FacetGrid at 0x238f1a17040>
```



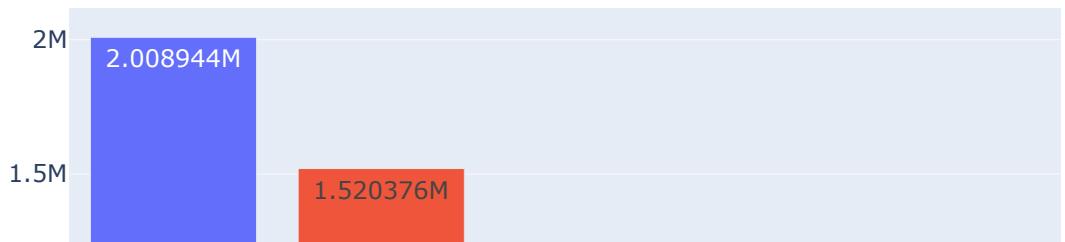
In [23]: # Q. in which country maximum people were died due to Meningitis
A. india
Q. in which year death percentage was high
A. the year was 1990

```
In [24]: diseases = [ 'Meningitis',
    "Alzheimer's Disease and Other Dementias", "Parkinson's Disease",
    'Nutritional Deficiencies', 'Malaria', 'Drowning',
    'Interpersonal Violence', 'Maternal Disorders', 'HIV/AIDS',
    'Drug Use Disorders', 'Tuberculosis', 'Cardiovascular Diseases',
    'Lower Respiratory Infections', 'Neonatal Disorders',
    'Alcohol Use Disorders', 'Self-harm', 'Exposure to Forces of Nature',
    'Diarrheal Diseases', 'Environmental Heat and Cold Exposure',
    'Neoplasms', 'Conflict and Terrorism', 'Diabetes Mellitus',
    'Chronic Kidney Disease', 'Poisonings', 'Protein-Energy Malnutrition',
    'Road Injuries', 'Chronic Respiratory Diseases',
    'Cirrhosis and Other Chronic Liver Diseases', 'Digestive Diseases',
    'Fire, Heat, and Hot Substances', 'Acute Hepatitis']
```

```
In [25]: # Top 10 Country name No.of People died from Meningitis
data = df.groupby(['Country/Territory'])["Meningitis"].sum().sort_values(ascending
```

```
In [26]: px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index,te
```

No. of People died from Meningitis



05. Alzheimer's Disease and Other Dementias - No. of People died from Alzheimer's Disease and Other Dementias

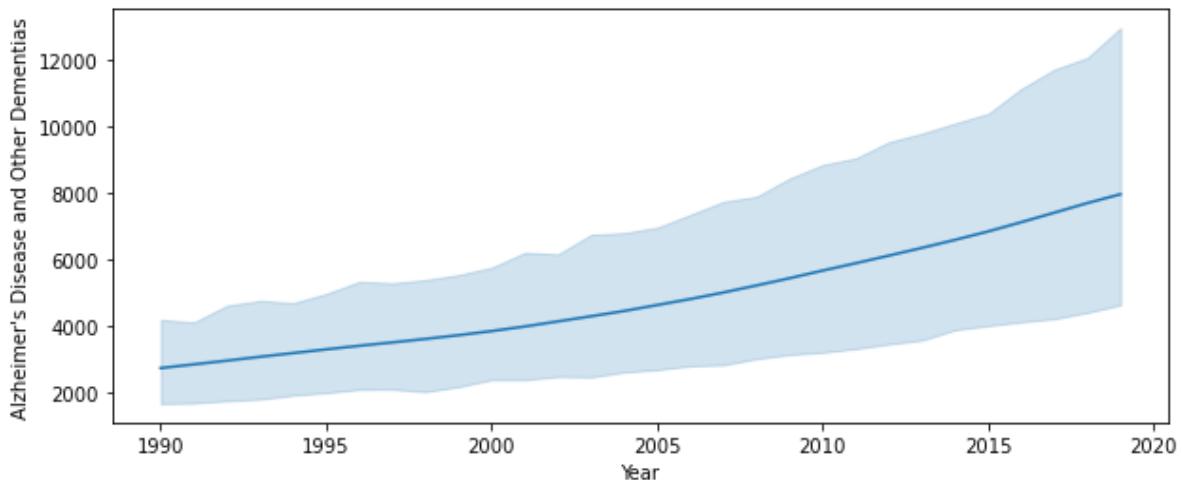
```
In [27]: ## No.of People died from Alzheimer's Disease and Other Dementias
```

```
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js
In [28]: df['ALZHEIMER S DISEASE AND OTHER DEMENTIAS'].describe()
```

```
Out[28]: count      6120.000000
          mean       4864.189379
          std        18220.659072
          min         0.000000
          25%        90.000000
          50%        666.500000
          75%        2456.250000
          max       320715.000000
          Name: Alzheimer's Disease and Other Dementias, dtype: float64
```

```
In [29]: plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Alzheimer's Disease and Other Dementias")
```

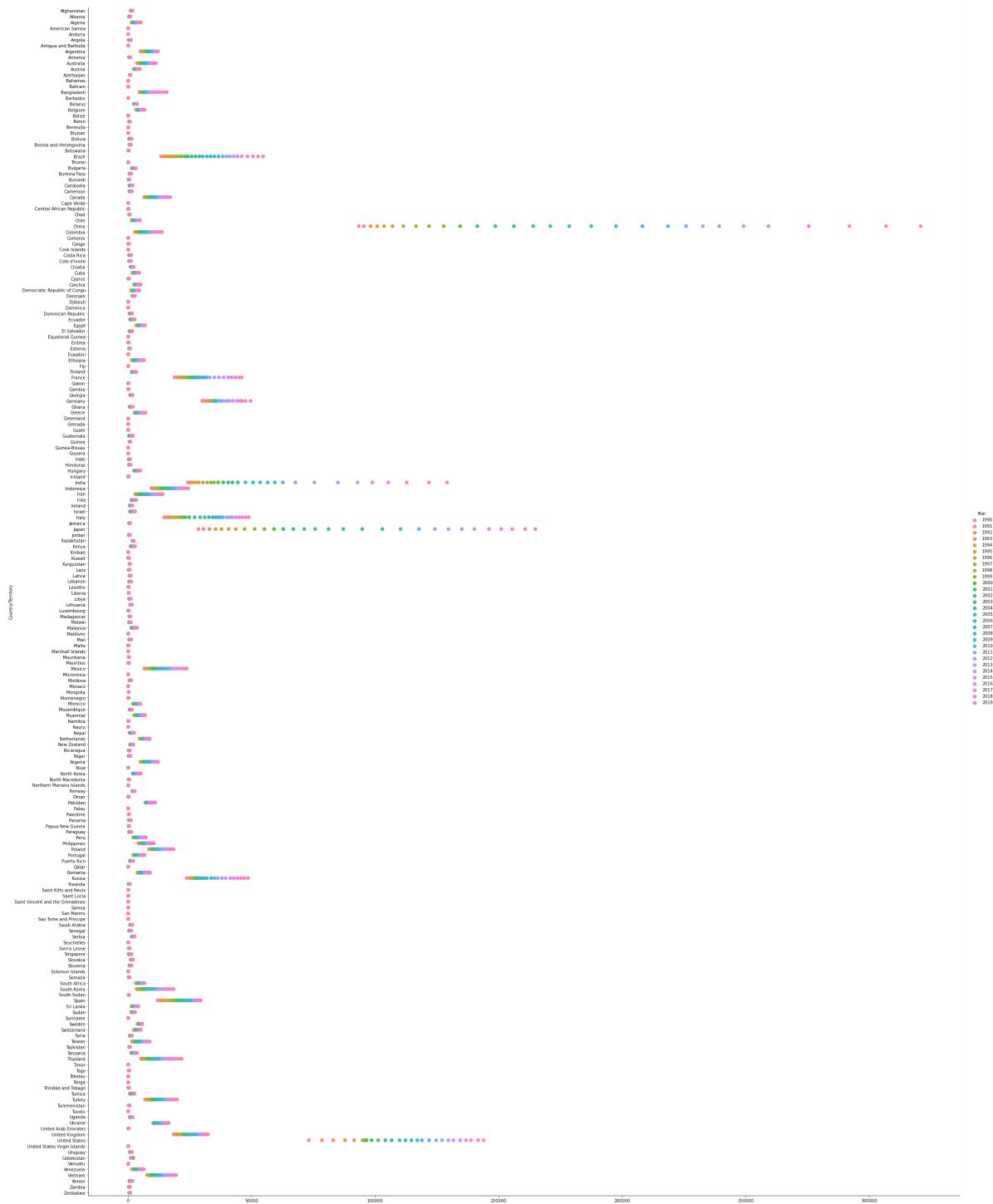
```
Out[29]: <AxesSubplot:xlabel='Year', ylabel="Alzheimer's Disease and Other Dementias">
```



```
In [30]: # No.of People died from Alzheimer's Disease is following up trend pattern for over 30 years
```

```
In [31]: sns.factorplot(x="Alzheimer's Disease and Other Dementias",y='Country/Territory',hue=
```

```
Out[31]: <seaborn.axisgrid.FacetGrid at 0x238f2abc3d0>
```

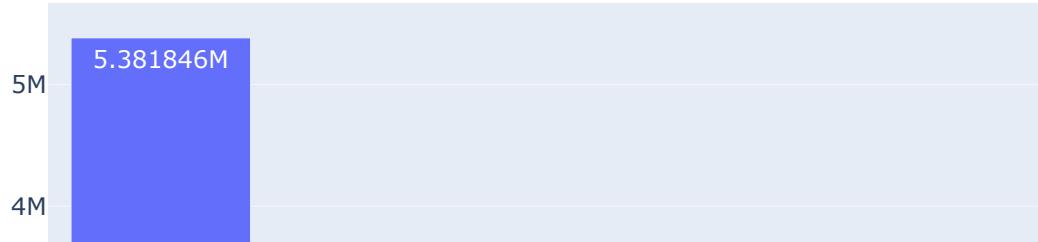


```
In [32]: # Q. in which country maximum people were died due to Alzheimer's Disease
# A. China
# Q. in which year death percentage was high
# A. the year was 1990
```

```
In [33]: data = df.groupby(['Country/Territory'])["Alzheimer's Disease and Other Dementias"]
```

```
In [34]: px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index,te
```

No. of People died from Alzheimer's Disease and Other Den



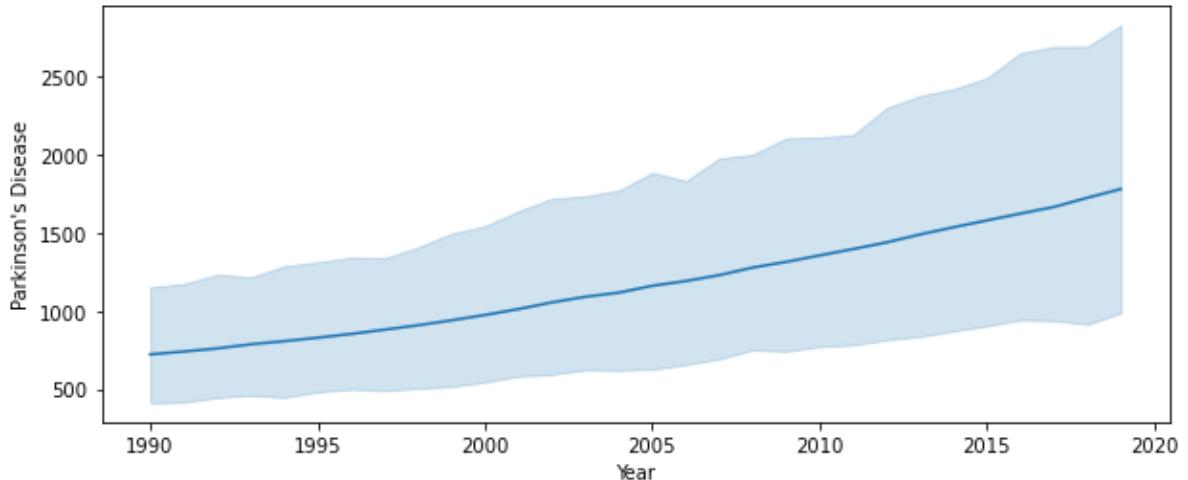
06. Parkinson's Disease - No. of People died from Parkinson's Disease

```
In [35]: df["Parkinson's Disease"].describe()
```

```
Out[35]: count    6120.000000
mean     1173.169118
std      4616.156238
min      0.000000
25%     27.000000
50%     164.000000
75%     609.250000
max     76990.000000
Name: Parkinson's Disease, dtype: float64
```

```
In [36]: plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Parkinson's Disease")
```

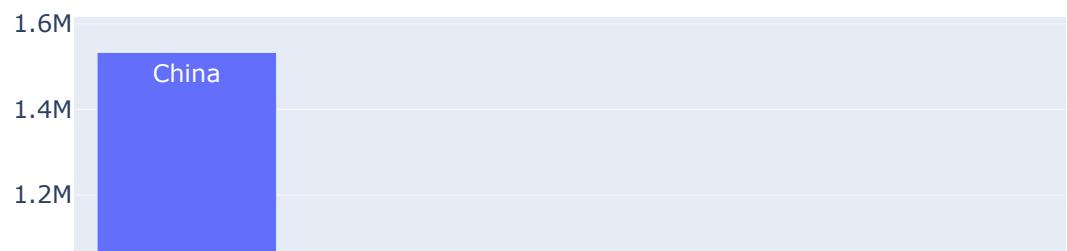
```
Out[36]: <AxesSubplot:xlabel='Year', ylabel="Parkinson's Disease">
```



```
In [37]: data = df.groupby(['Country/Territory'])["Parkinson's Disease"].sum().sort_values():
```

```
In [38]: px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index,ti:
```

Parkinson's Disease - No. of People died from Parkinson's Disease



07 Nutritional Deficiencies - No. of People died from Nutritional Deficiencies

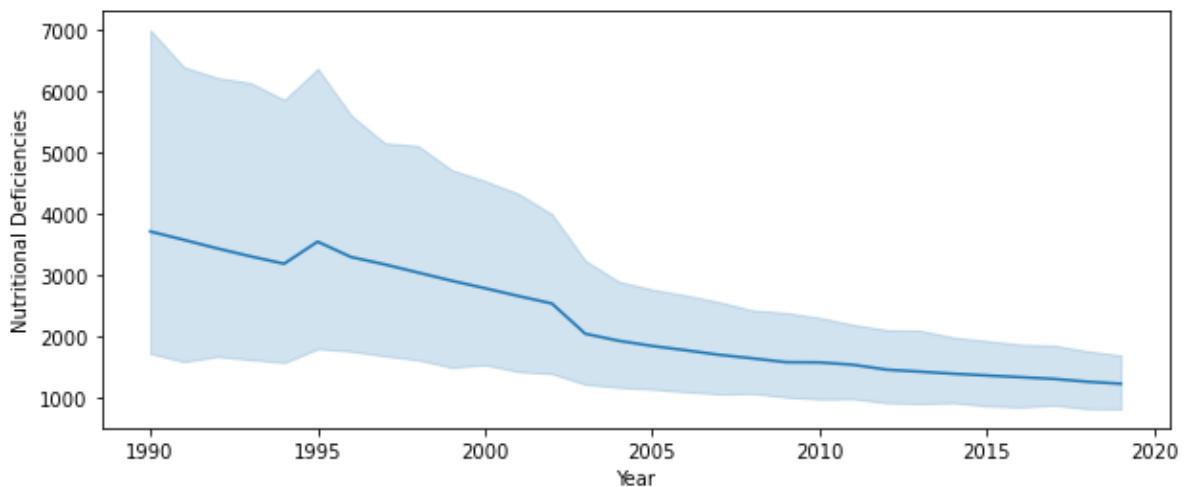
```
In [39]: df["Nutritional Deficiencies"].describe():
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
Out[39]: count    6120.000000
mean     2253.600000
std      10483.633601
min      0.000000
25%     9.000000
50%    119.000000
75%   1167.250000
max   268223.000000
Name: Nutritional Deficiencies, dtype: float64
```

```
In [40]: plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Nutritional Deficiencies")
```

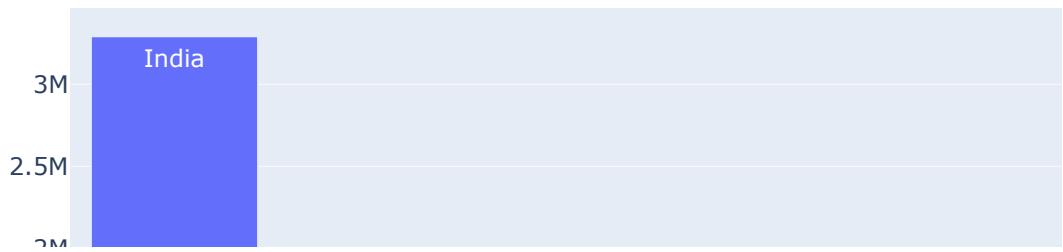
```
Out[40]: <AxesSubplot:xlabel='Year', ylabel='Nutritional Deficiencies'>
```



```
In [41]: data = df.groupby(['Country/Territory'])["Nutritional Deficiencies"].sum().sort_values(ascending=False)
```

```
In [42]: px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index, title="Nutritional Deficiencies")
```

No. of People died from Nutritional Deficiencies



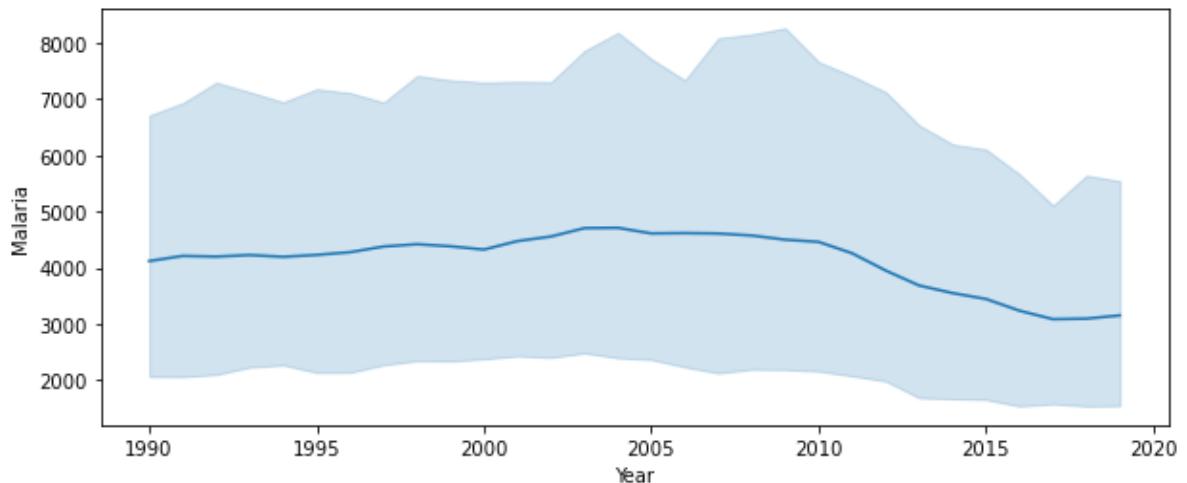
08. Malaria - No. of People died from Malaria

```
In [43]: df["Malaria"].describe()
```

```
Out[43]: count      6120.000000
mean       4140.960131
std        18427.753137
min         0.000000
25%        0.000000
50%        0.000000
75%        393.000000
max       280604.000000
Name: Malaria, dtype: float64
```

```
In [44]: plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Malaria")
```

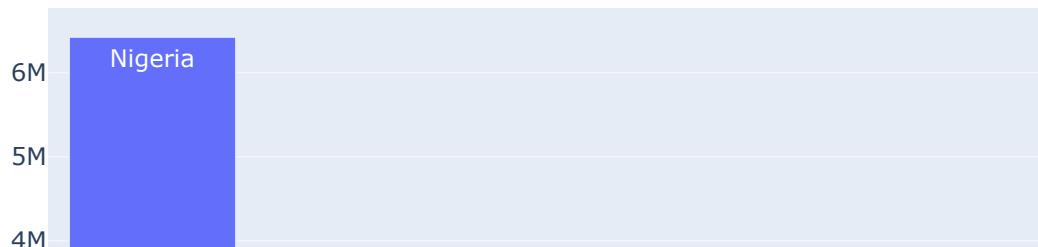
```
Out[44]: <AxesSubplot:xlabel='Year', ylabel='Malaria'>
```



```
In [45]: data = df.groupby(['Country/Territory'])["Malaria"].sum().sort_values(ascending = False)
```

```
In [46]: px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index, title="No. of People died from Malaria")
```

No. of People died from Malaria



09. Drowning - No. of People died from Drowning

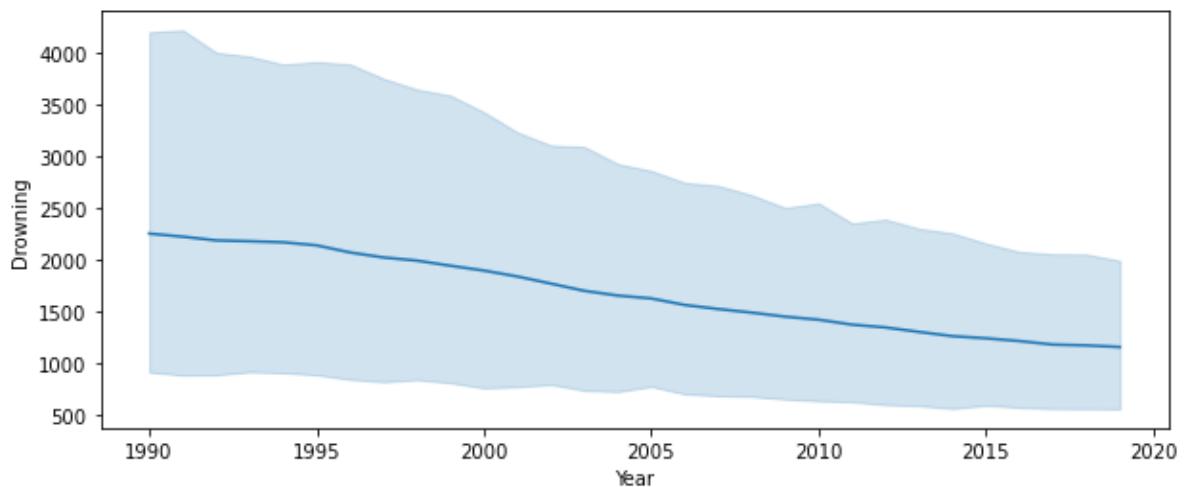
```
In [47]: df["Drowning"].describe()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
Out[47]: count    6120.000000
          mean     1683.333170
          std      8877.018366
          min      0.000000
          25%     34.000000
          50%     177.000000
          75%     698.000000
          max     153773.000000
          Name: Drowning, dtype: float64
```

```
In [48]: plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Drowning")
```

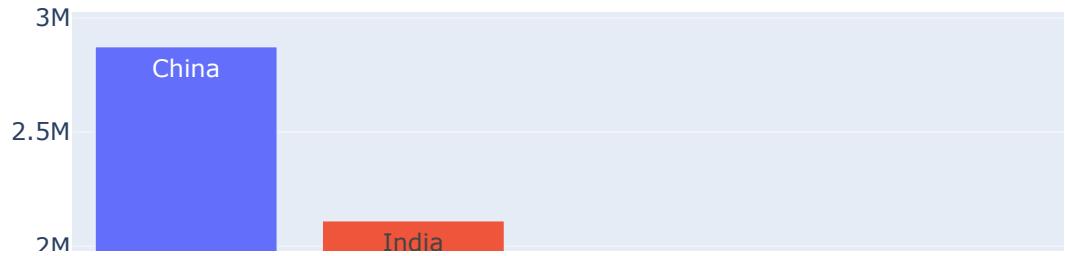
```
Out[48]: <AxesSubplot:xlabel='Year', ylabel='Drowning'>
```



```
In [49]: data = df.groupby(['Country/Territory'])["Drowning"].sum().sort_values(ascending = True)
```

```
In [50]: px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index, title="Cause Of Death")
```

No. of People died from Drowning



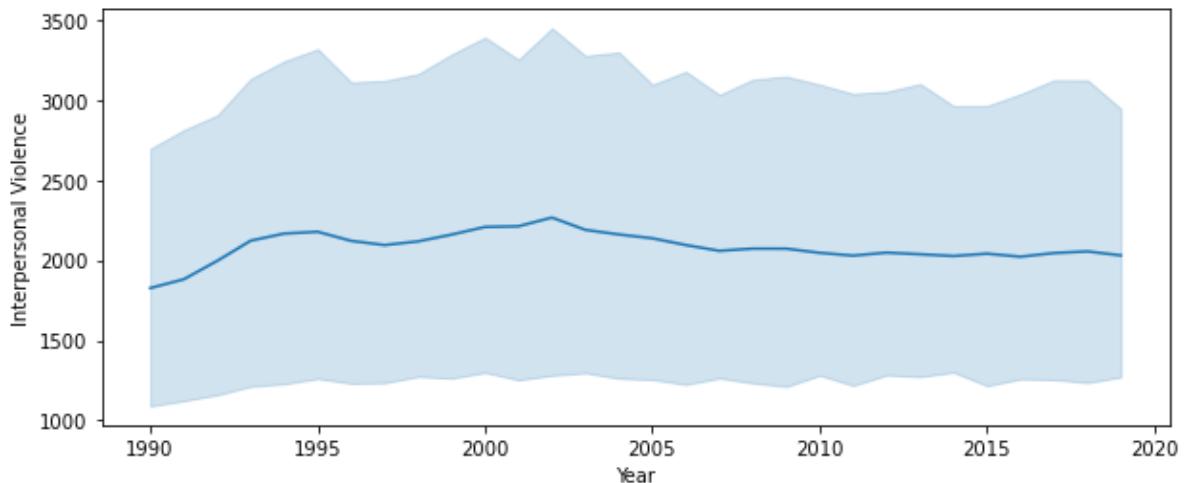
10. Interpersonal Violence - No. of People died from Interpersonal Violence

```
In [51]: df["Interpersonal Violence"].describe()
```

```
Out[51]: count    6120.000000
mean     2083.797222
std      6917.006075
min      0.000000
25%     40.000000
50%     265.000000
75%     877.000000
max     69640.000000
Name: Interpersonal Violence, dtype: float64
```

```
In [52]: plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Interpersonal Violence")
```

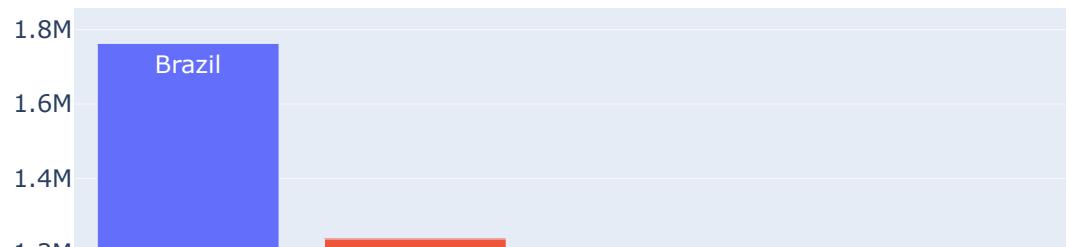
```
Out[52]: <AxesSubplot:xlabel='Year', ylabel='Interpersonal Violence'>
```



```
In [53]: data = df.groupby(['Country/Territory'])["Interpersonal Violence"].sum().sort_values(ascending=False)
```

```
In [54]: px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index, title="No. of People died from Interpersonal Violence")
```

No. of People died from Interpersonal Violence



11. Maternal Disorders - No. of People died from Maternal Disorders

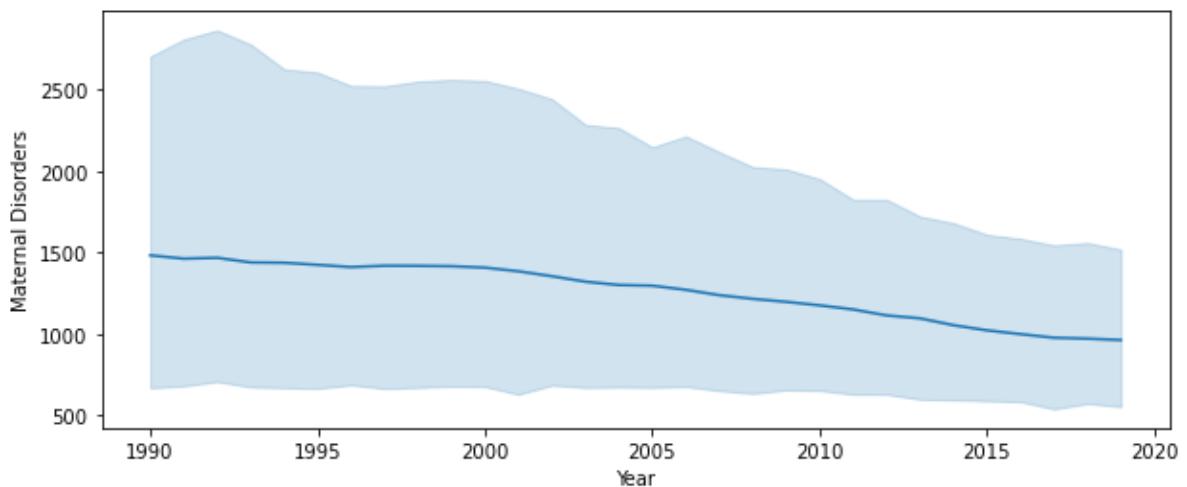
```
In [55]: df["Maternal Disorders"].describe()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
Out[55]: count    6120.000000
mean     1262.589216
std      6057.973183
min      0.000000
25%      5.000000
50%      54.000000
75%      734.000000
max     107929.000000
Name: Maternal Disorders, dtype: float64
```

```
In [56]: plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Maternal Disorders")
```

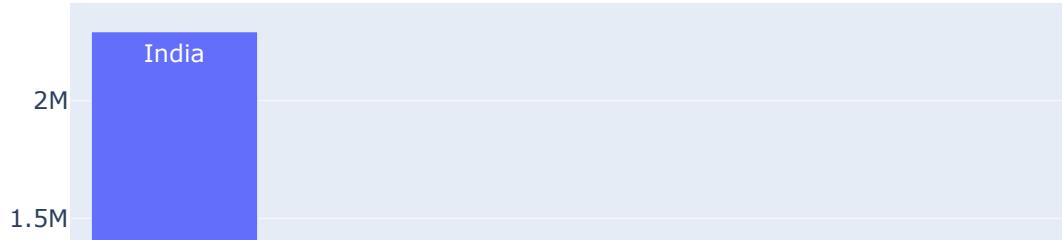
```
Out[56]: <AxesSubplot:xlabel='Year', ylabel='Maternal Disorders'>
```



```
In [57]: data = df.groupby(['Country/Territory'])["Maternal Disorders"].sum().sort_values(ascending=False)
```

```
In [58]: px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index, title="Maternal Disorders Deaths")
```

Maternal Disorders - No. of People died from Maternal Disorders



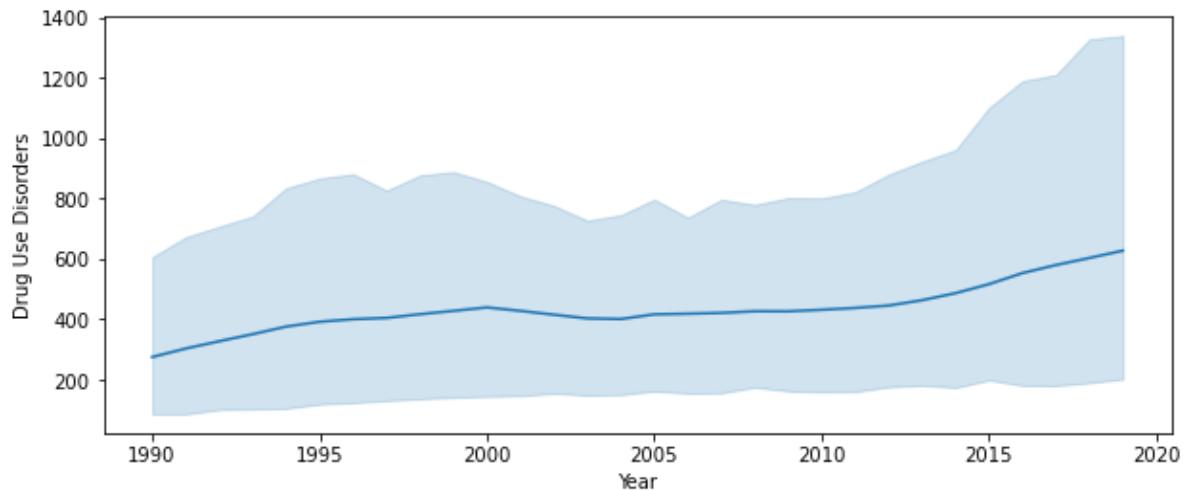
12. Drug Use Disorders - No. of People died from Drug Use Disorders

```
In [59]: df["Drug Use Disorders"].describe()
```

```
Out[59]: count    6120.000000
mean      434.006699
std       2898.761628
min       0.000000
25%      3.000000
50%     20.000000
75%    129.000000
max     65717.000000
Name: Drug Use Disorders, dtype: float64
```

```
In [60]: plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Drug Use Disorders")
```

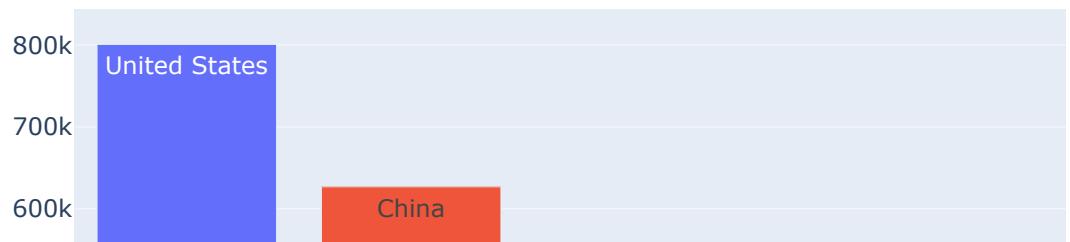
```
Out[60]: <AxesSubplot:xlabel='Year', ylabel='Drug Use Disorders'>
```



```
In [61]: data = df.groupby(['Country/Territory'])["Drug Use Disorders"].sum().sort_values(ascending=False)
```

```
In [62]: px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index, title="Drug Use Disorders - No. of People died from Drug Use Disorders")
```

Drug Use Disorders - No. of People died from Drug Use Disorders



13. Tuberculosis - No. of People died from Tuberculosis

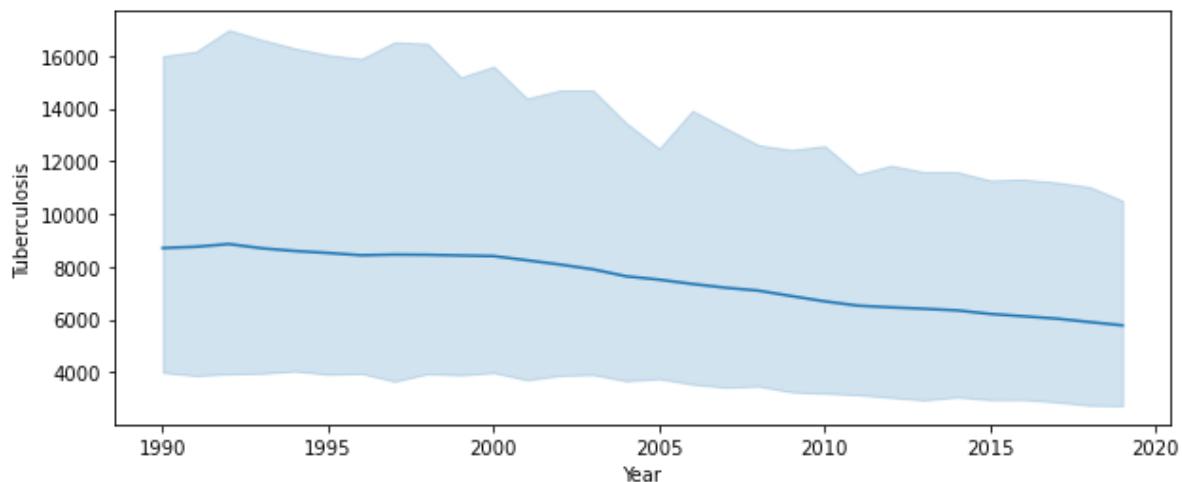
```
In [63]: df["Tuberculosis"].describe()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
Out[63]: count      6120.000000
mean       7491.928595
std        39549.977578
min         0.000000
25%        35.000000
50%        417.000000
75%       2924.250000
max       657515.000000
Name: Tuberculosis, dtype: float64
```

```
In [64]: plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Tuberculosis")
```

```
Out[64]: <AxesSubplot:xlabel='Year', ylabel='Tuberculosis'>
```



```
In [65]: data = df.groupby(['Country/Territory'])["Tuberculosis"].sum().sort_values(ascending=True)
```

```
In [66]: px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index,ti...
```

Tuberculosis - No. of People died from Tuberculosis



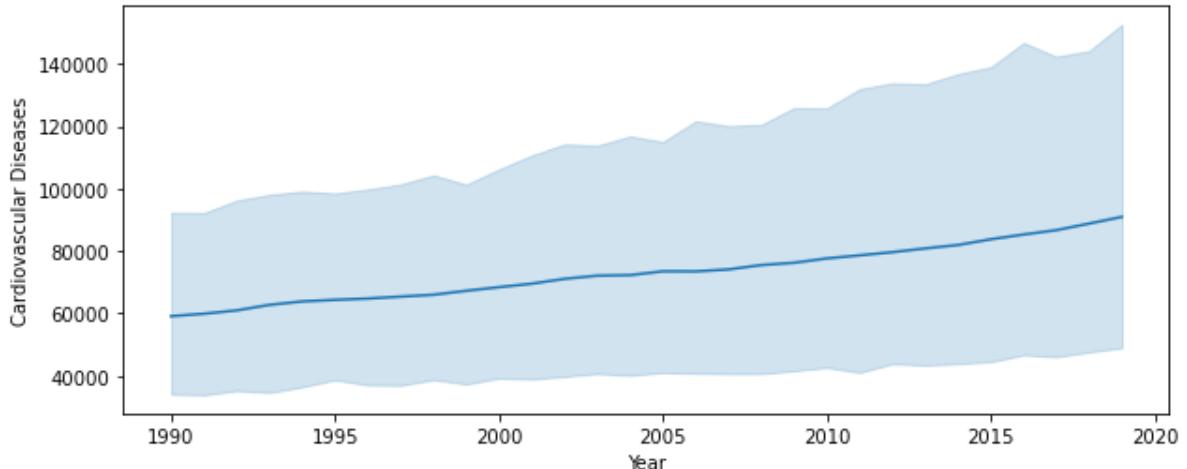
14. Cardiovascular Diseases - No. of People died from Cardiovascular Diseases

```
In [67]: df["Cardiovascular Diseases"].describe()
```

```
Out[67]: count    6.120000e+03
mean      7.316045e+04
std       2.915775e+05
min       4.000000e+00
25%      2.028000e+03
50%      1.174200e+04
75%      4.254650e+04
max      4.584273e+06
Name: Cardiovascular Diseases, dtype: float64
```

```
In [68]: plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Cardiovascular Diseases")
```

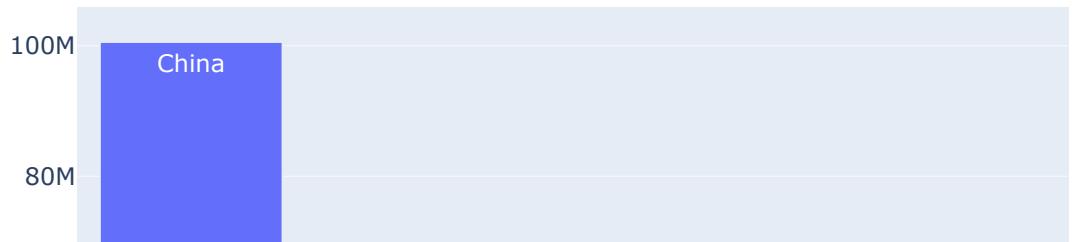
```
Out[68]: <AxesSubplot:xlabel='Year', ylabel='Cardiovascular Diseases'>
```



```
In [69]: data = df.groupby(['Country/Territory'])["Cardiovascular Diseases"].sum().sort_values(ascending=False)
```

```
In [70]: px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index, title="Cardiovascular Diseases - No. of People died from Cardiovascular Diseases")
```

Cardiovascular Diseases - No. of People died from Cardiovascular Diseases



15. Lower Respiratory Infections - No. of People died from Lower Respiratory

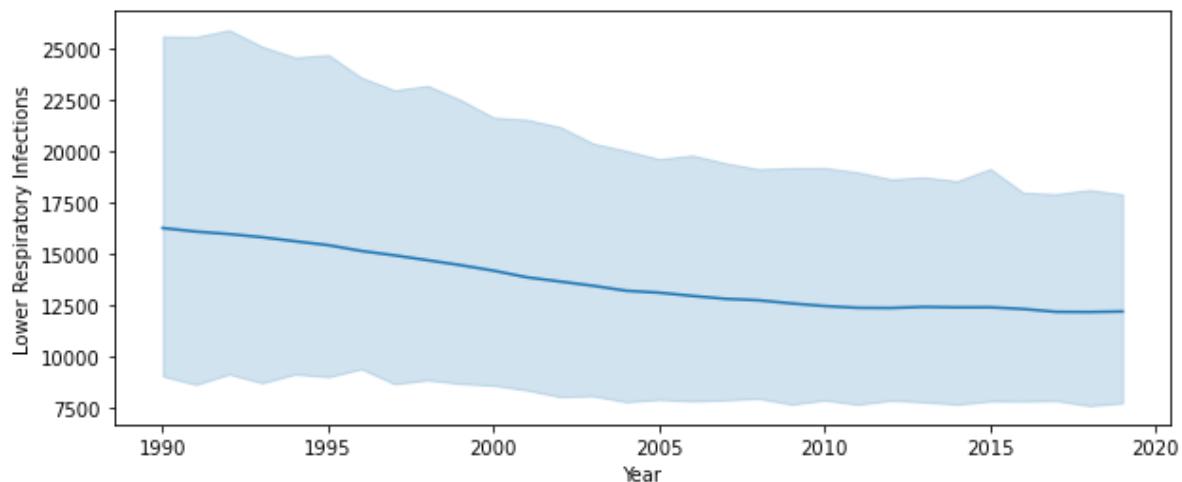
```
In [71]: df["Lower Respiratory Infections"].describe()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
Out[71]: count      6120.000000
mean       13687.914706
std        48031.720009
min         0.000000
25%       345.000000
50%       2126.500000
75%      10161.250000
max      690913.000000
Name: Lower Respiratory Infections, dtype: float64
```

```
In [72]: plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Lower Respiratory Infections")
```

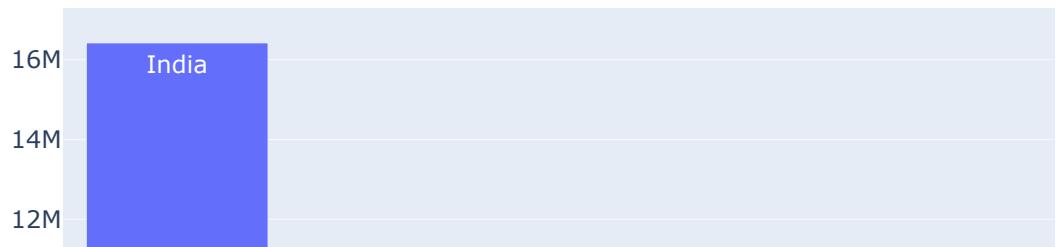
```
Out[72]: <AxesSubplot:xlabel='Year', ylabel='Lower Respiratory Infections'>
```



```
In [73]: data = df.groupby(['Country/Territory'])["Lower Respiratory Infections"].sum().sort_values(ascending=False)
```

```
In [74]: px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index, title="Lower Respiratory Infections")
```

Lower Respiratory Infections - No. of People died from Lower Respiratory Infections



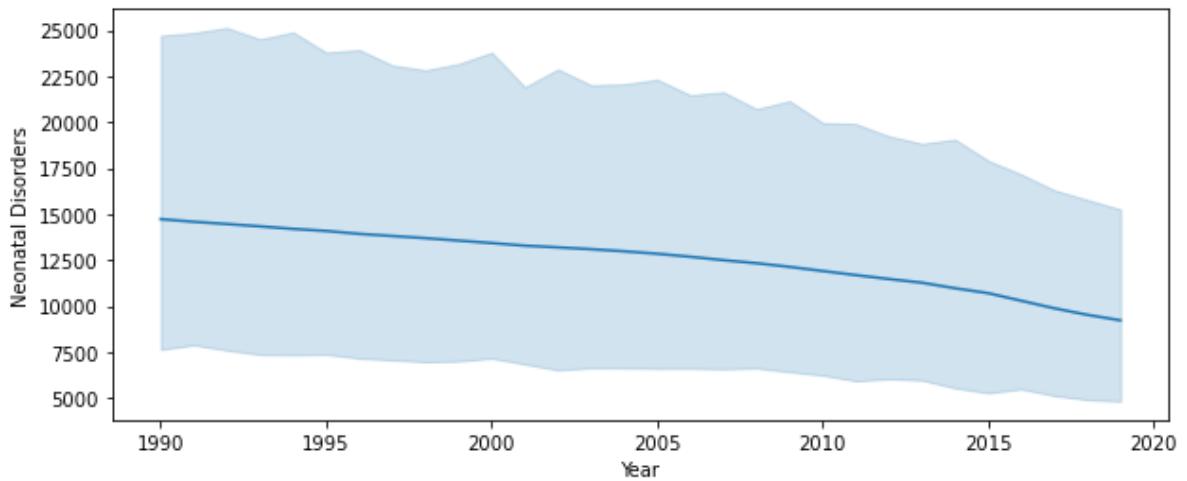
16. Neonatal Disorders - No. of People died from Neonatal Disorders

```
In [75]: df["Neonatal Disorders"].describe()
```

```
Out[75]: count      6120.000000
mean       12558.942647
std        56058.366412
min         0.000000
25%      131.000000
50%      916.000000
75%     7419.750000
max     852761.000000
Name: Neonatal Disorders, dtype: float64
```

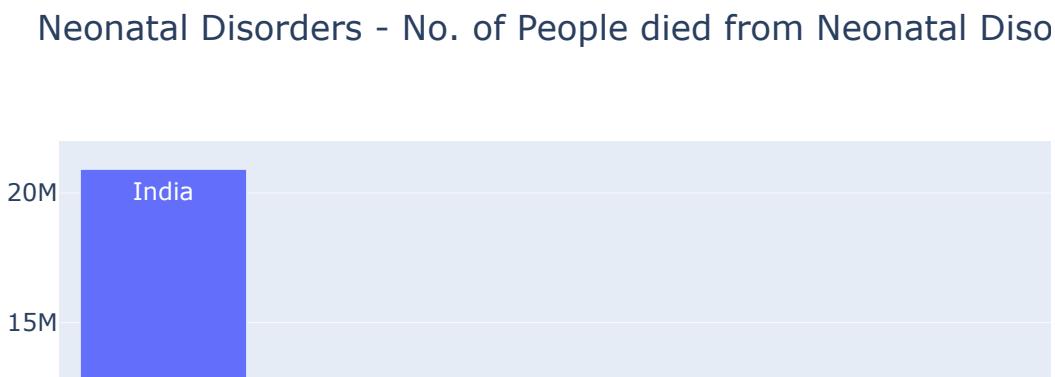
```
In [76]: plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Neonatal Disorders")
```

```
Out[76]: <AxesSubplot:xlabel='Year', ylabel='Neonatal Disorders'>
```



```
In [77]: data = df.groupby(['Country/Territory'])["Neonatal Disorders"].sum().sort_values(ascending=False)
```

```
In [78]: px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index, title="Neonatal Disorders - No. of People died from Neonatal Disorders")
```



17. Alcohol Use Disorders - No. of People died from Alcohol Use Disorders

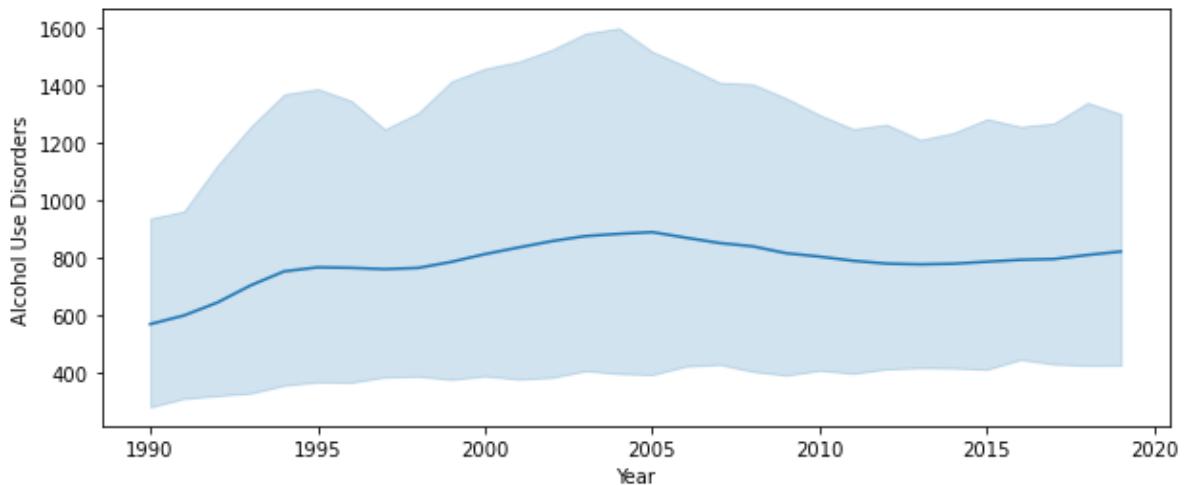
```
In [79]: df["Alcohol Use Disorders"].describe()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
Out[79]: count    6120.000000
          mean     787.421242
          std      3545.823616
          min      0.000000
          25%     9.000000
          50%    80.000000
          75%   316.000000
          max   55200.000000
Name: Alcohol Use Disorders, dtype: float64
```

```
In [80]: plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Alcohol Use Disorders")
```

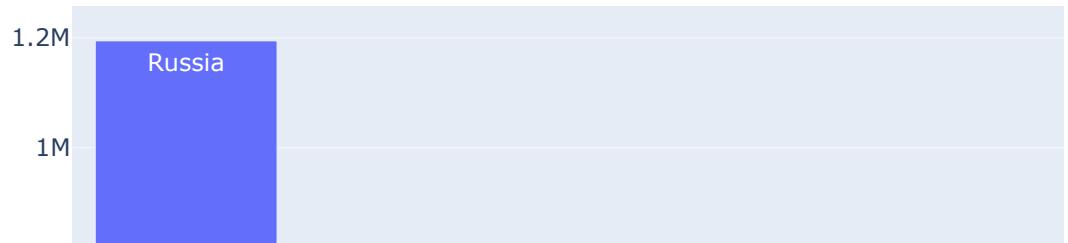
```
Out[80]: <AxesSubplot:xlabel='Year', ylabel='Alcohol Use Disorders'>
```



```
In [81]: data = df.groupby(['Country/Territory'])["Alcohol Use Disorders"].sum().sort_values()
```

```
In [82]: px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index,ti
```

Alcohol Use Disorders - No. of People died from Alcohol Use



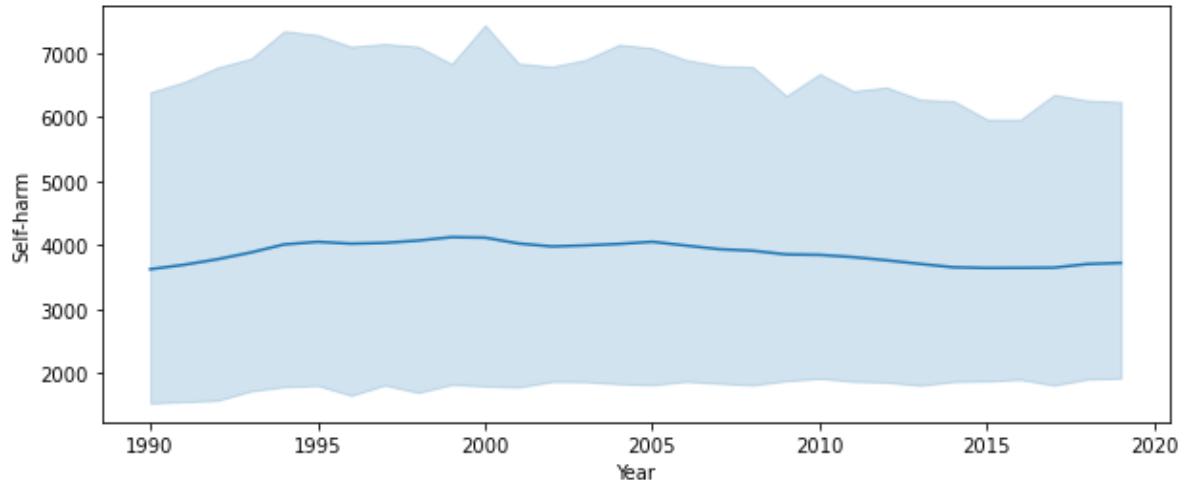
18. Self-harm - No. of People died from Self-harm

```
In [83]: df["Self-harm"].describe()
```

```
Out[83]: count      6120.000000
mean       3874.825327
std        18425.616418
min        0.000000
25%       94.000000
50%      533.000000
75%     1882.250000
max      220357.000000
Name: Self-harm, dtype: float64
```

```
In [84]: plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Self-harm")
```

```
Out[84]: <AxesSubplot:xlabel='Year', ylabel='Self-harm'>
```



```
In [85]: data = df.groupby(['Country/Territory'])["Self-harm"].sum().sort_values(ascending=True)
```

```
In [86]: px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index,ti
```

Self-harm - No. of People died from Self-harm



19. Exposure to Forces of Nature - No. of People died from Exposure to Forces of Nature

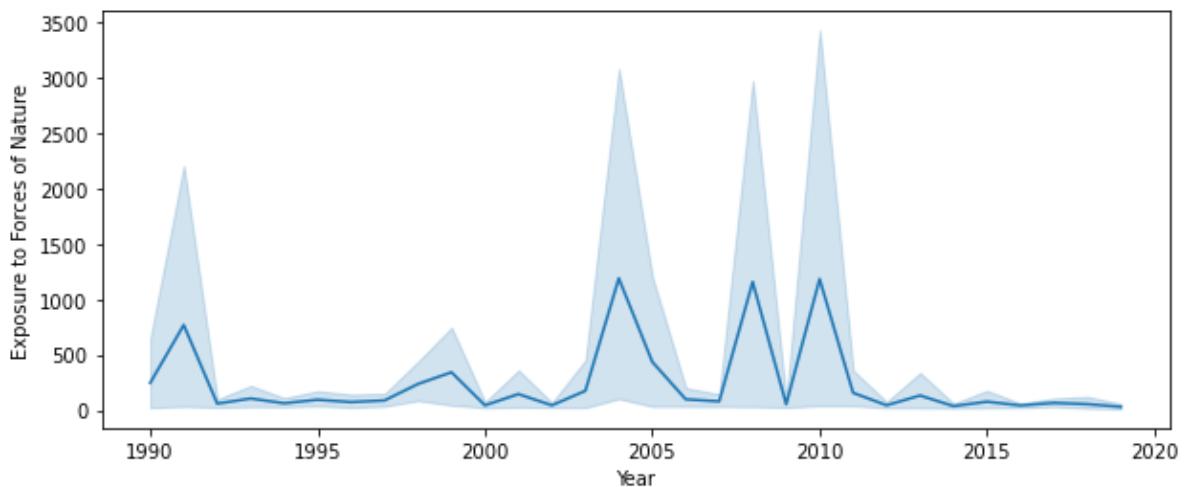
```
In [87]: df["Exposure to Forces of Nature"].describe()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
Out[87]: count      6120.000000
mean       243.485621
std        4717.104377
min         0.000000
25%        0.000000
50%        0.000000
75%       12.000000
max      222641.000000
Name: Exposure to Forces of Nature, dtype: float64
```

```
In [88]: plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Exposure to Forces of Nature")
```

```
Out[88]: <AxesSubplot:xlabel='Year', ylabel='Exposure to Forces of Nature'>
```



```
In [89]: data = df.groupby(['Country/Territory'])["Exposure to Forces of Nature"].sum().sort_index()
```

```
In [90]: px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index, title="Exposure to Forces of Nature by Country/Territory")
```

Exposure to Forces of Nature - No. of People died from Exp



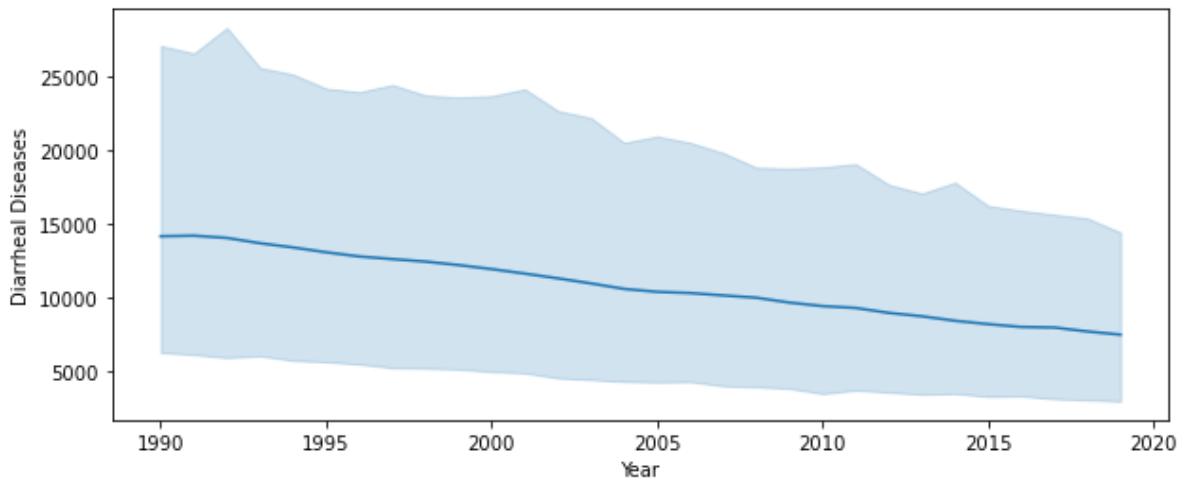
20. Diarrheal Diseases - No. of People died from Diarrheal Diseases

```
In [91]: df["Diarrheal Diseases"].describe()
```

```
Out[91]: count    6.120000e+03
mean      1.082280e+04
std       6.541617e+04
min       0.000000e+00
25%      2.000000e+01
50%      2.965000e+02
75%      3.946750e+03
max      1.119477e+06
Name: Diarrheal Diseases, dtype: float64
```

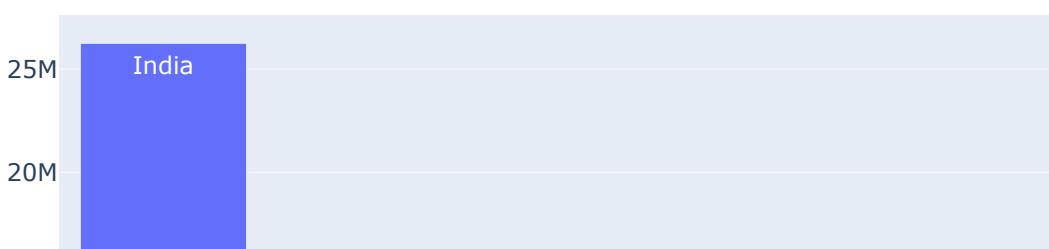
```
In [92]: plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Diarrheal Diseases")
```

```
Out[92]: <AxesSubplot:xlabel='Year', ylabel='Diarrheal Diseases'>
```



```
In [93]: data = df.groupby(['Country/Territory'])["Diarrheal Diseases"].sum().sort_values(ascending=False)
```

```
In [94]: px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index, title="Diarrheal Diseases - No. of People died from Diarrheal Diseases")
```



21. Environmental Heat and Cold Exposure - No. of People died from Environmental Heat and Cold Exposure

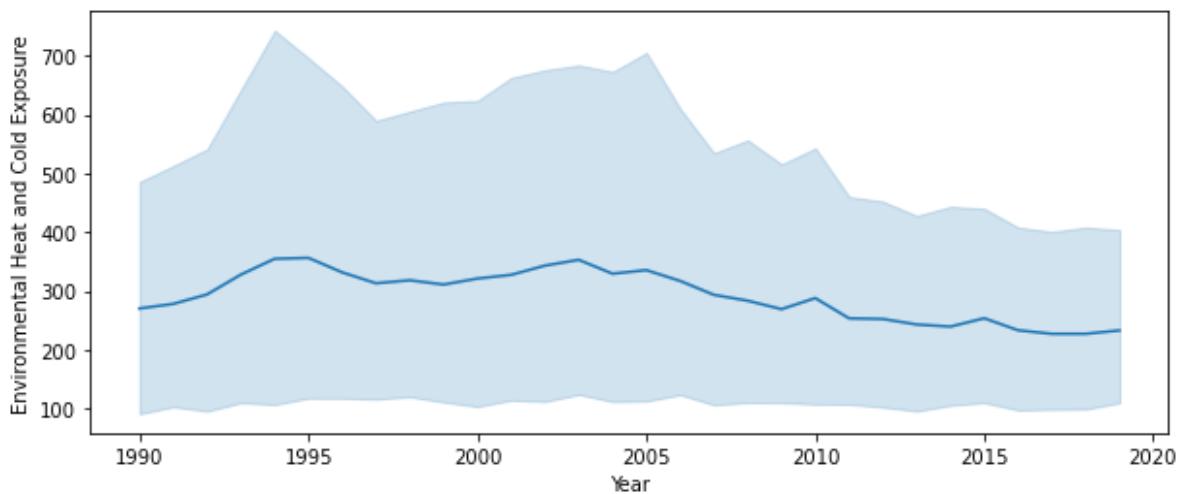
```
In [95]: df["Environmental Heat and Cold Exposure"].describe()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
Out[95]: count    6120.000000
mean     292.295915
std      1704.466356
min      0.000000
25%     2.000000
50%     21.000000
75%    109.000000
max    29048.000000
Name: Environmental Heat and Cold Exposure, dtype: float64
```

```
In [96]: plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Environmental Heat and Cold Exposure")
```

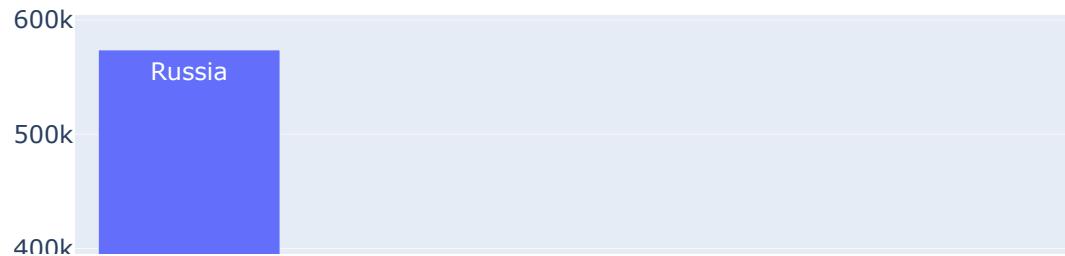
```
Out[96]: <AxesSubplot:xlabel='Year', ylabel='Environmental Heat and Cold Exposure'>
```



```
In [97]: data = df.groupby(['Country/Territory'])["Environmental Heat and Cold Exposure"].sum().reset_index()
```

```
In [98]: px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index, title="Environmental Heat and Cold Exposure by Country/Territory")
```

Environmental Heat and Cold Exposure - No. of People died



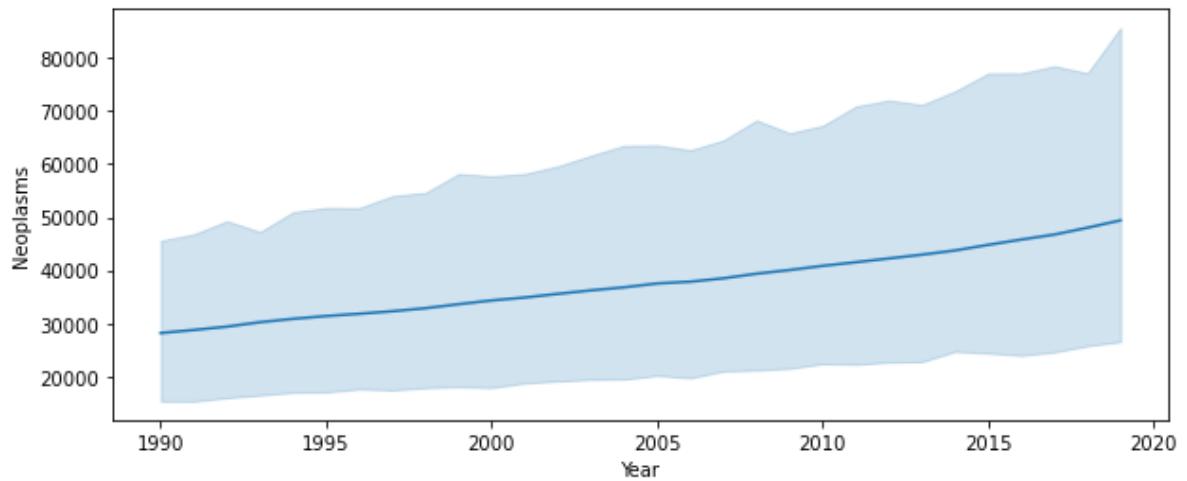
22. Neoplasms - No. of People died from Neoplasms

```
In [99]: df["Neoplasms"].describe()
```

```
Out[99]: count    6.120000e+03
          mean     3.754224e+04
          std      1.615584e+05
          min      1.000000e+00
          25%     8.097500e+02
          50%     5.629500e+03
          75%     2.014775e+04
          max     2.716551e+06
          Name: Neoplasms, dtype: float64
```

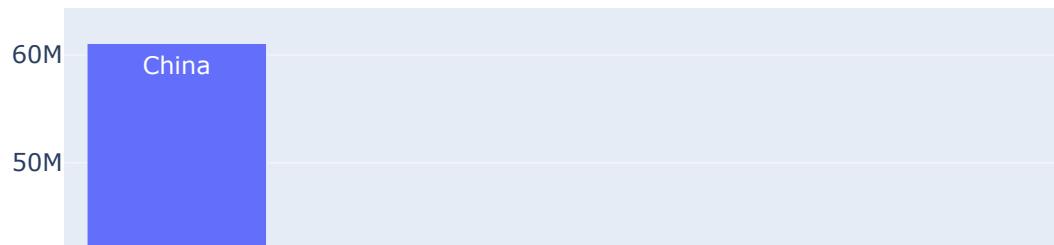
```
In [100...]: plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Neoplasms")
```

```
Out[100]: <AxesSubplot:xlabel='Year', ylabel='Neoplasms'>
```



```
In [101...]: data = df.groupby(['Country/Territory'])["Neoplasms"].sum().sort_values(ascending=True)
In [102...]: px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index,ti...
```

Neoplasms - No. of People died from Neoplasms



23. Conflict and Terrorism - No. of People died from Conflict and Terrorism

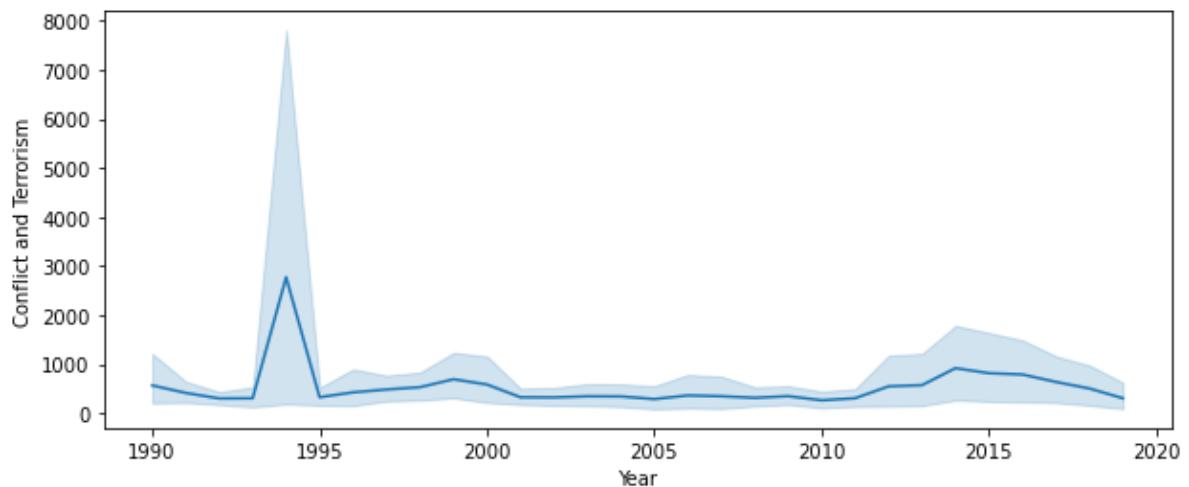
```
In [103...]: df[ "Conflict and Terrorism" ].describe()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
Out[103]: count      6120.000000
mean       538.243954
std        7033.308187
min        0.000000
25%       0.000000
50%       0.000000
75%      23.000000
max      503532.000000
Name: Conflict and Terrorism, dtype: float64
```

```
In [104... plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Conflict and Terrorism")
```

```
Out[104]: <AxesSubplot:xlabel='Year', ylabel='Conflict and Terrorism'>
```



```
In [105... data = df.groupby(['Country/Territory'])["Conflict and Terrorism"].sum().sort_values
```

```
In [106... px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index,ti
```

Conflict and Terrorism - No. of People died from Conflict and Terrorism



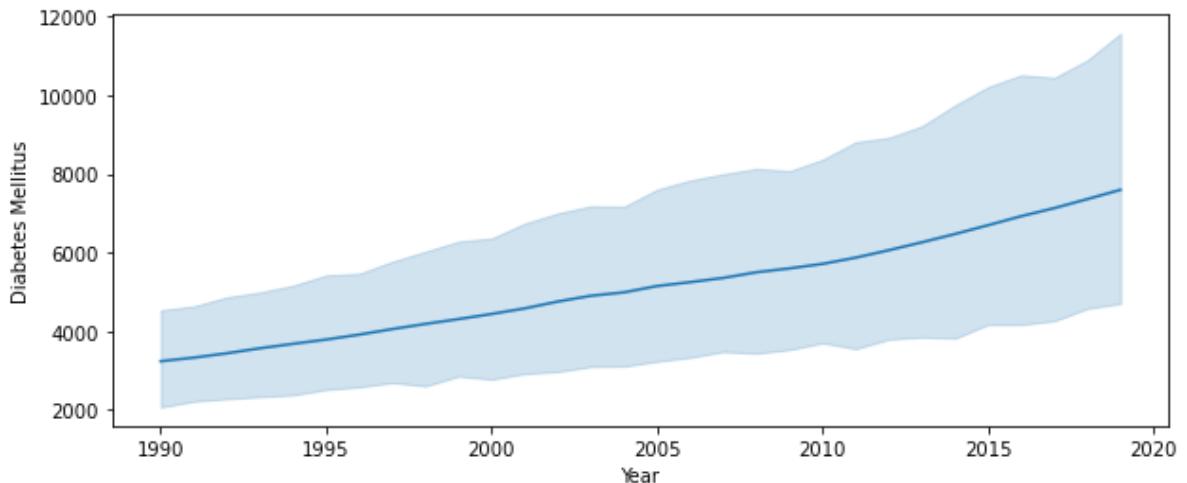
24. Diabetes Mellitus - No. of People died from Diabetes Mellitus

```
In [107]: df["Diabetes Mellitus"].describe()

Out[107]: count      6120.000000
            mean      5138.704575
            std       16773.081040
            min       1.000000
            25%      236.000000
            50%      1087.000000
            75%      2954.000000
            max      273089.000000
            Name: Diabetes Mellitus, dtype: float64

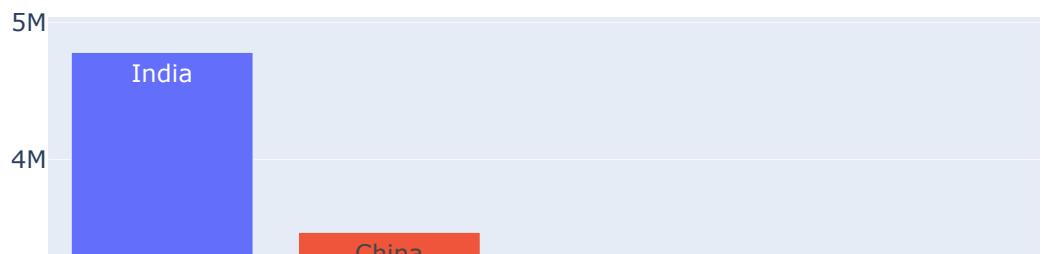
In [108]: plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Diabetes Mellitus")

Out[108]: <AxesSubplot:xlabel='Year', ylabel='Diabetes Mellitus'>
```



```
In [109...]: data = df.groupby(['Country/Territory'])["Diabetes Mellitus"].sum().sort_values(ascending=True)
In [110...]: px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index,ti...
```

Diabetes Mellitus - No. of People died from Diabetes Mellitus



25. Chronic Kidney Disease - No. of People died from Chronic Kidney Disease

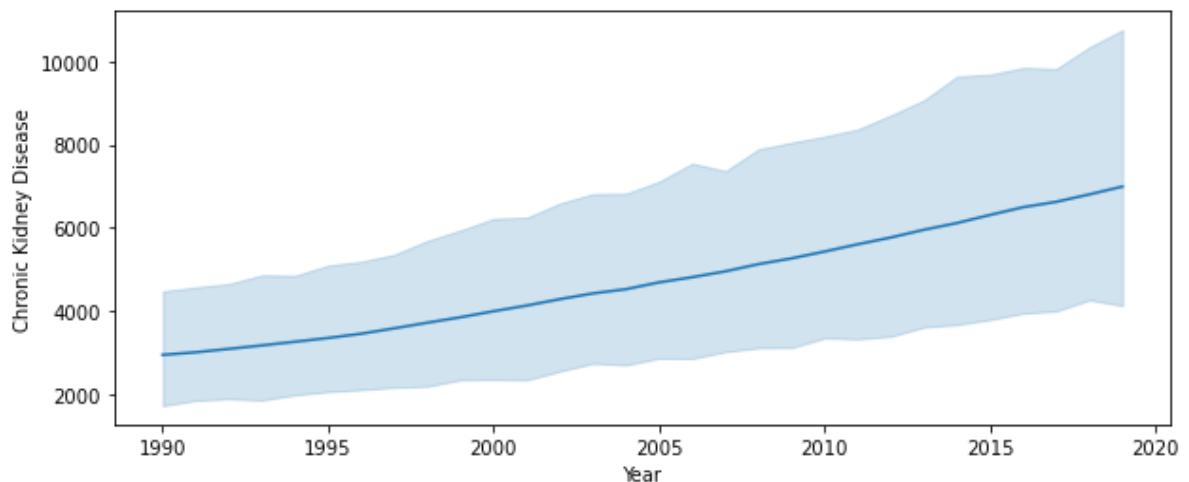
```
In [111...]: df["Chronic Kidney Disease"].describe()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
Out[111]: count      6120.000000
mean       4724.132680
std        16470.429969
min         0.000000
25%       145.750000
50%       822.000000
75%      2922.500000
max      222922.000000
Name: Chronic Kidney Disease, dtype: float64
```

```
In [112... plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Chronic Kidney Disease")
```

```
Out[112]: <AxesSubplot:xlabel='Year', ylabel='Chronic Kidney Disease'>
```



```
In [113... data = df.groupby(['Country/Territory'])["Chronic Kidney Disease"].sum().sort_values
```

```
In [114... px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index,ti
```

Chronic Kidney Disease - No. of People died from Chronic K



26. Poisonings - No. of People died from Poisoning

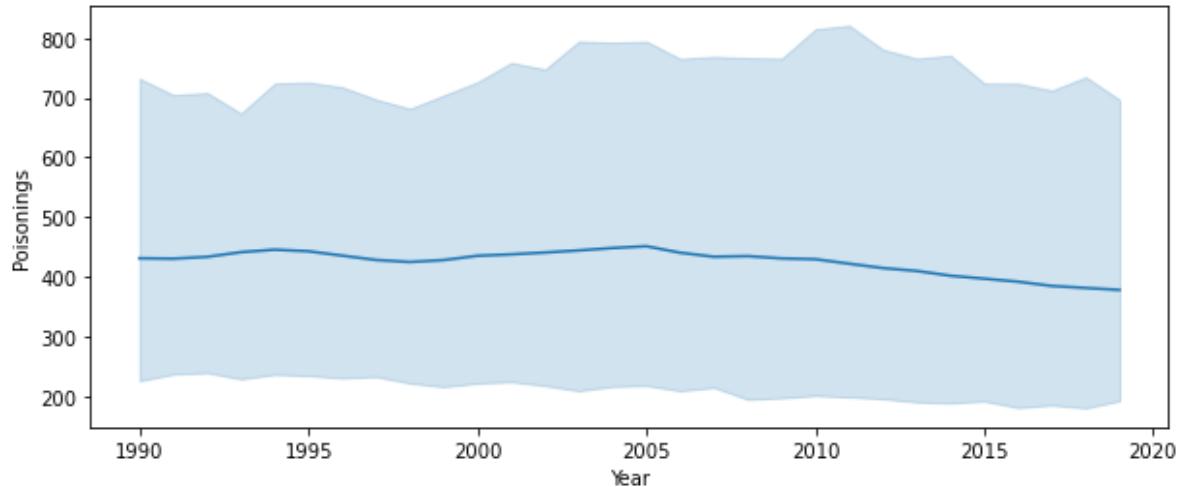
```
In [115]: df["Poisonings"].describe()
```

```
Out[115]:
```

count	6120.000000
mean	425.013399
std	2022.640521
min	0.000000
25%	6.000000
50%	52.500000
75%	254.000000
max	30883.000000
Name:	Poisonings, dtype: float64

```
In [116]: plt.figure(figsize=(10,4))  
sns.lineplot(data=df, x="Year", y="Poisonings")
```

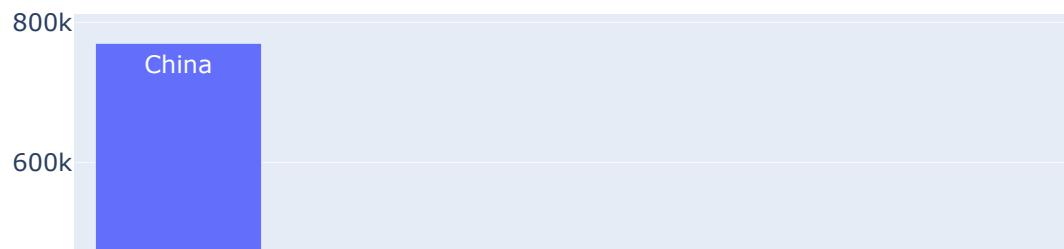
```
Out[116]: <AxesSubplot:xlabel='Year', ylabel='Poisonings'>
```



```
In [117...]: data = df.groupby(['Country/Territory'])["Poisonings"].sum().sort_values(ascending=True)
```

```
In [118...]: px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index,ti...
```

Poisonings - No. of People died from Poisoning



27. Protein-Energy Malnutrition - No. of People died from Protein-Energy Malnutrition

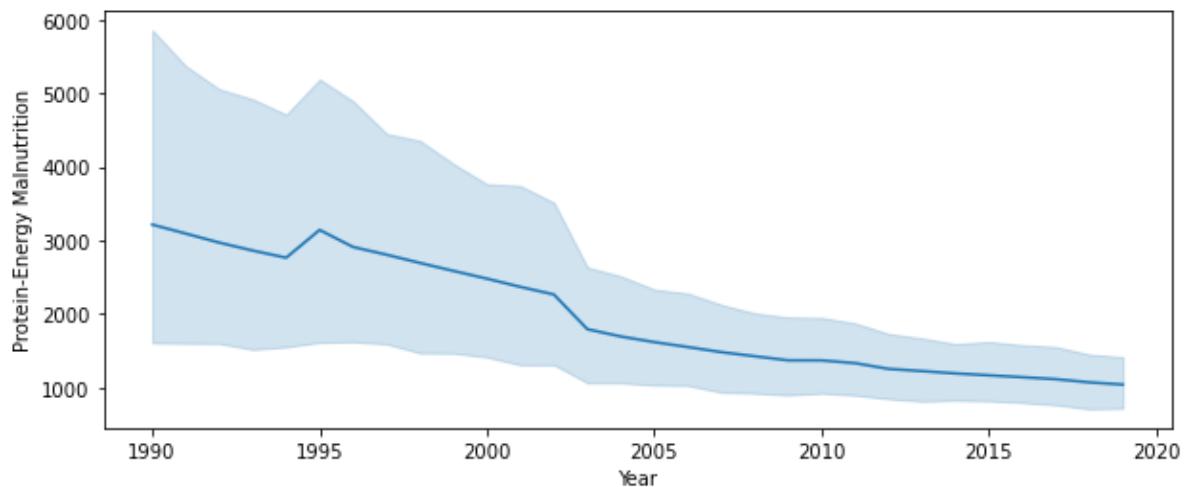
```
In [119...]: df["Protein-Energy Malnutrition"].describe()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
Out[119]: count    6120.000000
mean     1965.994281
std      8255.999063
min      0.000000
25%     5.000000
50%     92.000000
75%    1042.500000
max    202241.000000
Name: Protein-Energy Malnutrition, dtype: float64
```

```
In [120... plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Protein-Energy Malnutrition")
```

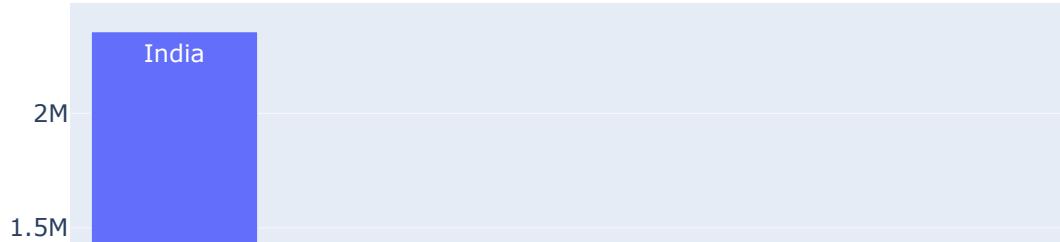
```
Out[120]: <AxesSubplot:xlabel='Year', ylabel='Protein-Energy Malnutrition'>
```



```
In [121... data = df.groupby(['Country/Territory'])["Protein-Energy Malnutrition"].sum().sort_
```

```
In [122... px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index,ti
```

Protein-Energy Malnutrition - No. of People died from Protein-Energy Malnutrition



28. Chronic Respiratory Diseases - No. of People died from Chronic Respiratory Diseases

```
In [123]: df["Chronic Respiratory Diseases"].describe()
```

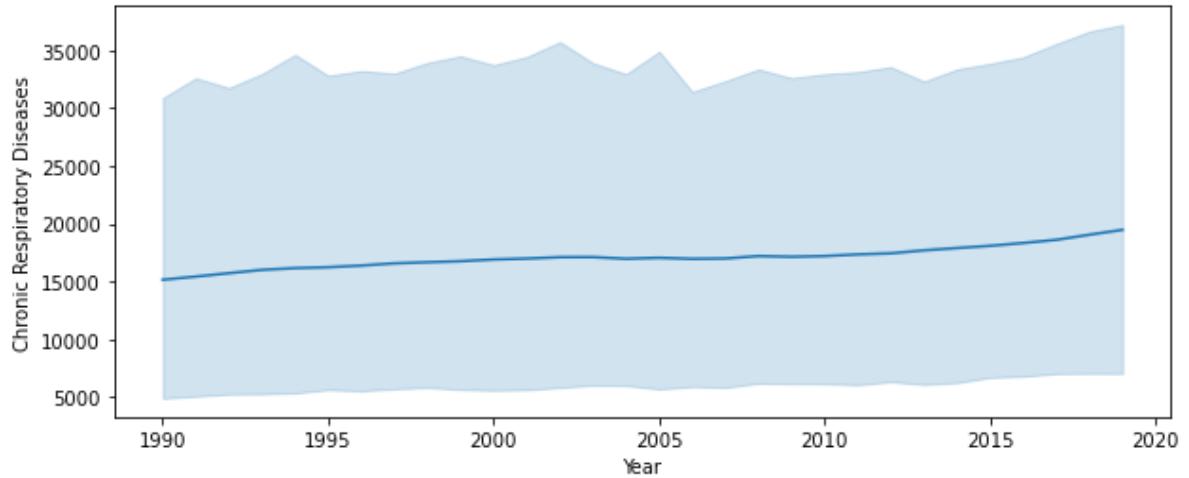
```
Out[123]:
```

count	6.120000e+03
mean	1.709237e+04
std	1.051572e+05
min	1.000000e+00
25%	2.890000e+02
50%	1.689000e+03
75%	5.249750e+03
max	1.366039e+06

Name: Chronic Respiratory Diseases, dtype: float64

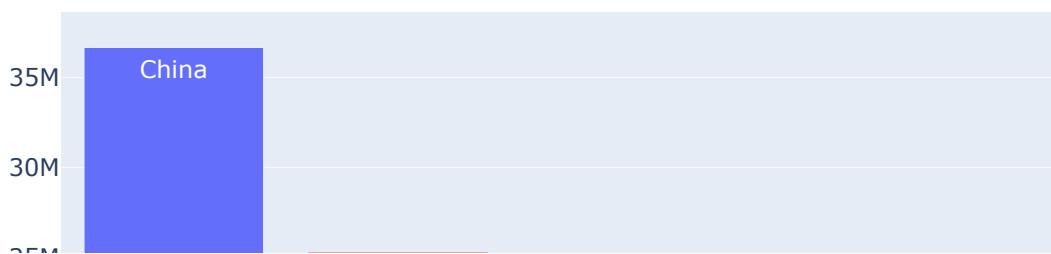
```
In [124]: plt.figure(figsize=(10,4))  
sns.lineplot(data=df, x="Year", y="Chronic Respiratory Diseases")
```

```
Out[124]: <AxesSubplot:xlabel='Year', ylabel='Chronic Respiratory Diseases'>
```



```
In [125...]: data = df.groupby(['Country/Territory'])["Chronic Respiratory Diseases"].sum().sort_index(ascending=True)
```

```
In [126...]: px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index, title="Chronic Respiratory Diseases - No. of People died from Chronic Respiratory Diseases")
```



29. Cirrhosis and Other Chronic Liver Diseases - No. of People died from Cirrhosis and Other Chronic Liver Diseases

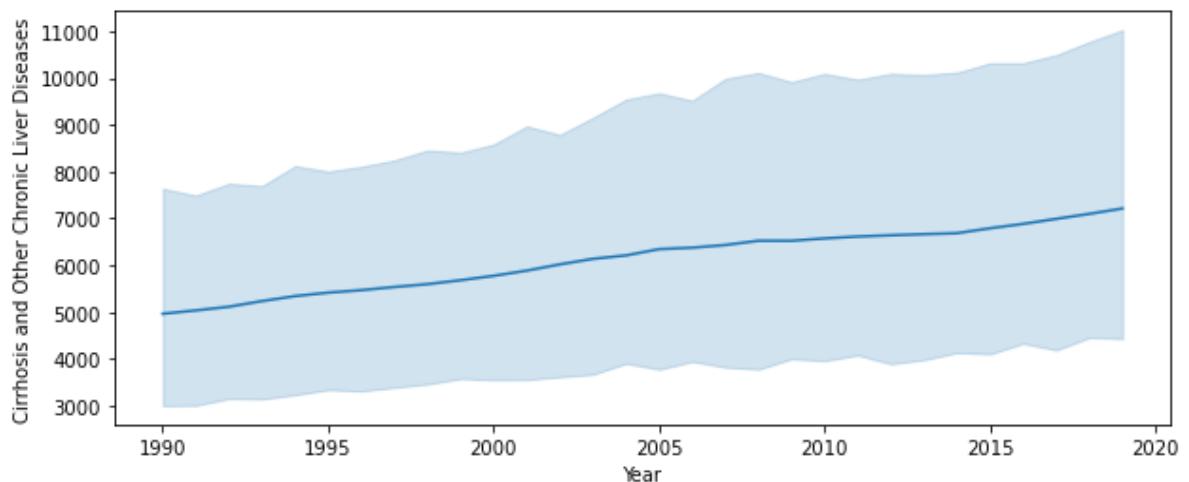
```
In [127...]: df["Cirrhosis and Other Chronic Liver Diseases"].describe()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
Out[127]: count    6120.000000
mean     6124.072059
std      20688.118580
min      0.000000
25%     154.000000
50%     1210.000000
75%     3547.250000
max     270037.000000
Name: Cirrhosis and Other Chronic Liver Diseases, dtype: float64
```

```
In [128... plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Cirrhosis and Other Chronic Liver Diseases")
```

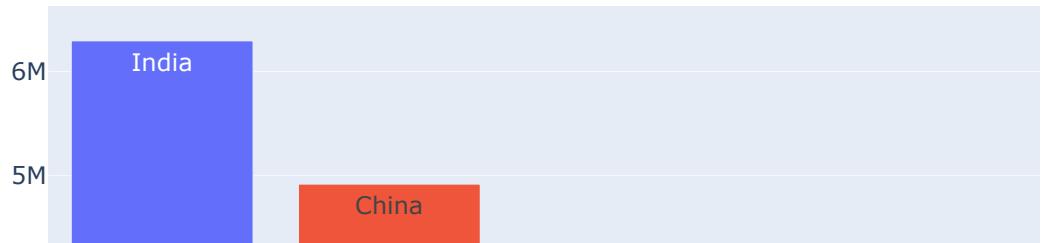
```
Out[128]: <AxesSubplot:xlabel='Year', ylabel='Cirrhosis and Other Chronic Liver Diseases'>
```



```
In [129... data = df.groupby(['Country/Territory'])["Cirrhosis and Other Chronic Liver Disease
```

```
In [130... px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index,ti
```

Cirrhosis and Other Chronic Liver Diseases - No. of People died from Cirrhosis and Other Chronic Liver Diseases



30. Digestive Diseases - No. of People died from Digestive Diseases

```
In [131]: df["Digestive Diseases"].describe()
```

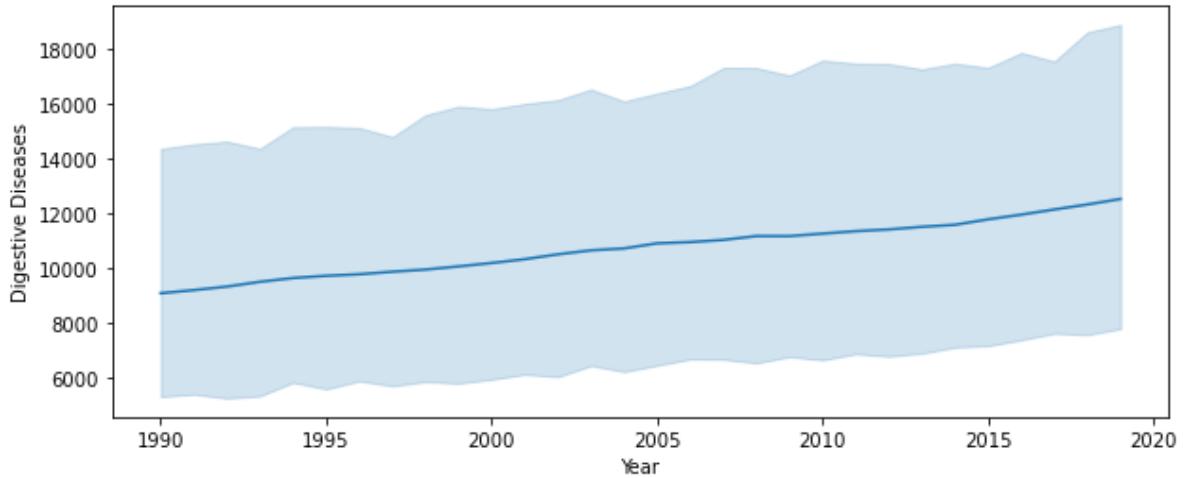
```
Out[131]:
```

count	6120.000000
mean	10725.267157
std	37228.051096
min	0.000000
25%	284.000000
50%	2185.000000
75%	6080.000000
max	464914.000000

Name: Digestive Diseases, dtype: float64

```
In [132]: plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Digestive Diseases")
```

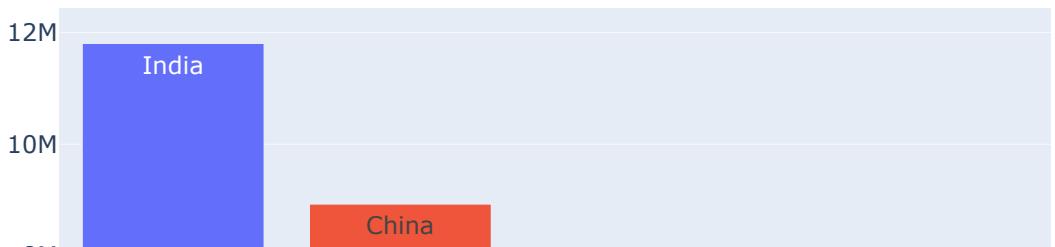
```
Out[132]: <AxesSubplot:xlabel='Year', ylabel='Digestive Diseases'>
```



```
In [133...]: data = df.groupby(['Country/Territory'])["Digestive Diseases"].sum().sort_values(ascending=False)
```

```
In [134...]: px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index, title="Digestive Diseases - No. of People died from Digestive Diseases")
```

Digestive Diseases - No. of People died from Digestive Diseases



31. Fire, Heat, and Hot Substances - No. of People died from Fire or Heat or any Hot Substances

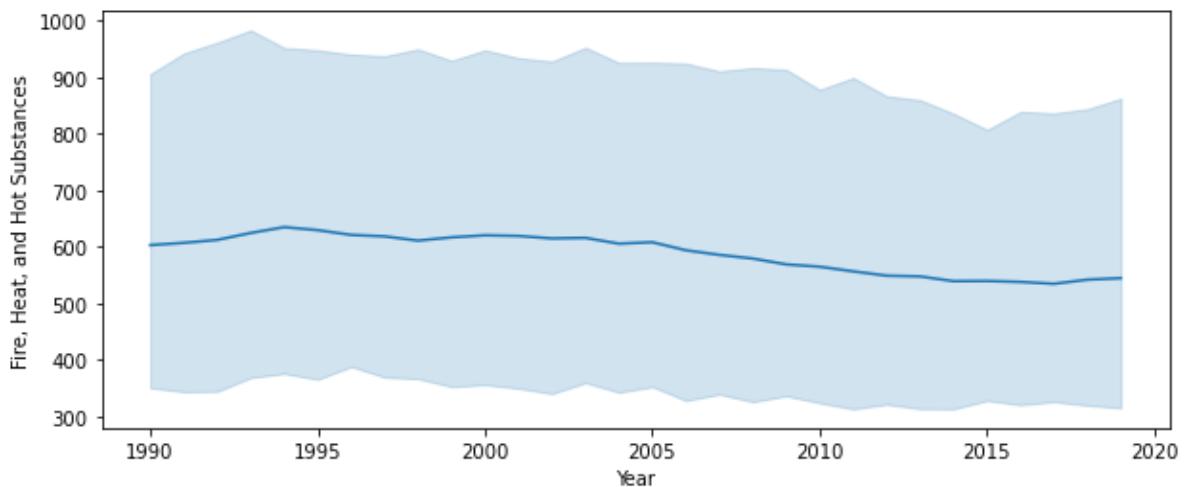
```
In [135...]: df[["Fire, Heat, and Hot Substances"]].describe()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
Out[135]: count    6120.000000
mean     588.711438
std      2128.595120
min      0.000000
25%     17.000000
50%     126.000000
75%     450.000000
max     25876.000000
Name: Fire, Heat, and Hot Substances, dtype: float64
```

```
In [136... plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Fire, Heat, and Hot Substances")
```

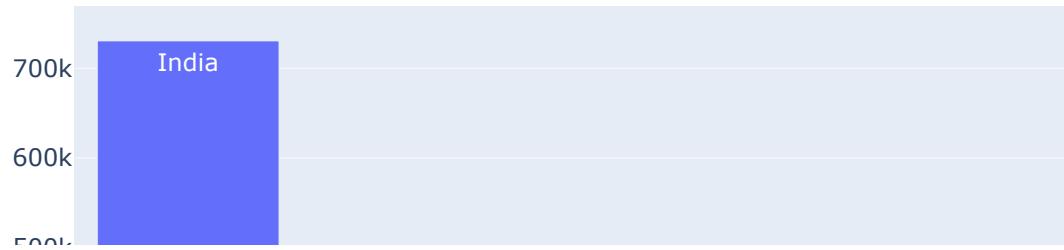
```
Out[136]: <AxesSubplot:xlabel='Year', ylabel='Fire, Heat, and Hot Substances'>
```



```
In [137... data = df.groupby(['Country/Territory'])["Fire, Heat, and Hot Substances"].sum().sort_values(ascending=False)
```

```
In [138... px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index, title="Cause Of Death")]
```

Fire, Heat, and Hot Substances - No. of People died from Fi



32. Acute Hepatitis - No. of People died from Acute Hepatitis

```
In [139]: df["Acute Hepatitis"].describe()
```

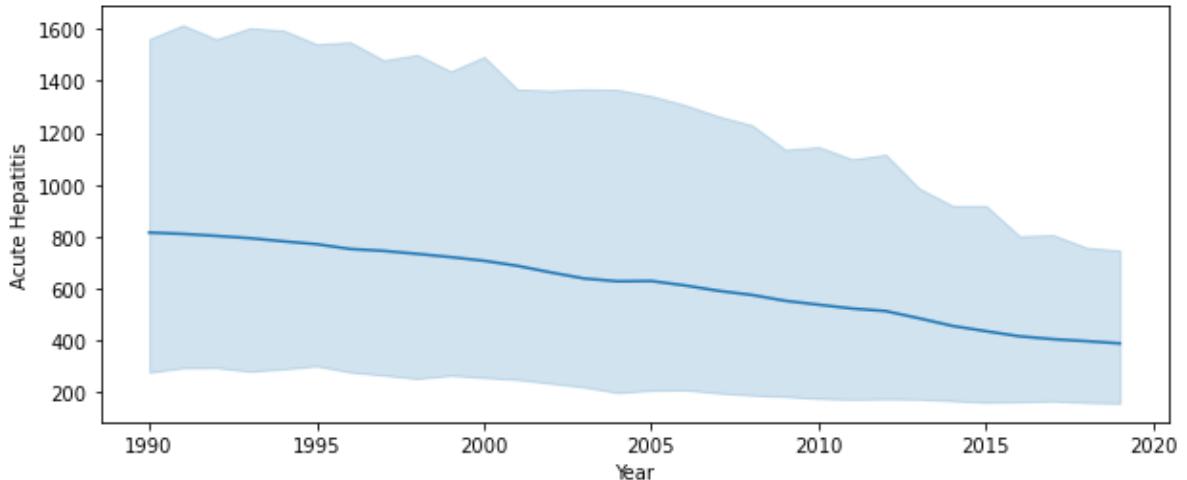
```
Out[139]:
```

count	6120.000000
mean	618.429902
std	4186.023497
min	0.000000
25%	2.000000
50%	15.000000
75%	160.000000
max	64305.000000

Name: Acute Hepatitis, dtype: float64

```
In [140]: plt.figure(figsize=(10,4))
sns.lineplot(data=df, x="Year", y="Acute Hepatitis")
```

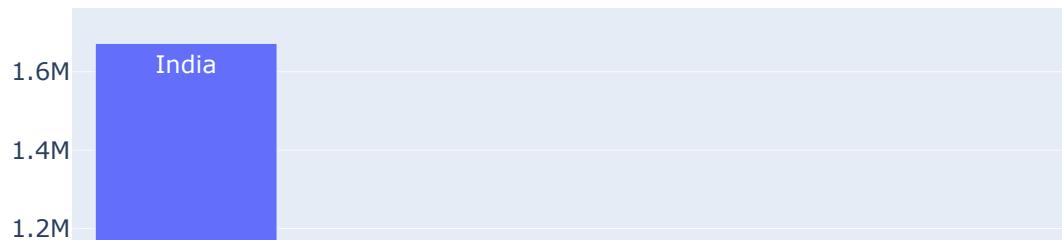
```
Out[140]: <AxesSubplot:xlabel='Year', ylabel='Acute Hepatitis'>
```



```
In [141...]: data = df.groupby(['Country/Territory'])["Acute Hepatitis"].sum().sort_values(ascending=True)
```

```
In [142...]: px.bar(data,x = data.index , y = data.values, text=data.index,color = data.index, title="Acute Hepatitis - No. of People died from Acute Hepatitis")
```

Acute Hepatitis - No. of People died from Acute Hepatitis



Severity Of disease in the countries

```
In [143...]: df1 = df.drop('Year',axis=1).groupby('Code').sum().reset_index()
```

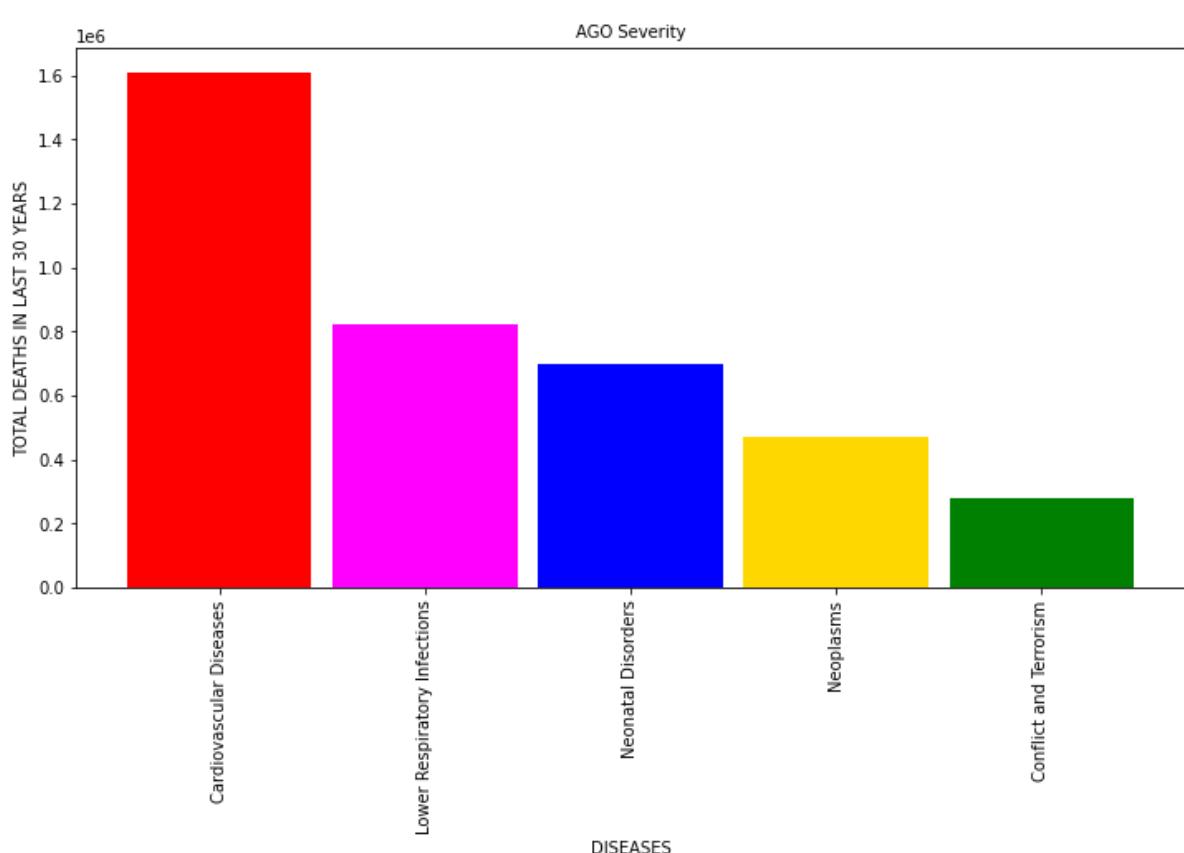
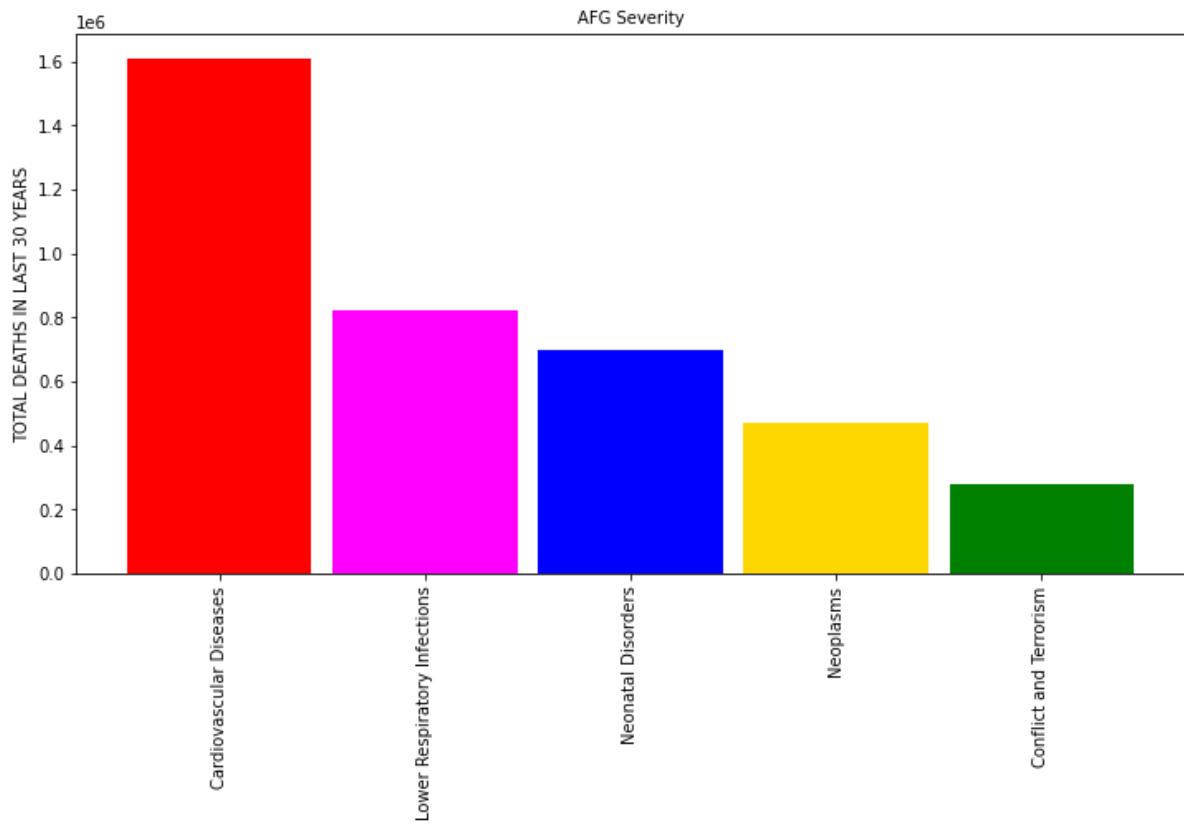
```
In [144...]: for x in df1.index:
    v=df1.Code.iloc[x]
```

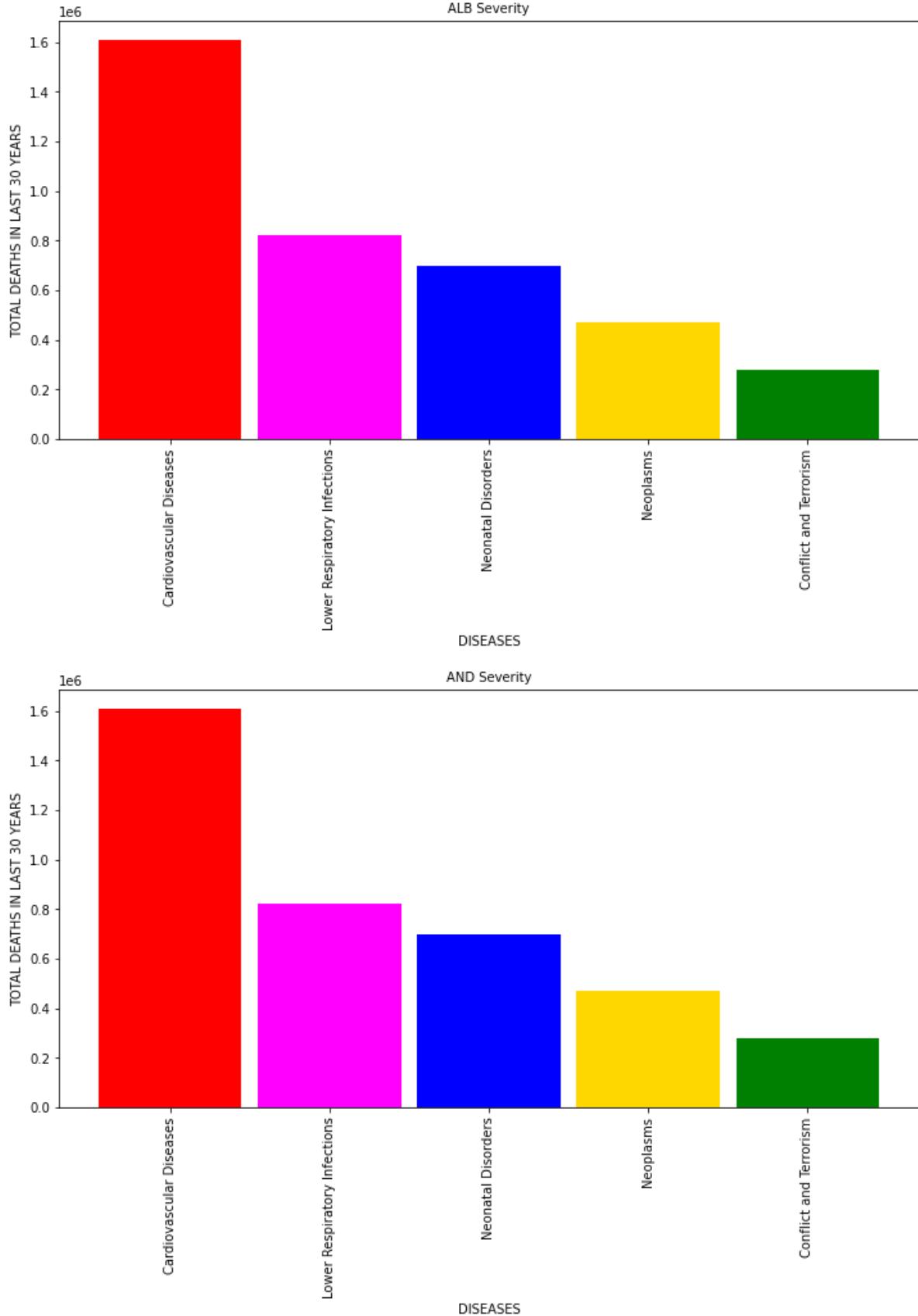
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js 0].nlargest(5)

```

plt.figure(figsize=(12,6))
plt.bar(data=temp ,x = temp.index , height = temp.values, width=0.9, color = [
    plt.xticks(rotation='vertical')
    plt.xlabel("DISEASES" , size = 10)
    plt.ylabel('TOTAL DEATHS IN LAST 30 YEARS',size = 10)
    plt.title(y.upper() +' Severity',size =10)

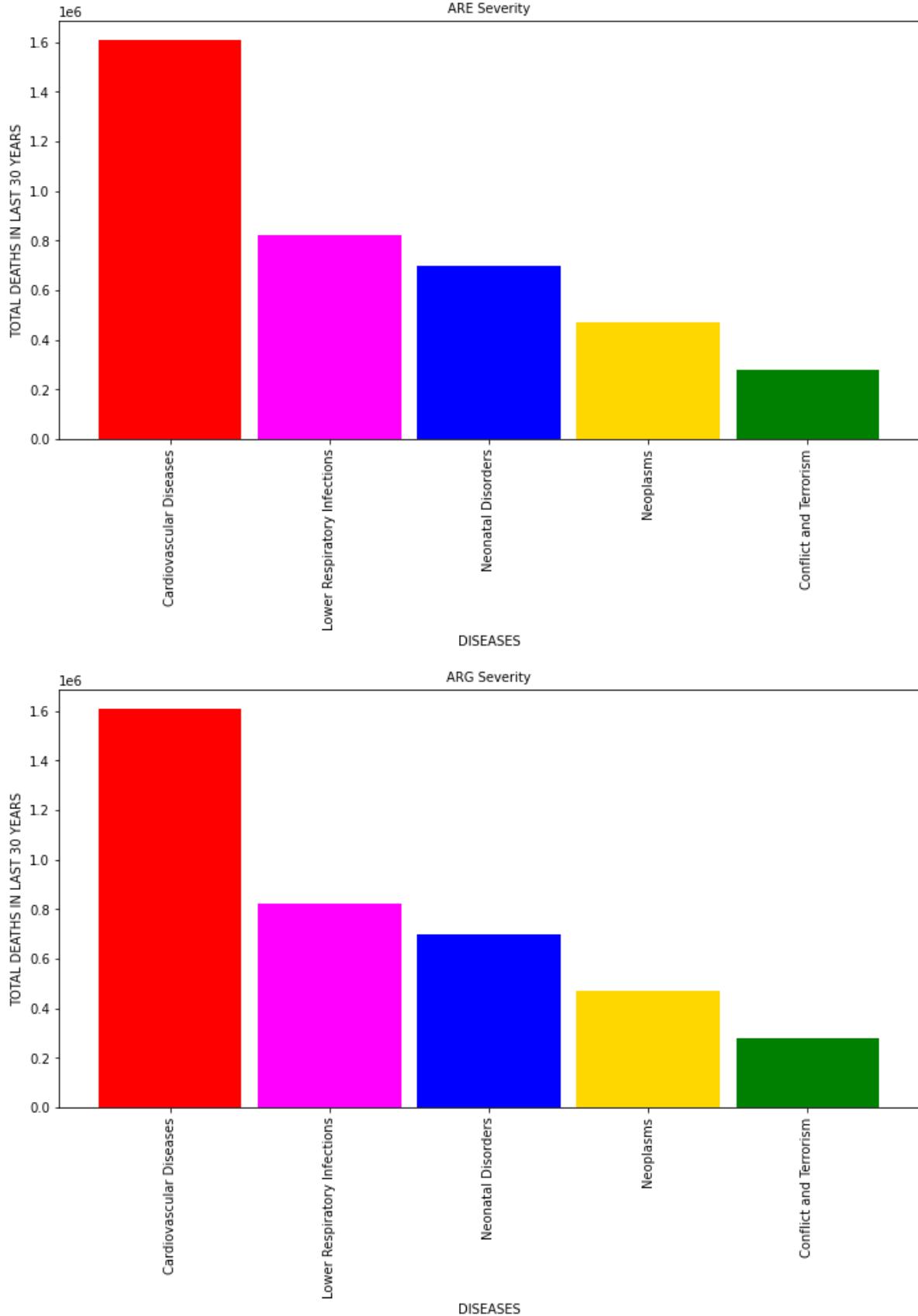
```



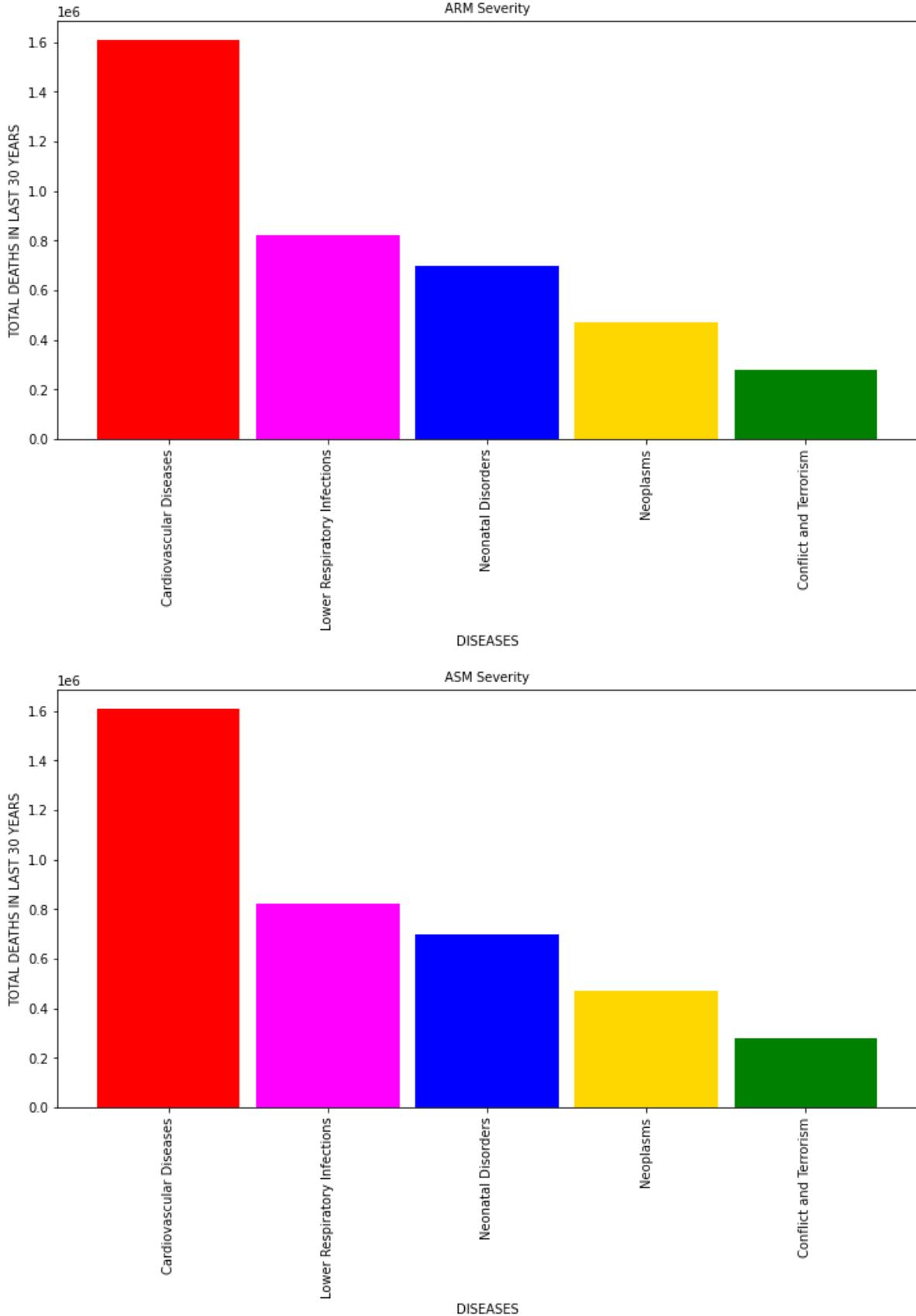


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death

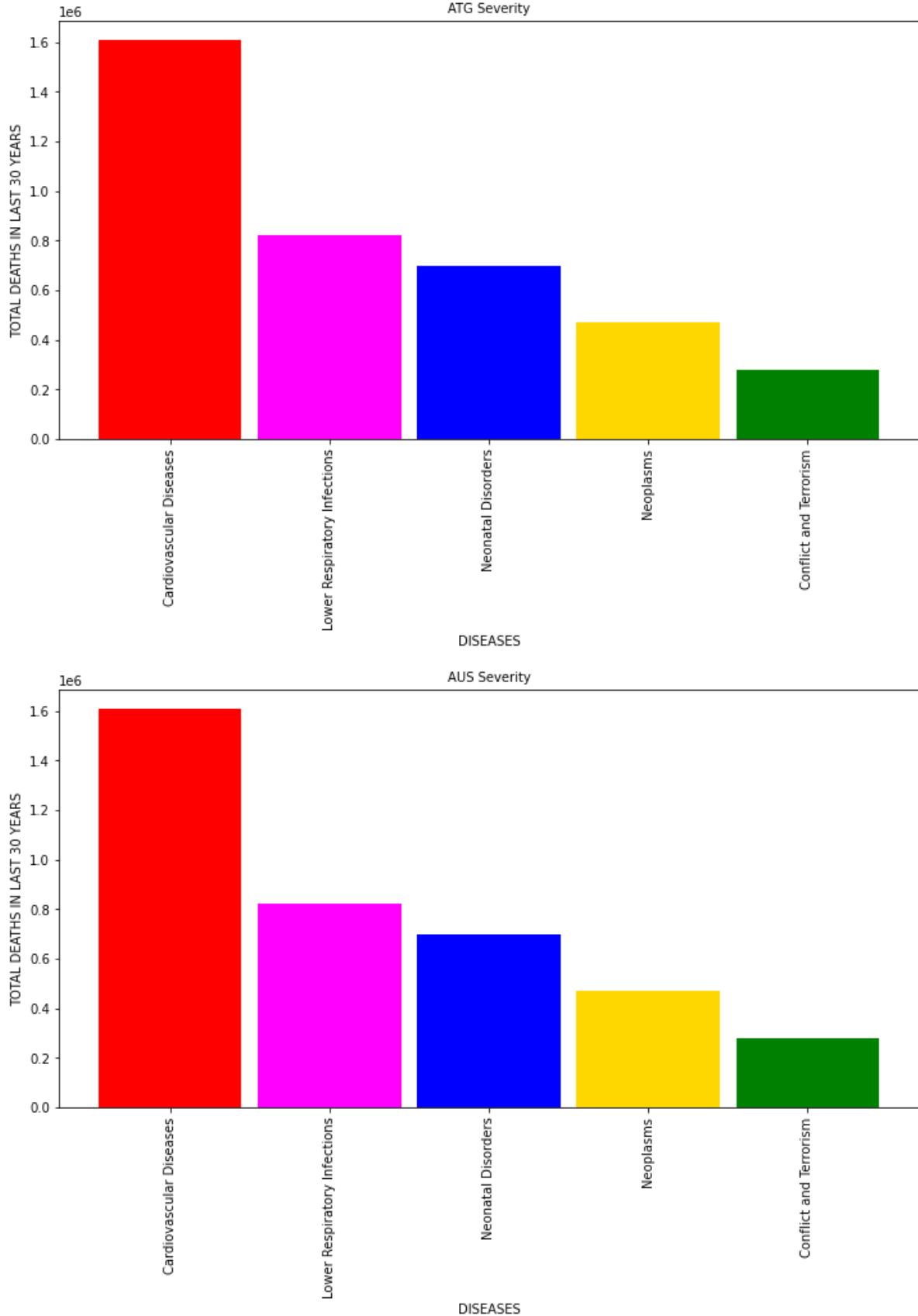


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



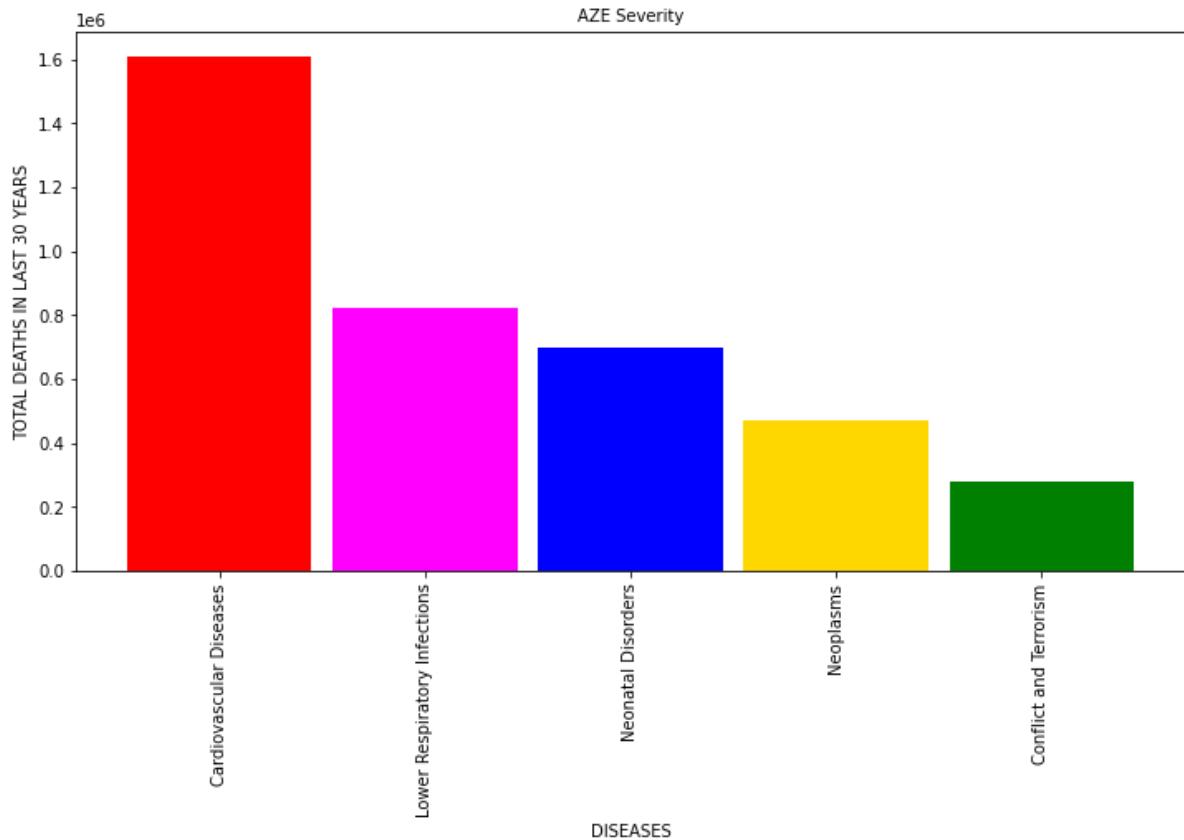
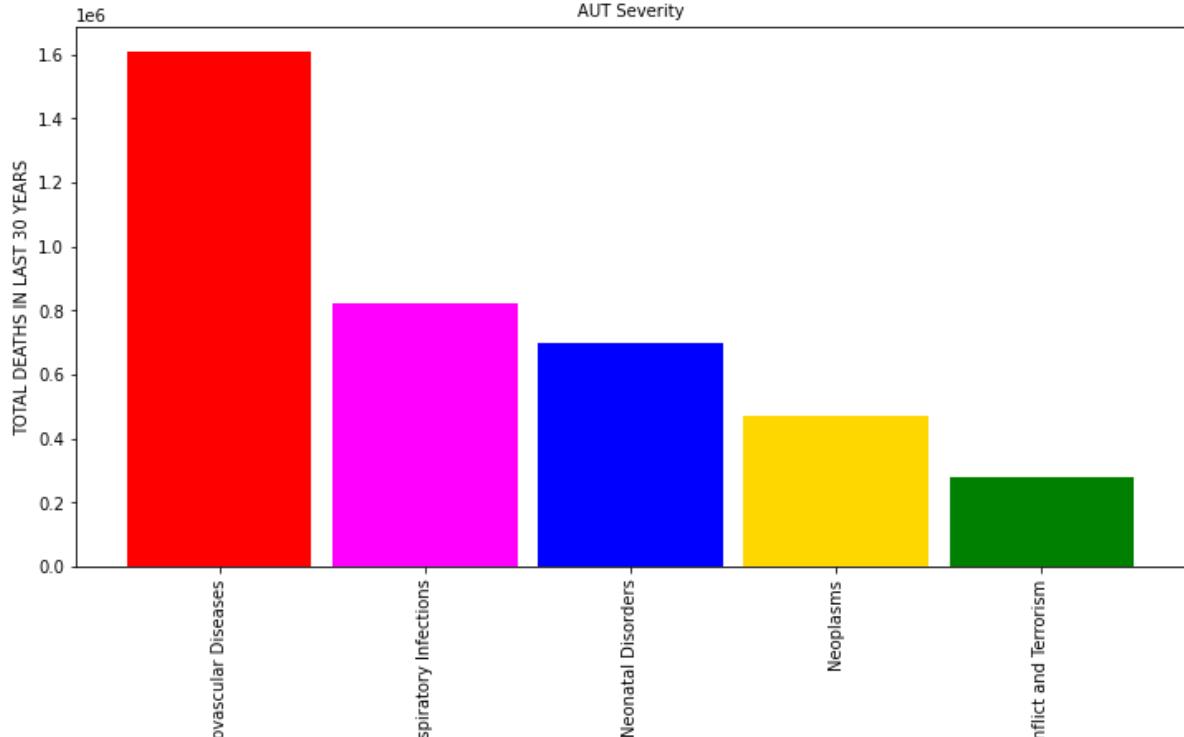
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



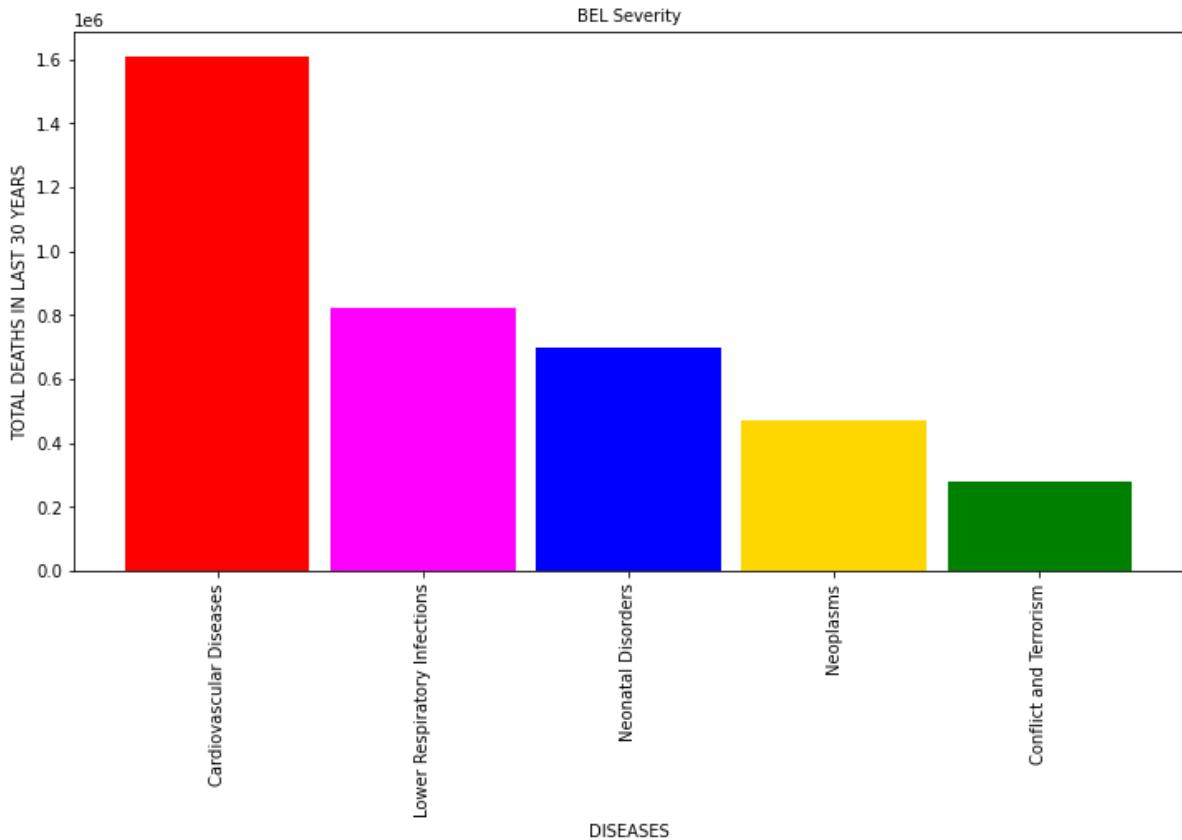
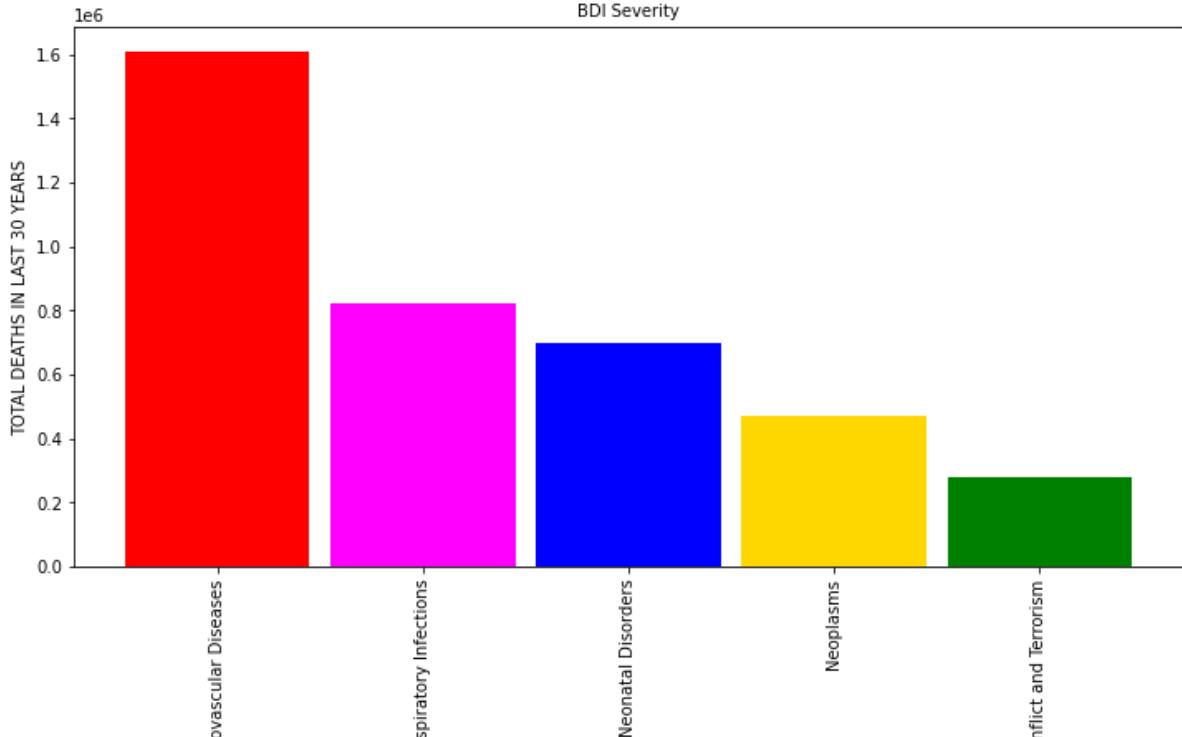
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



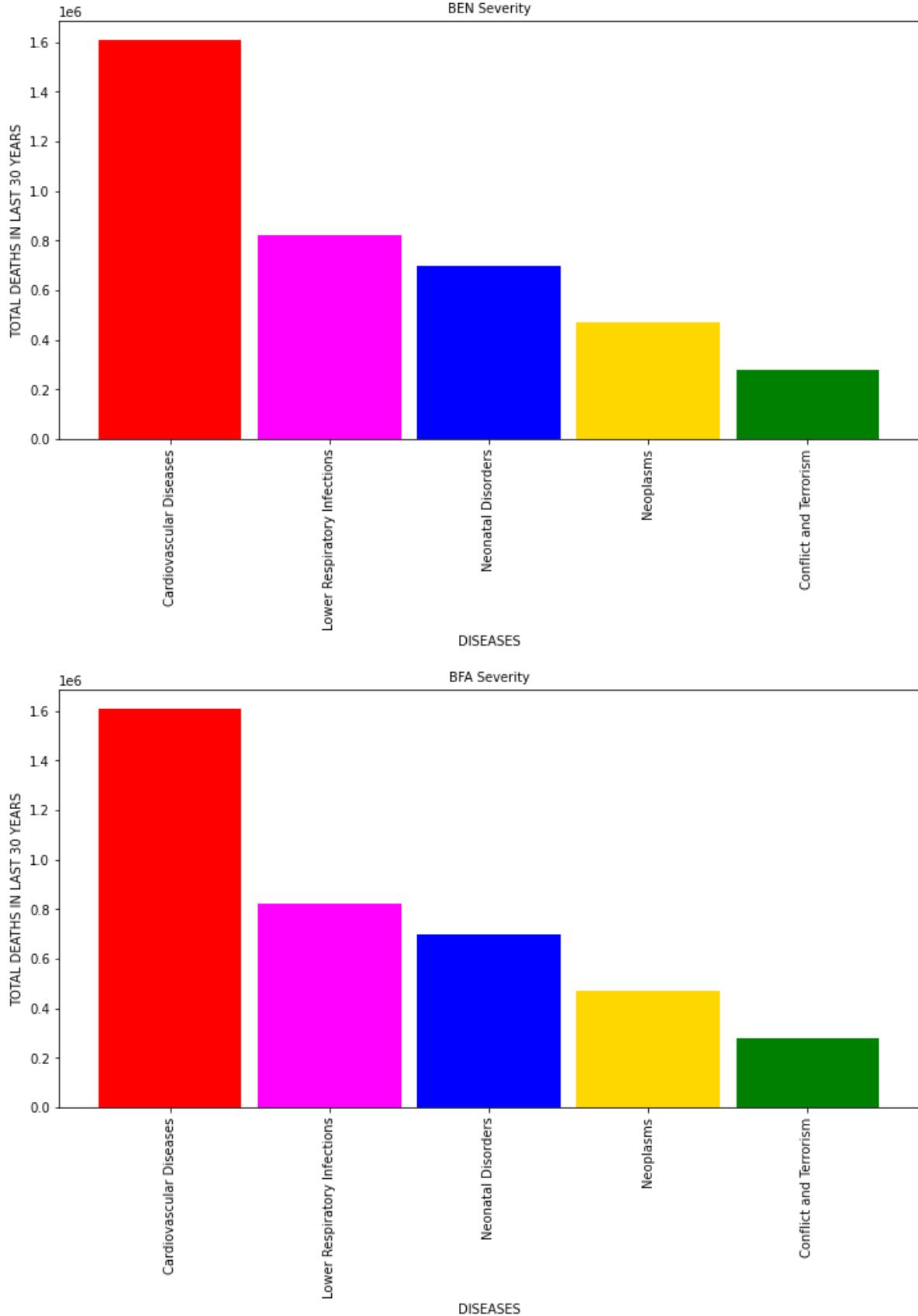
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



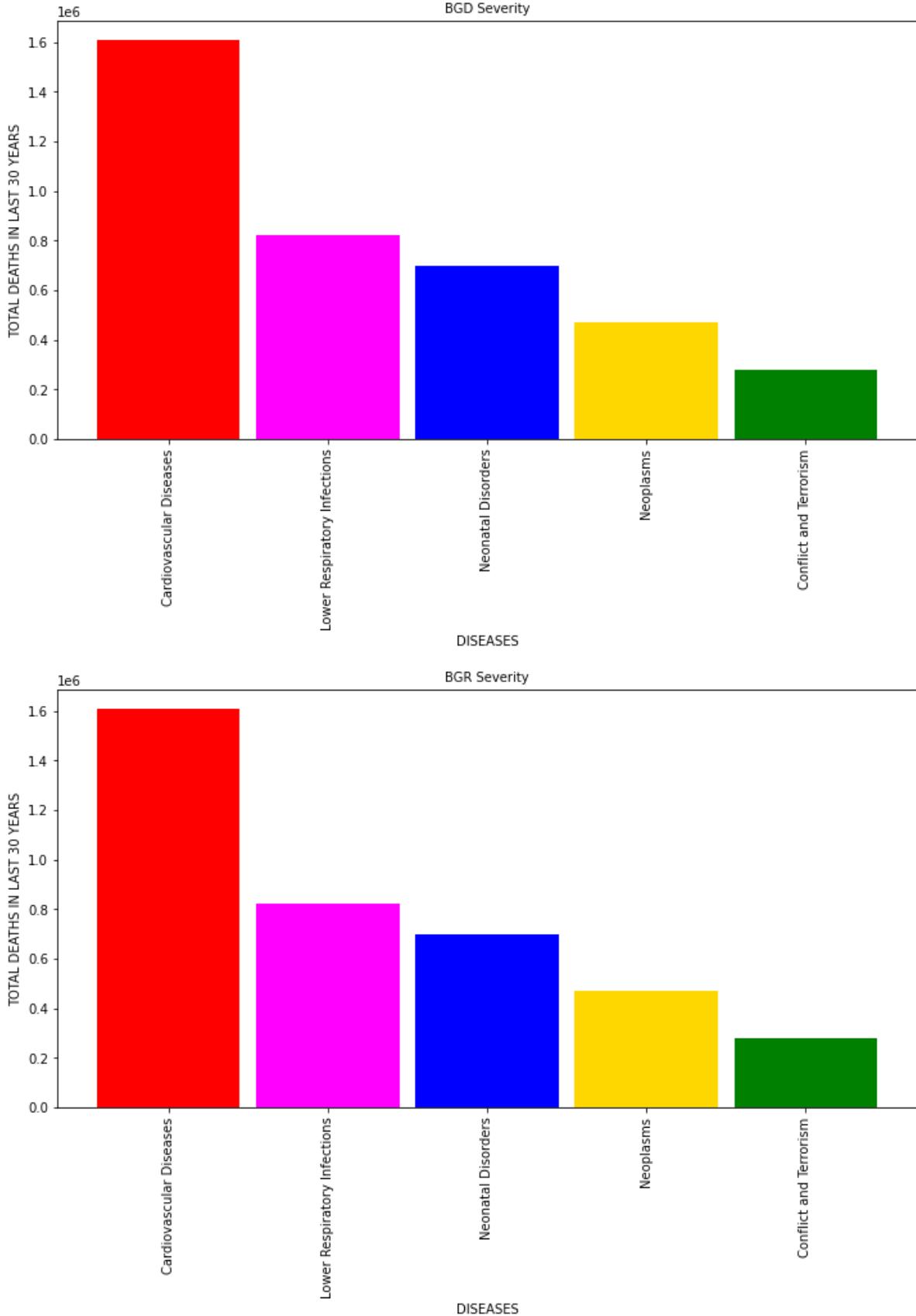
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



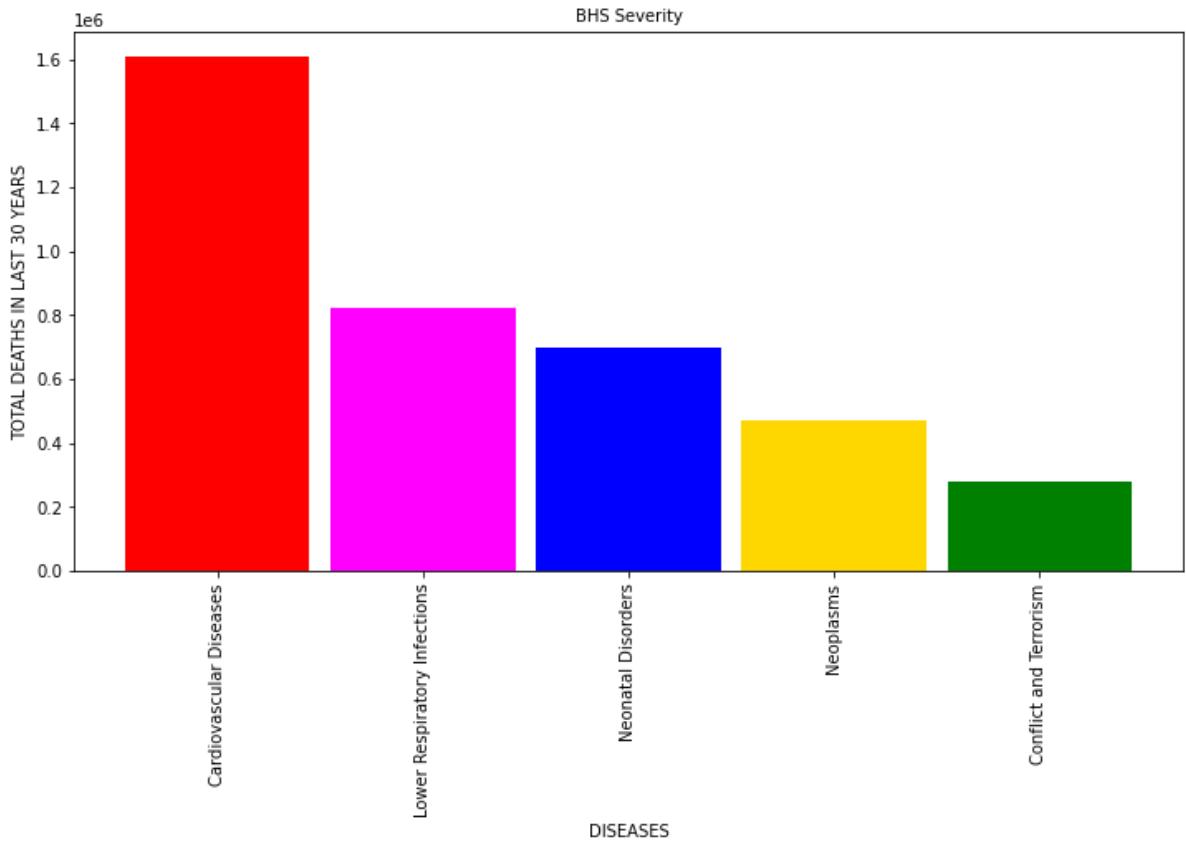
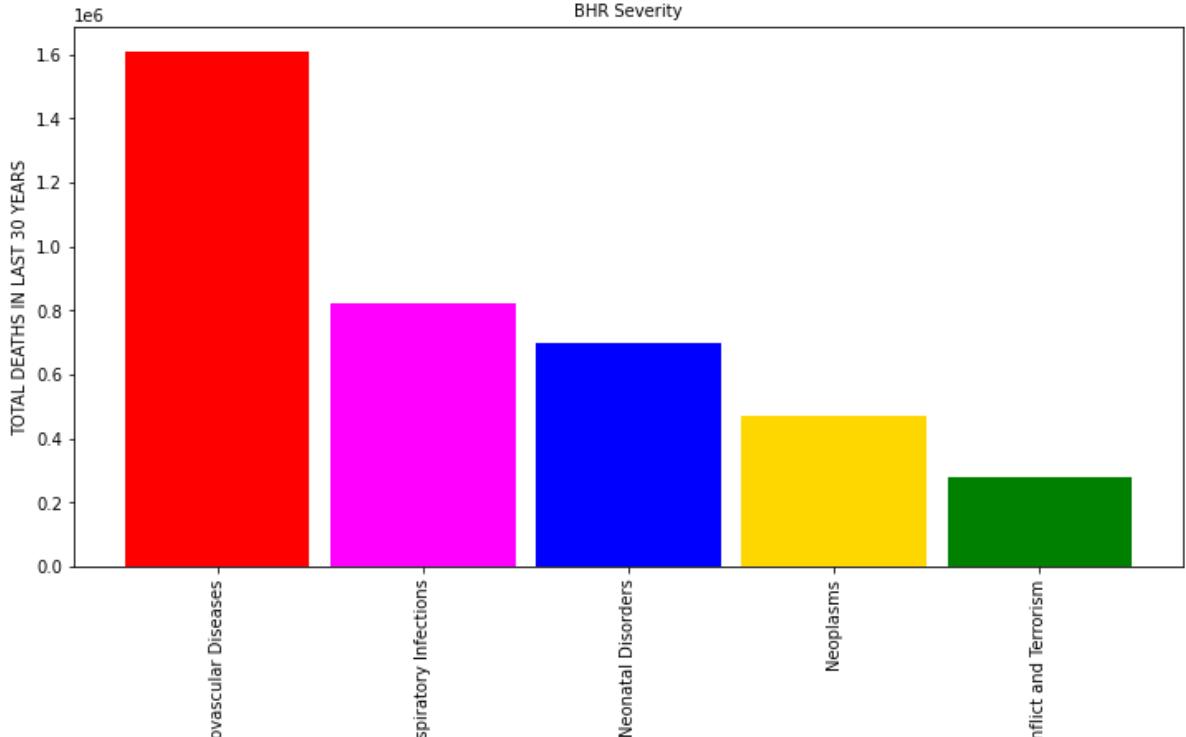
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



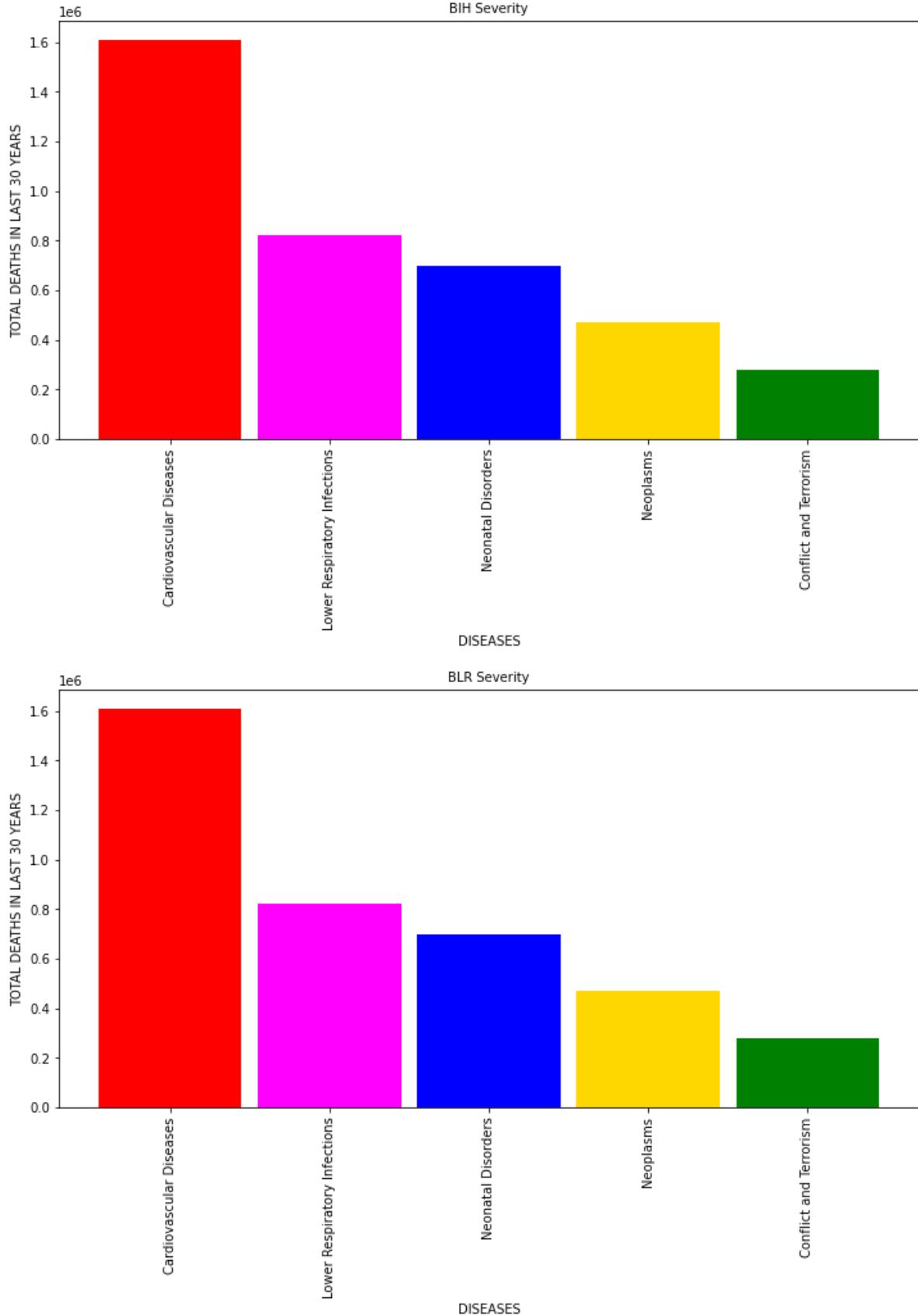
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



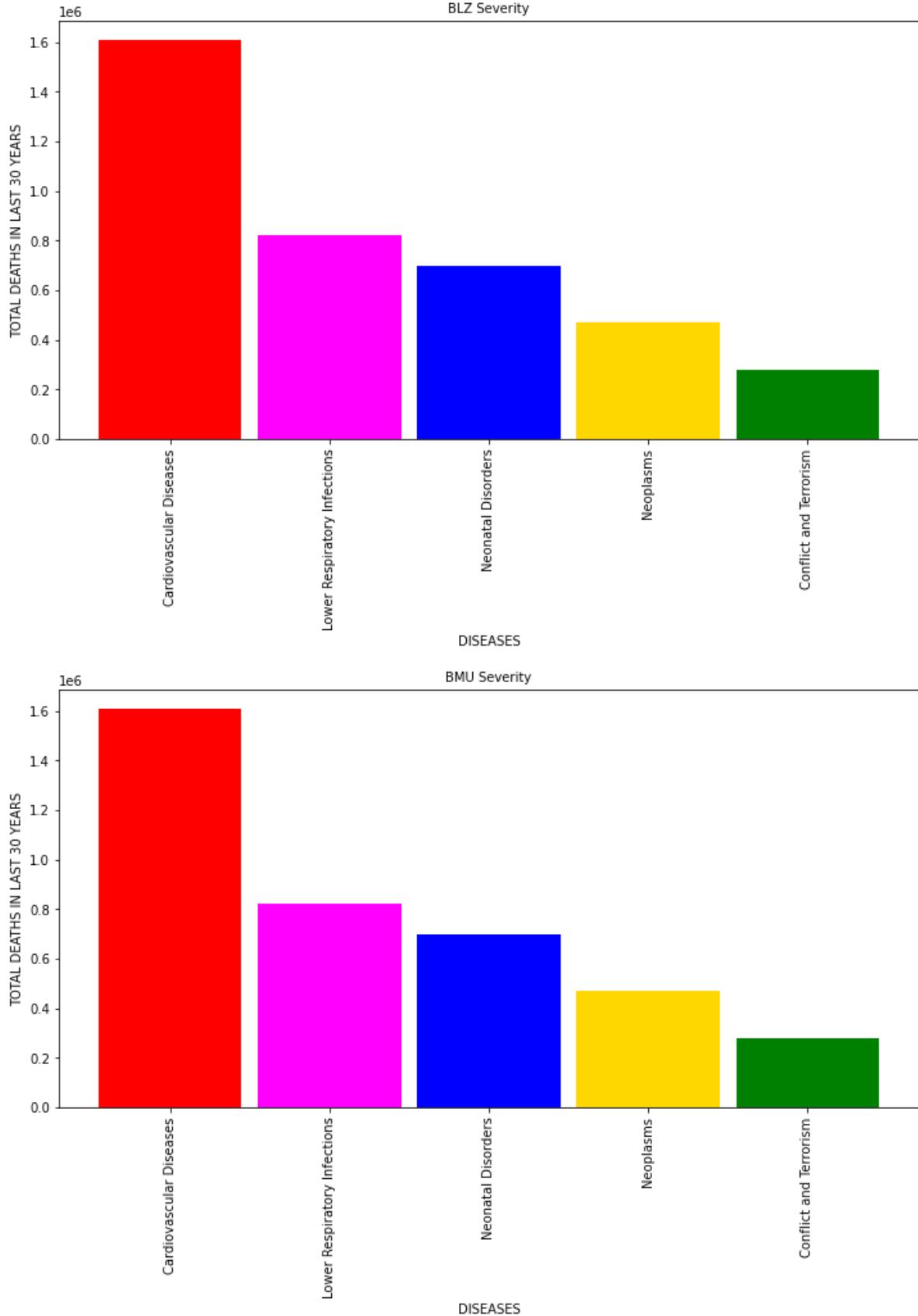
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



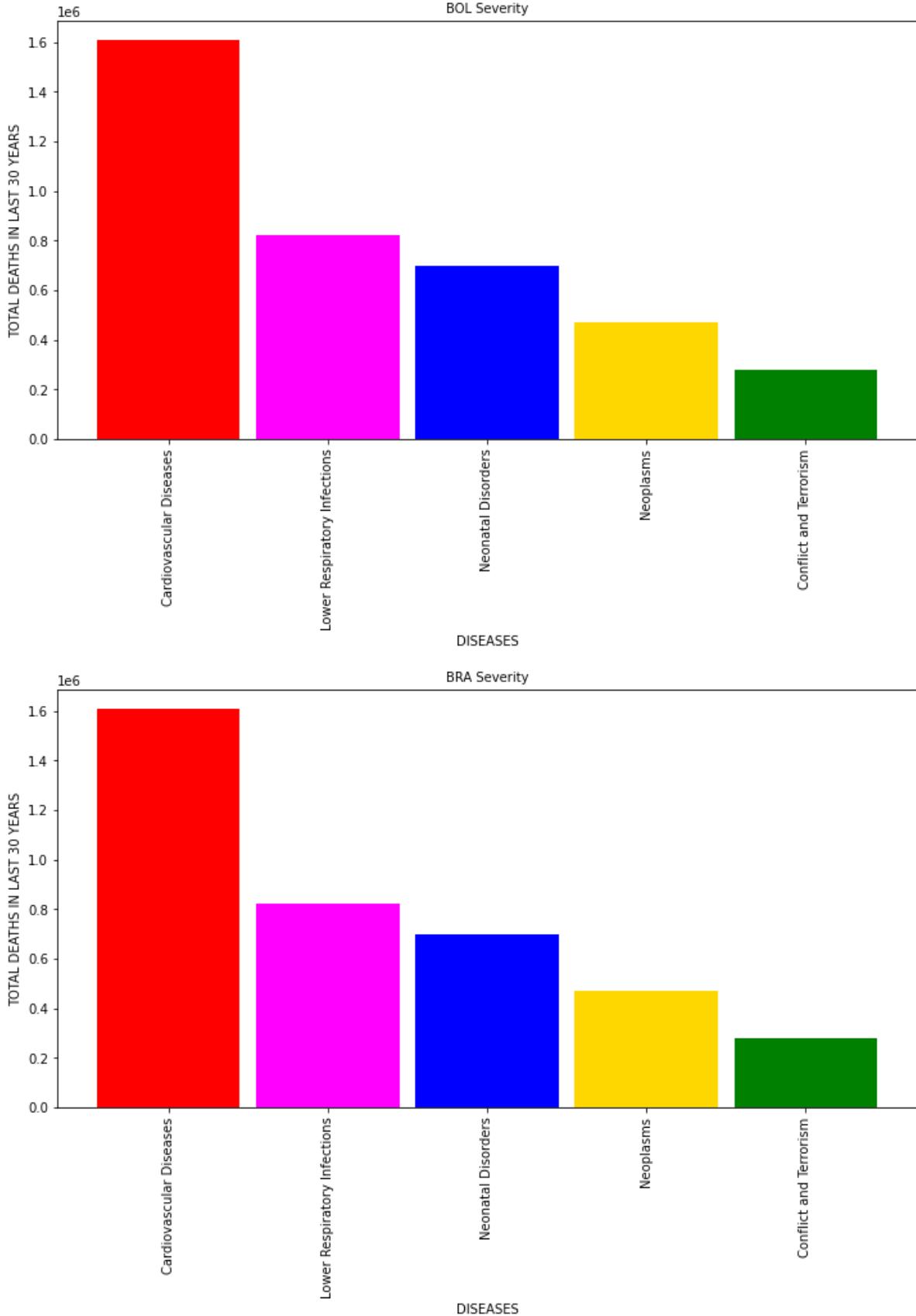
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



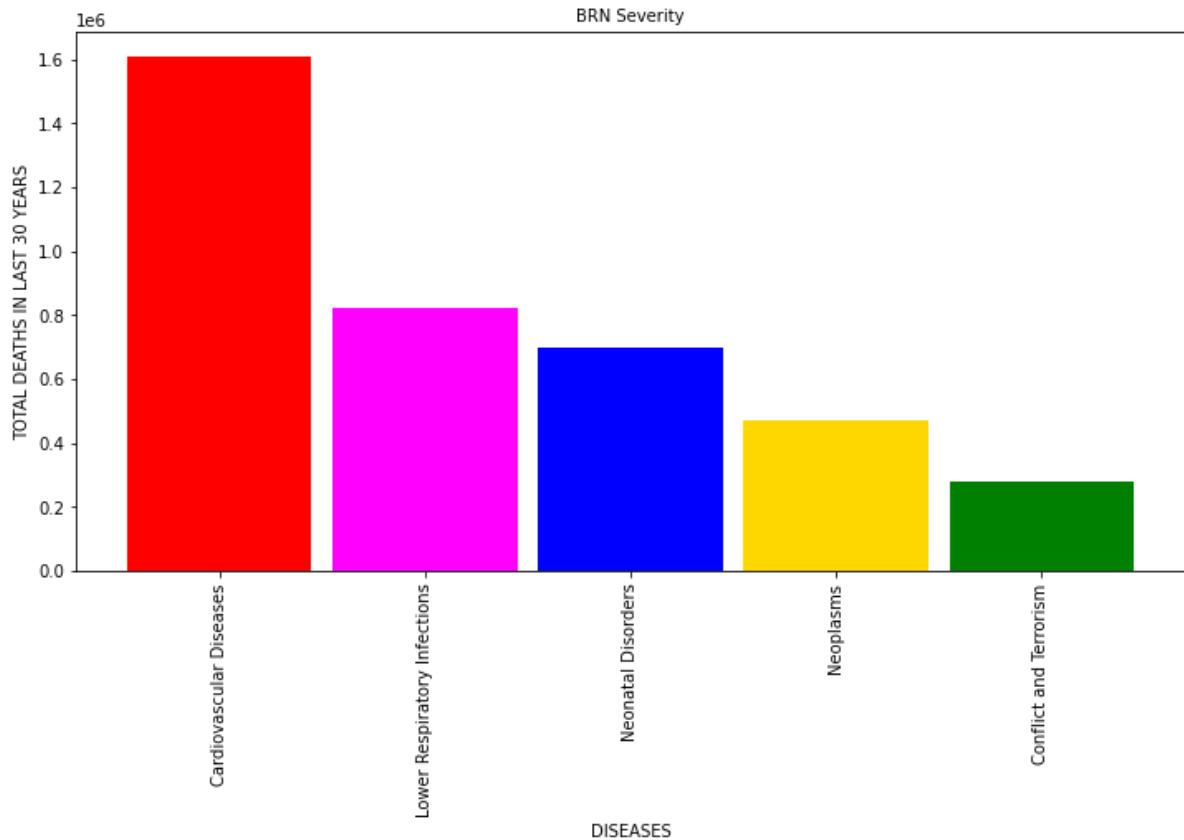
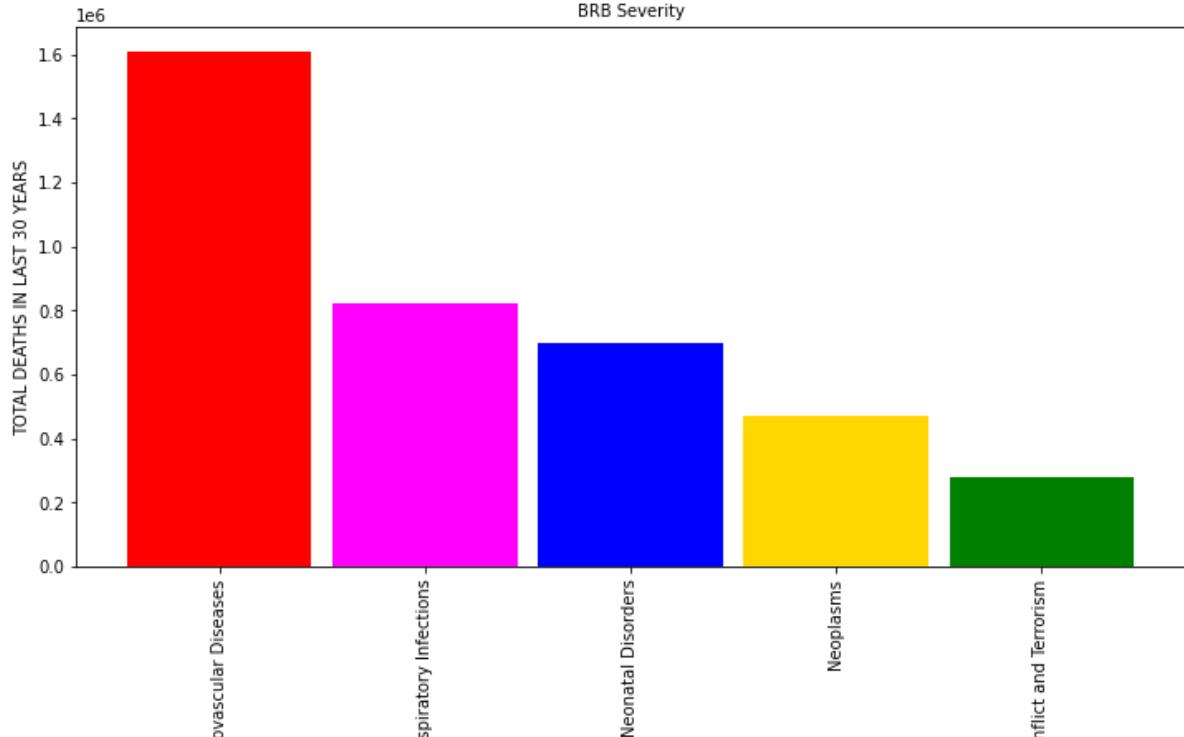
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



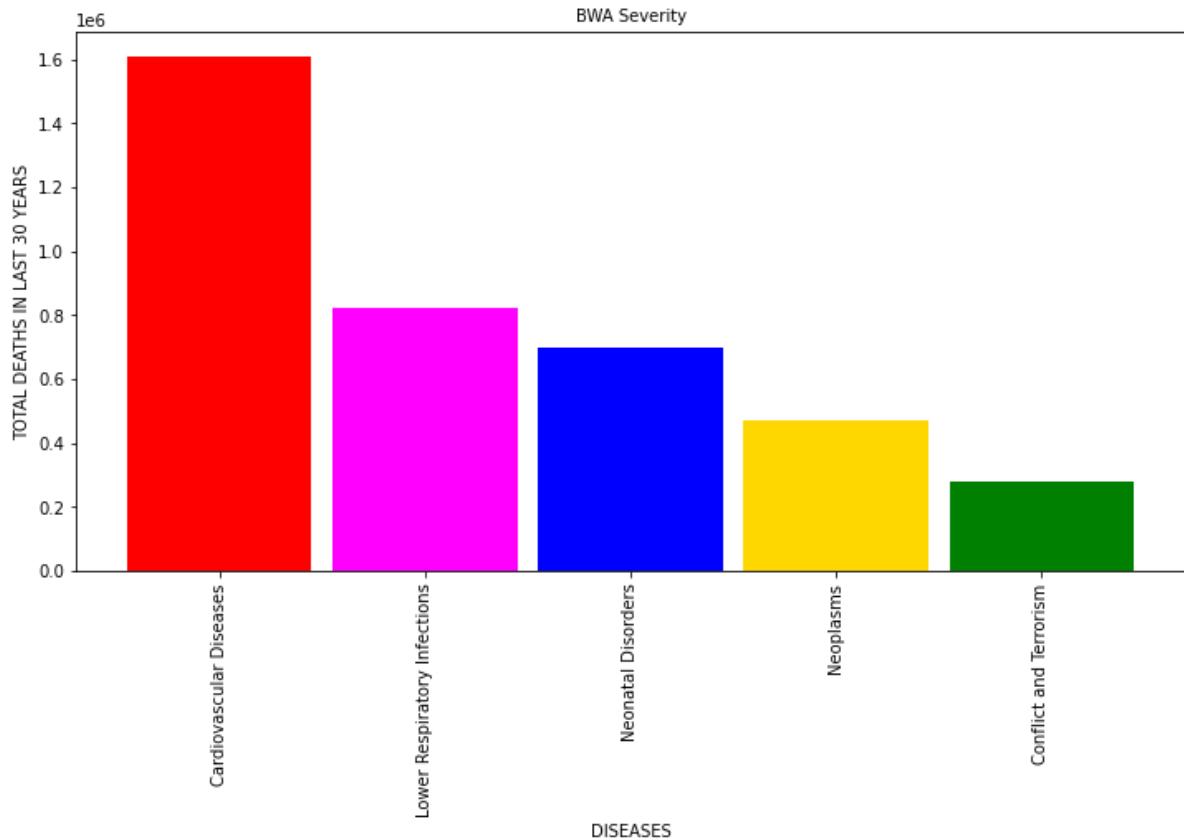
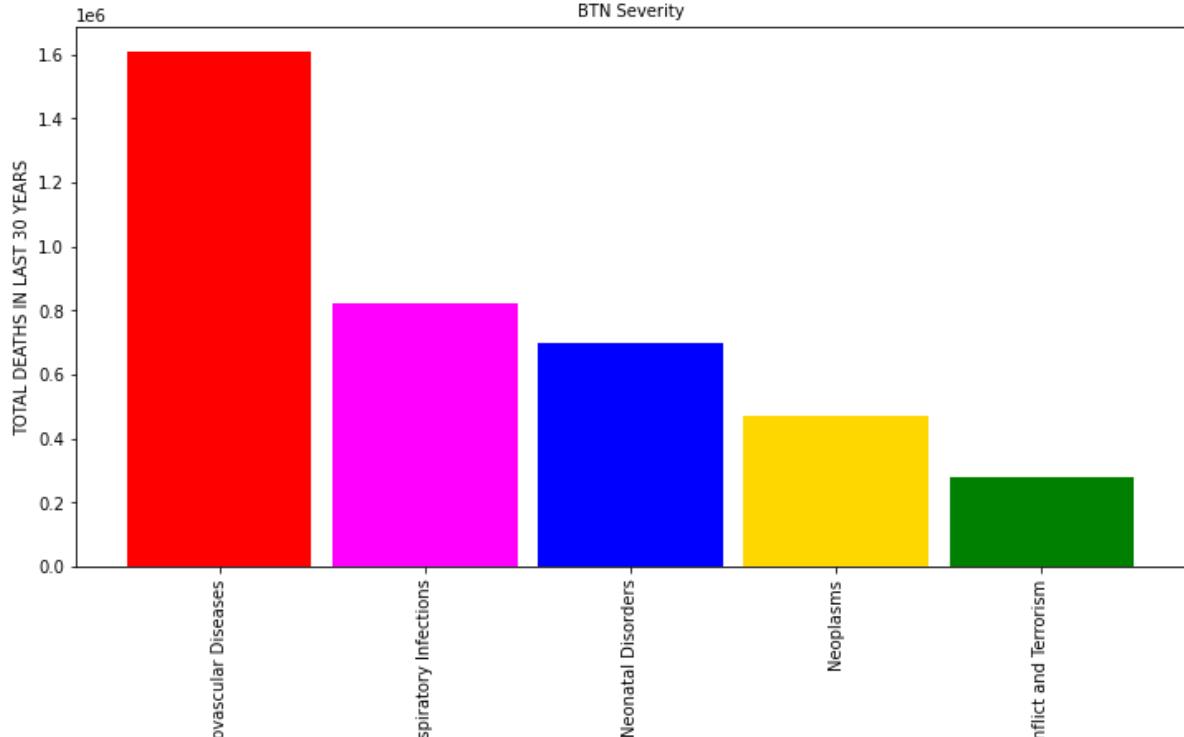
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



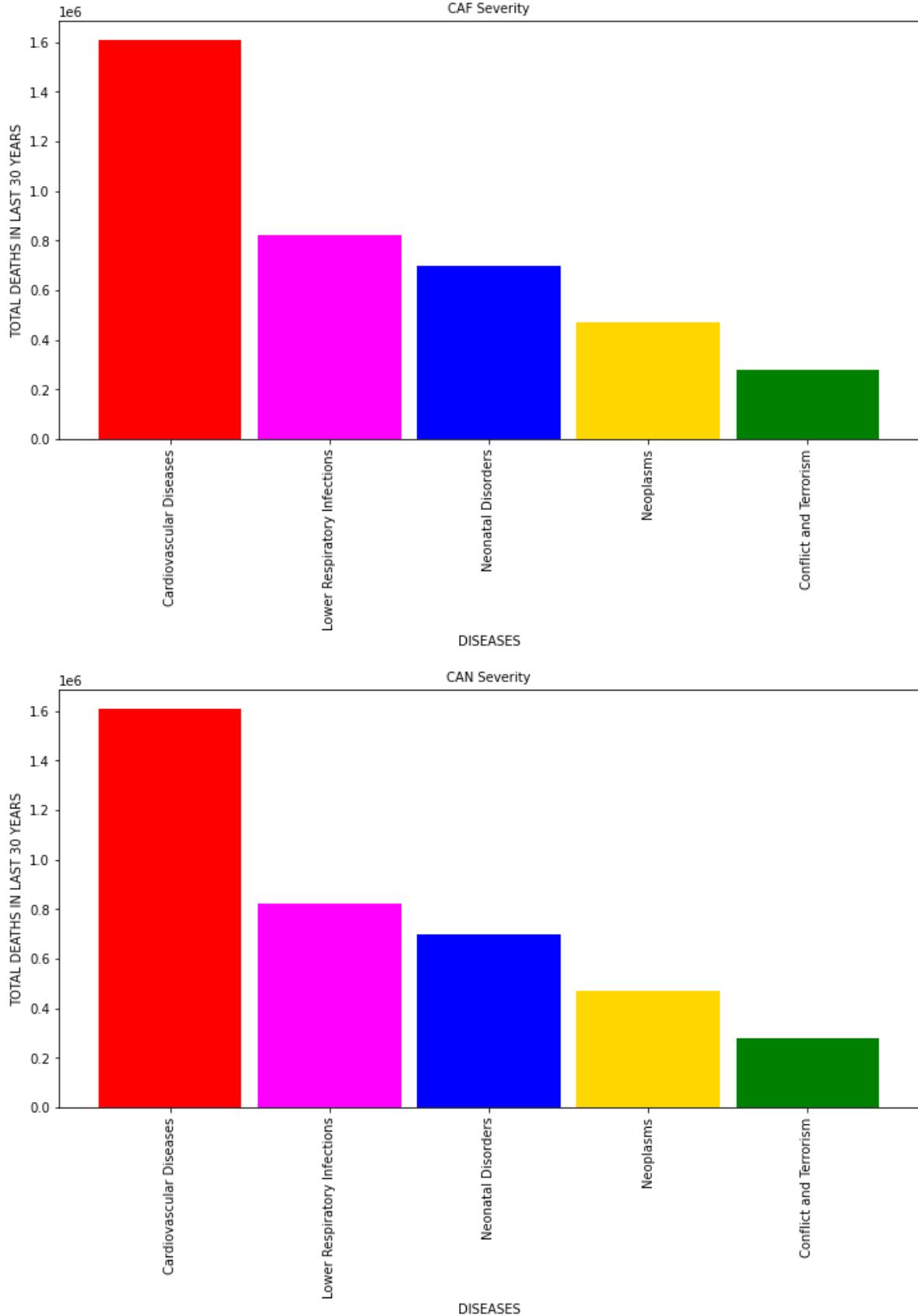
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



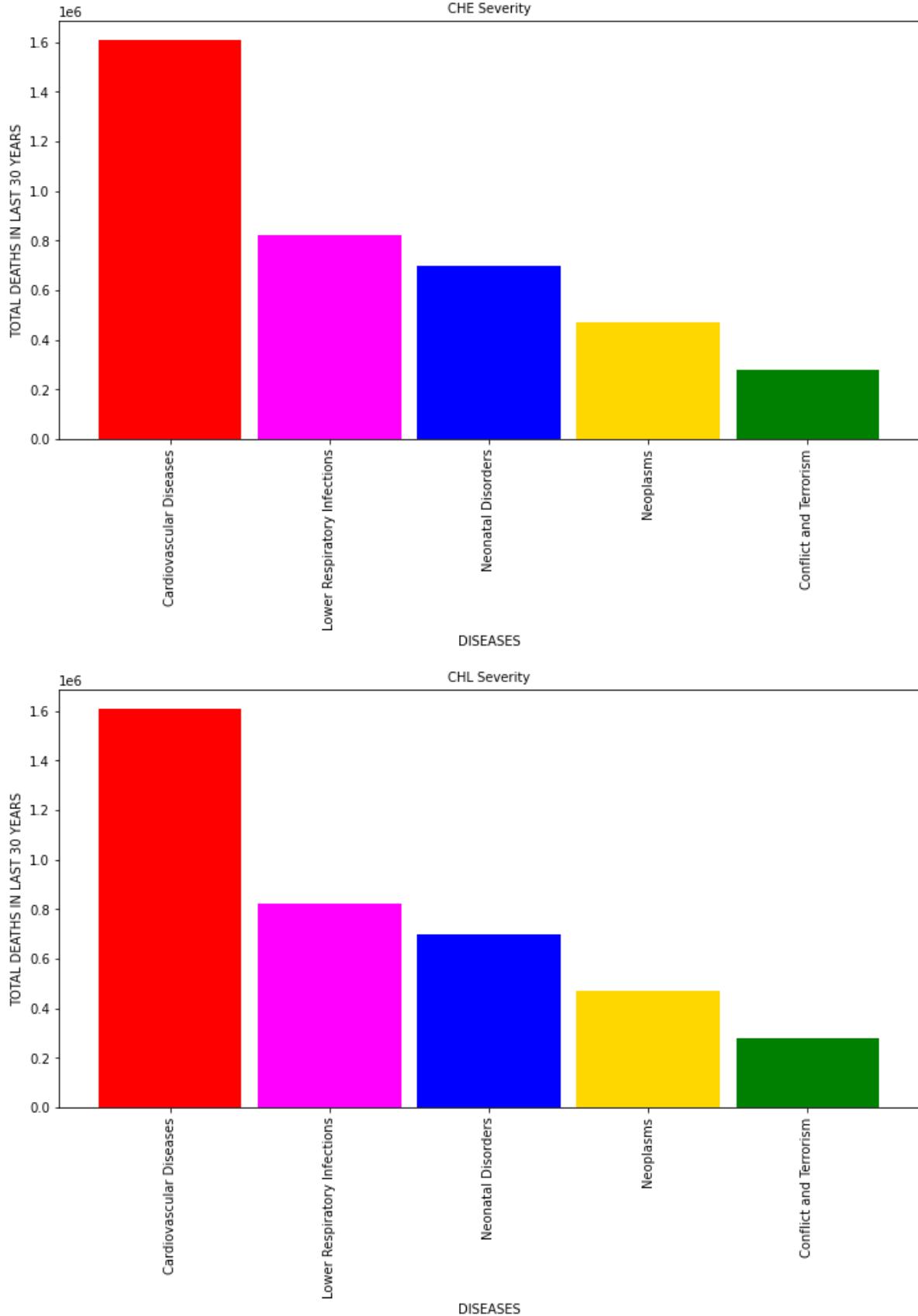
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death

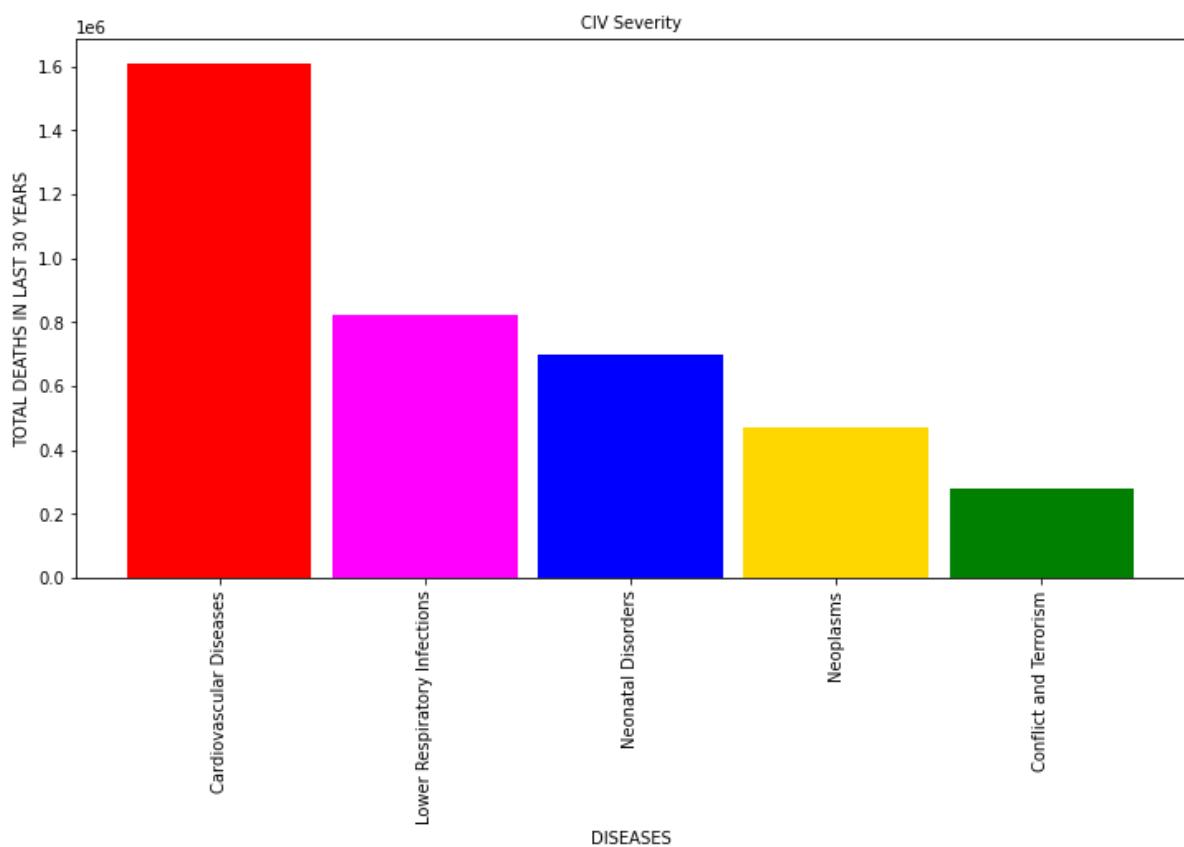
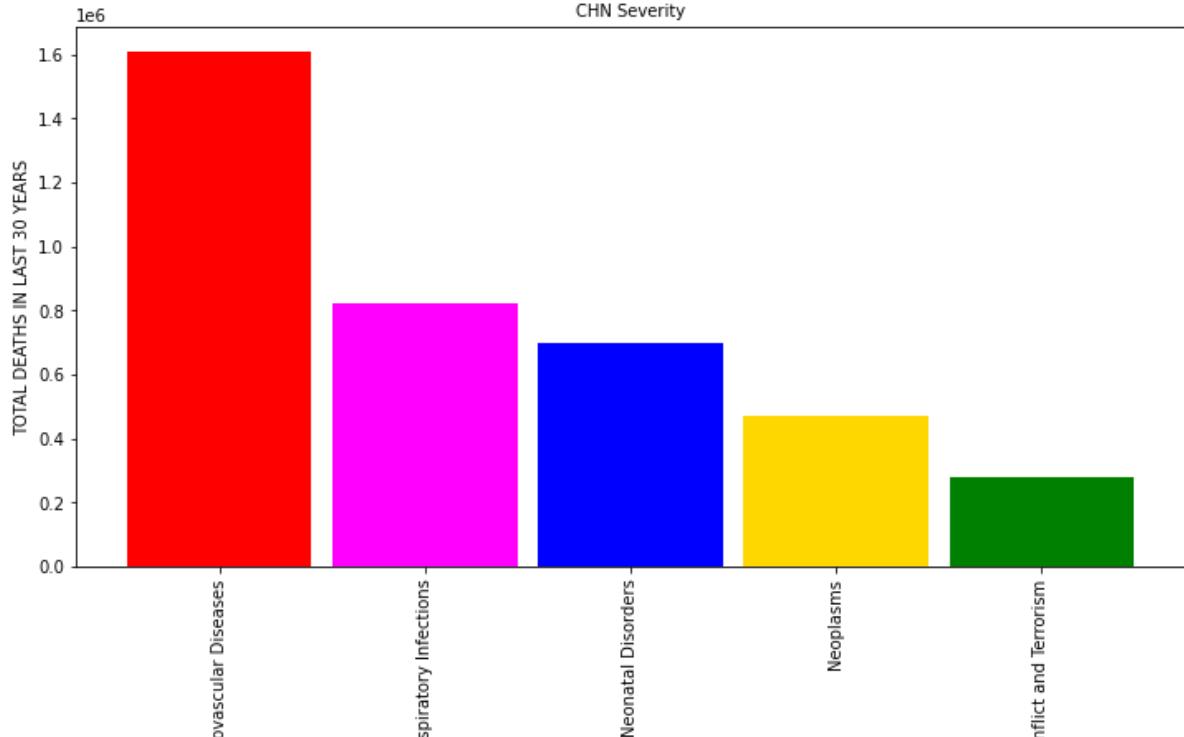


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

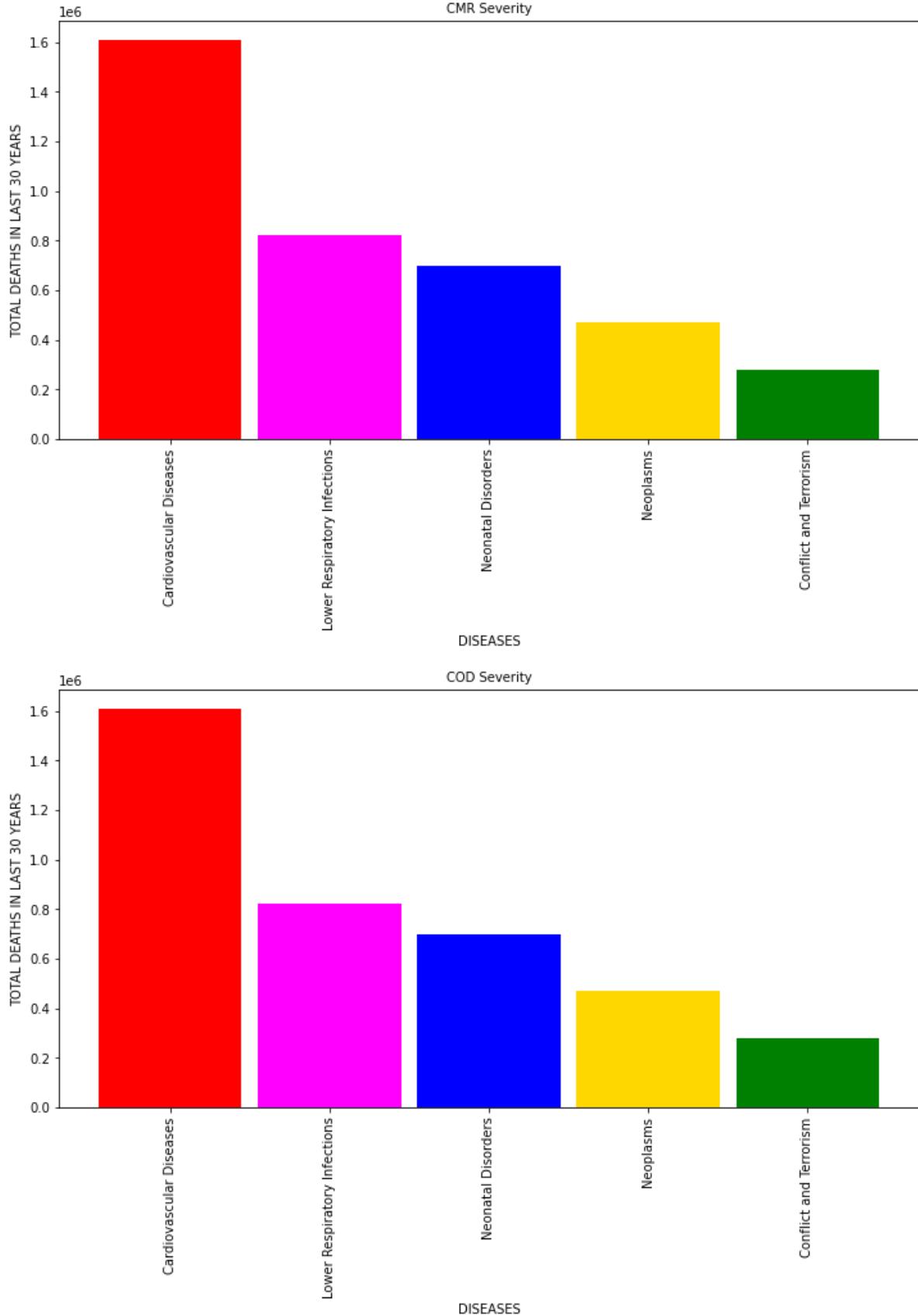
Cause Of Death



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

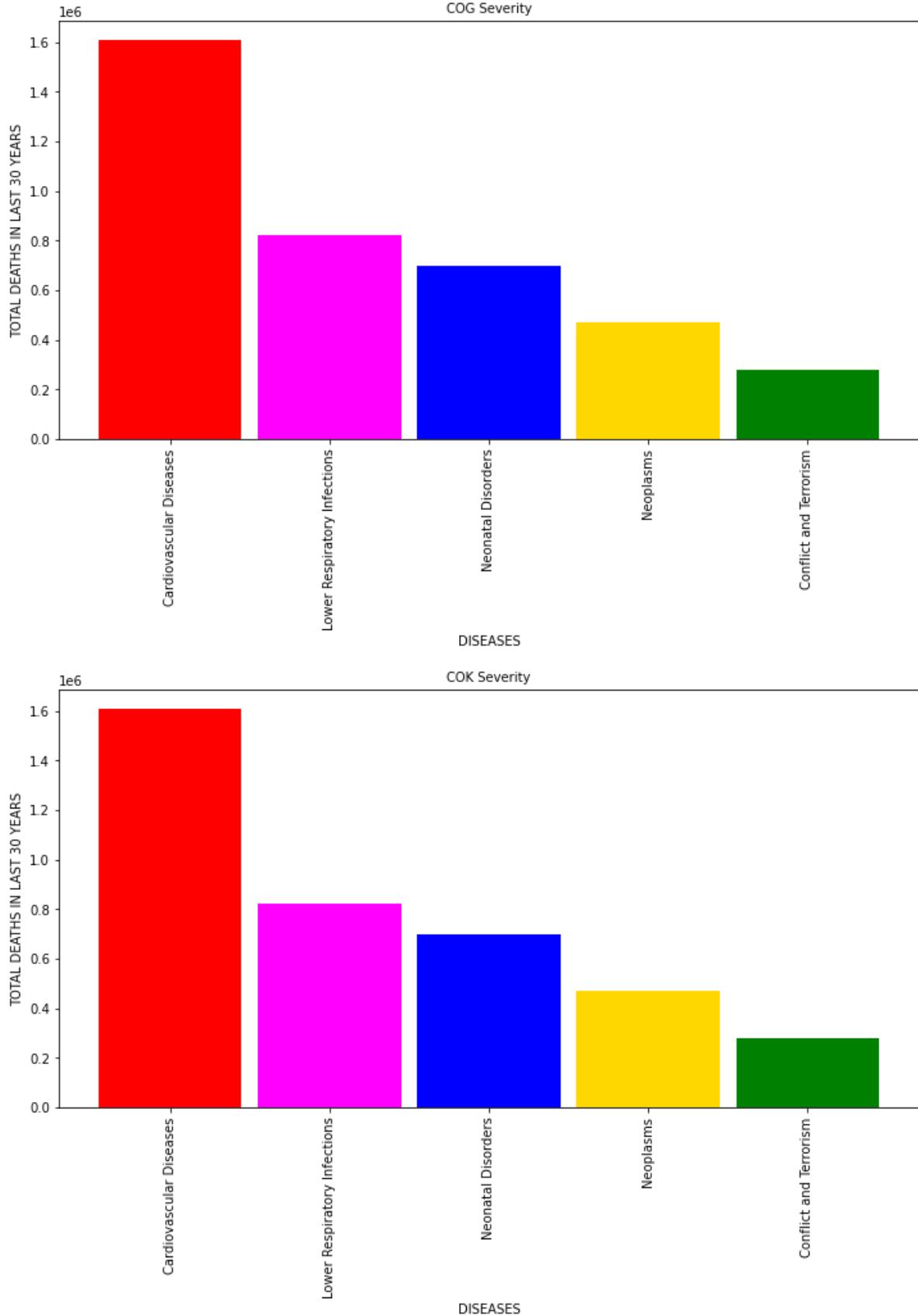


Cause Of Death



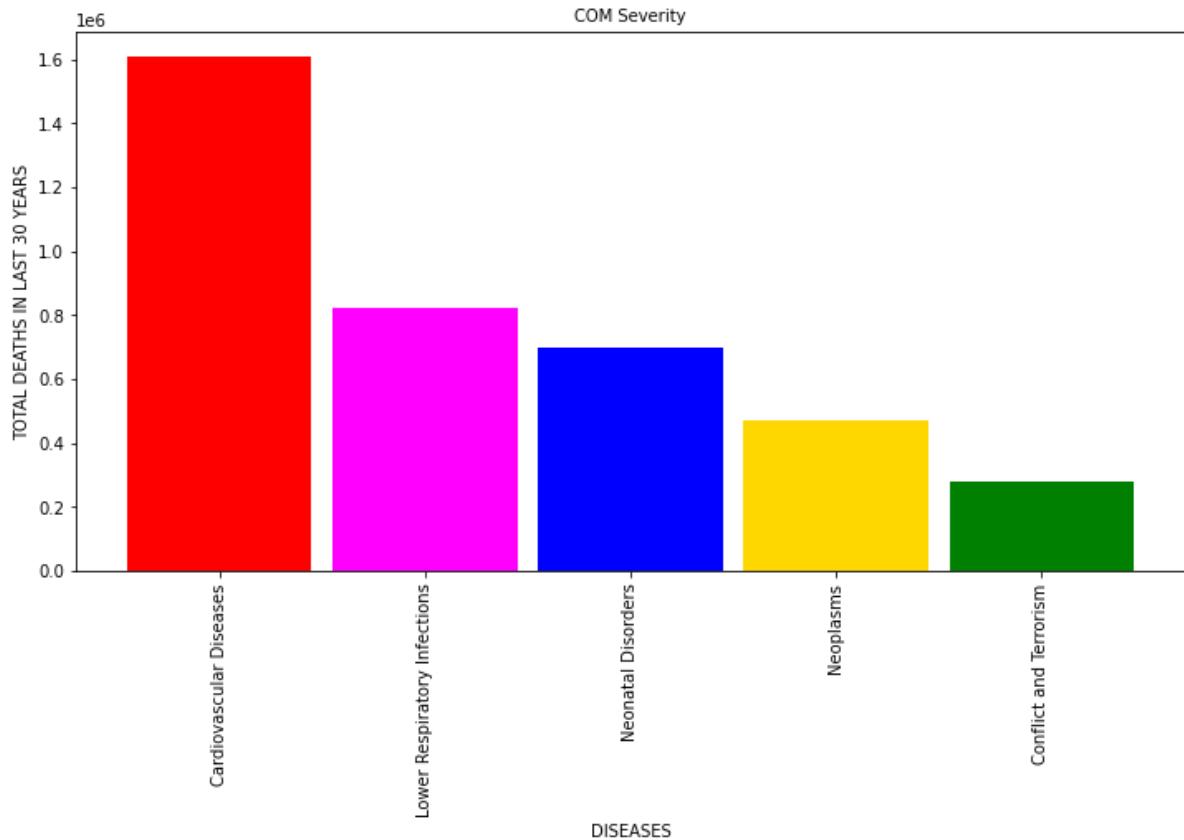
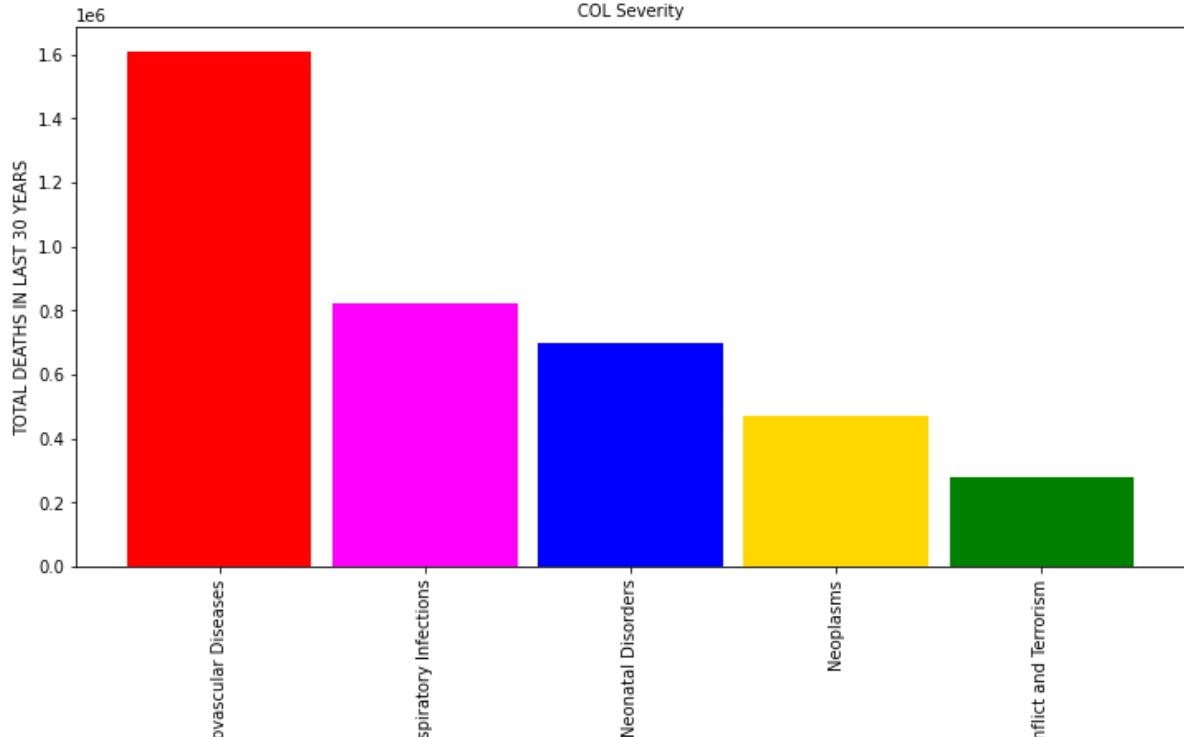
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



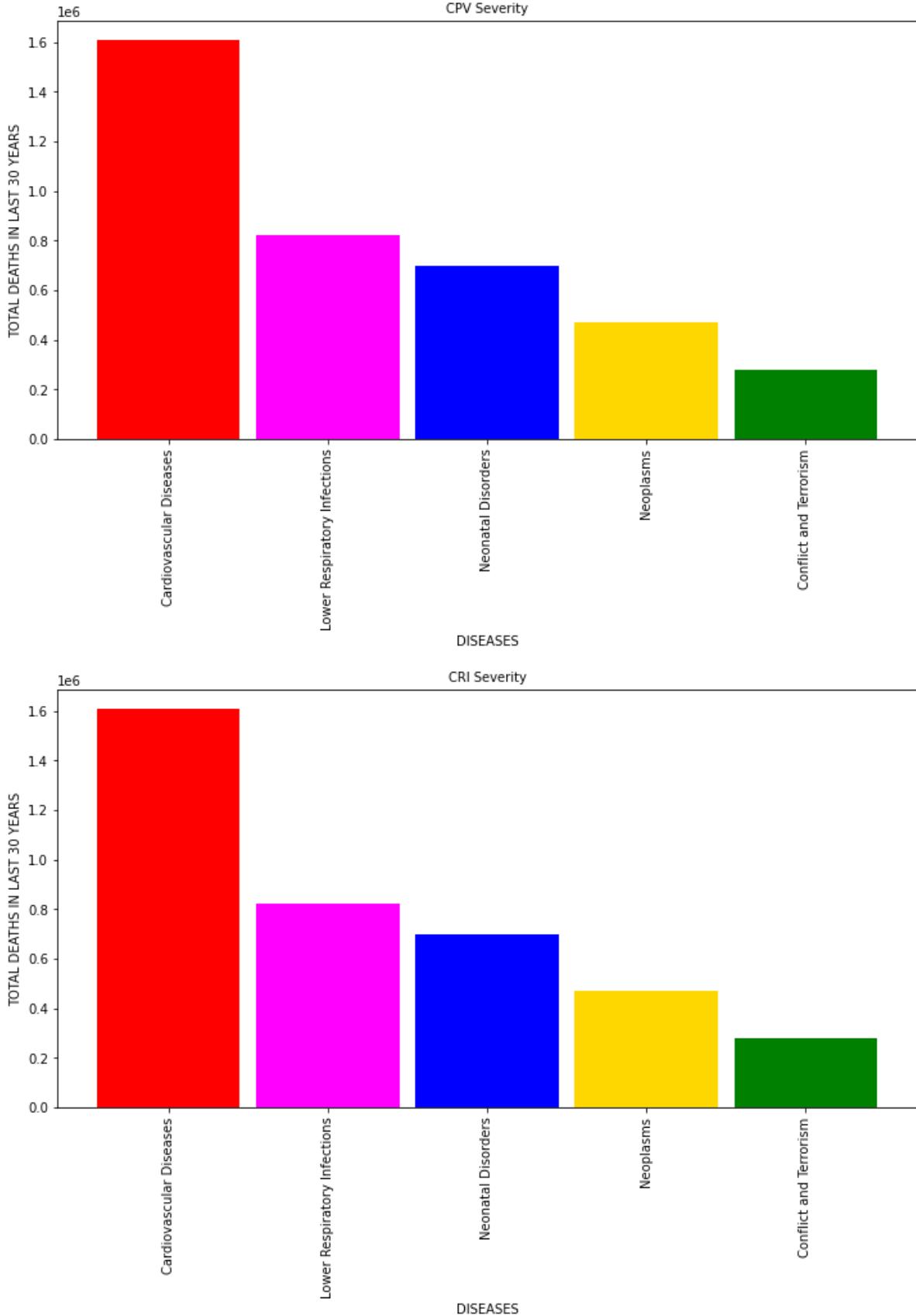
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



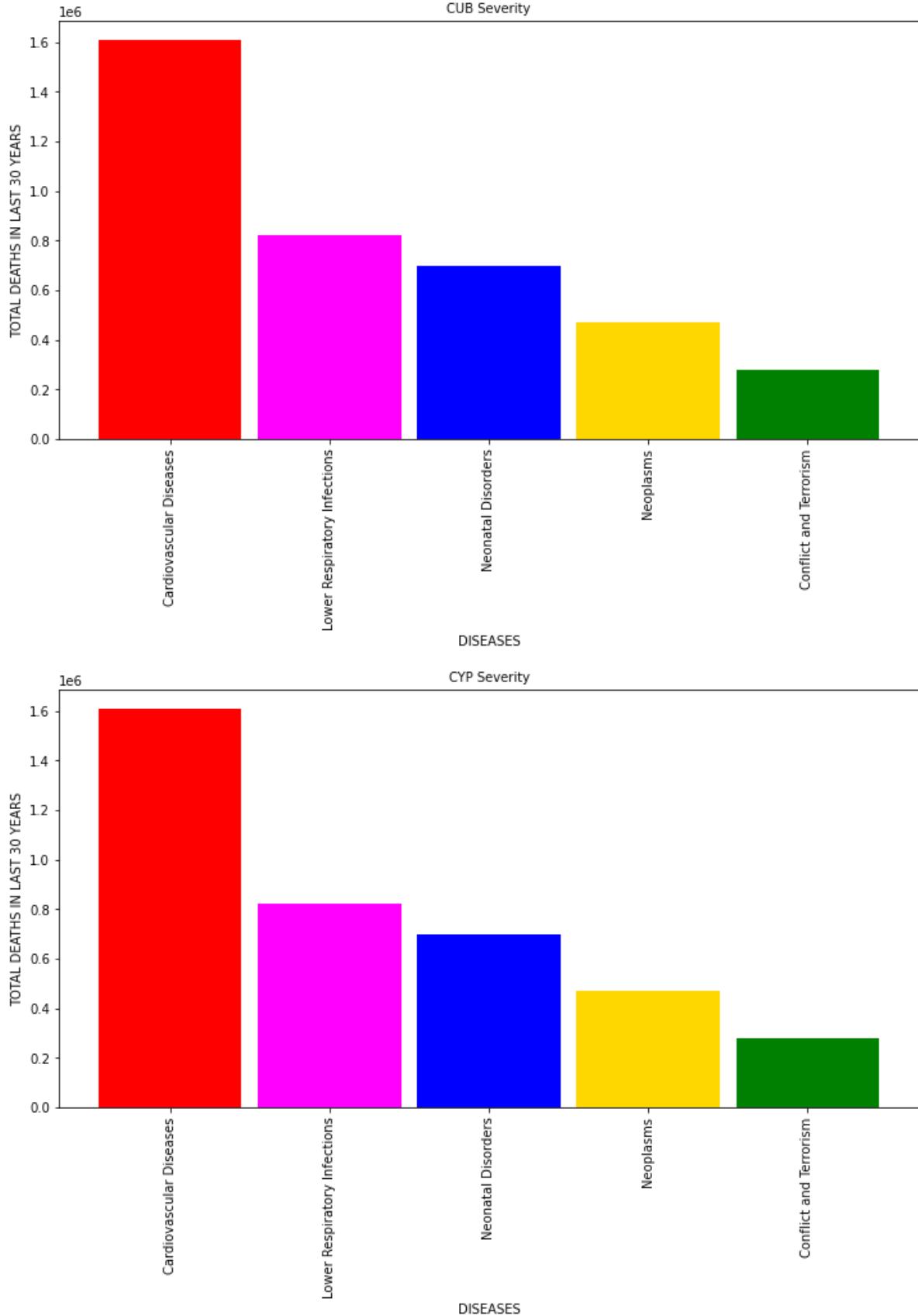
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



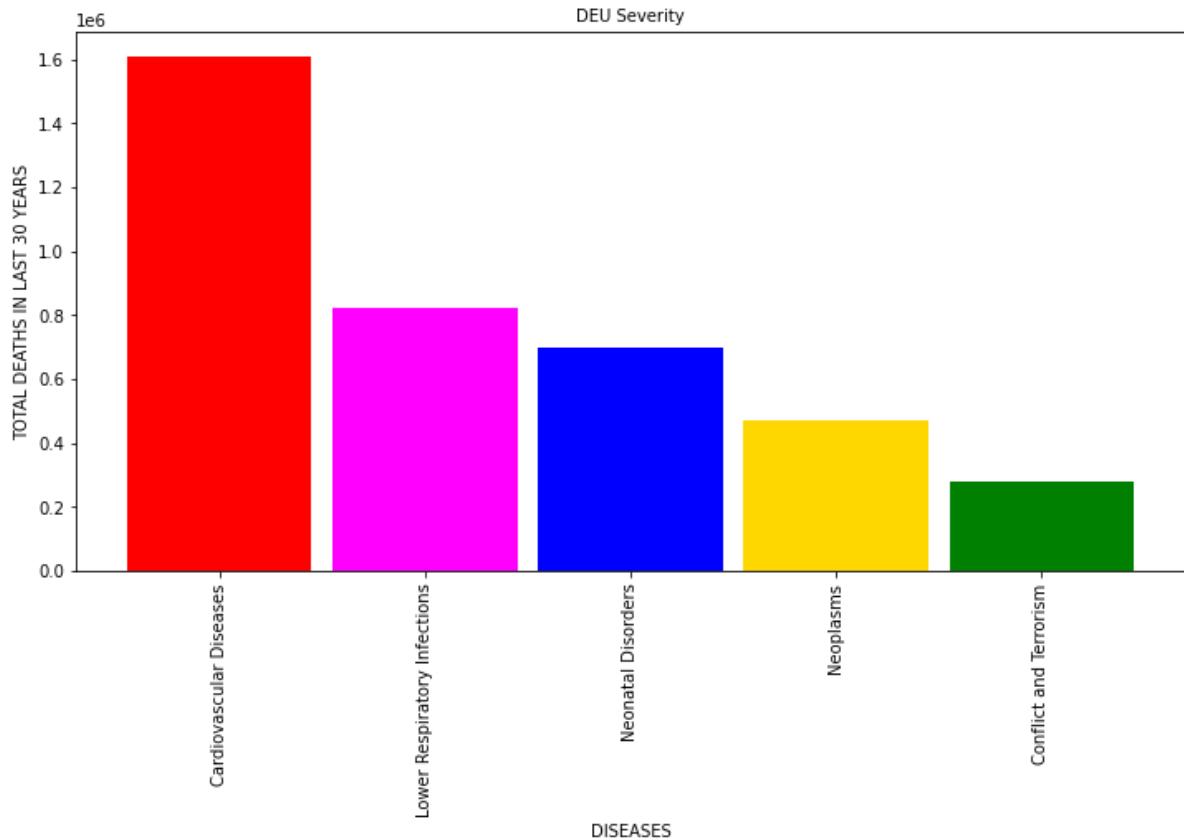
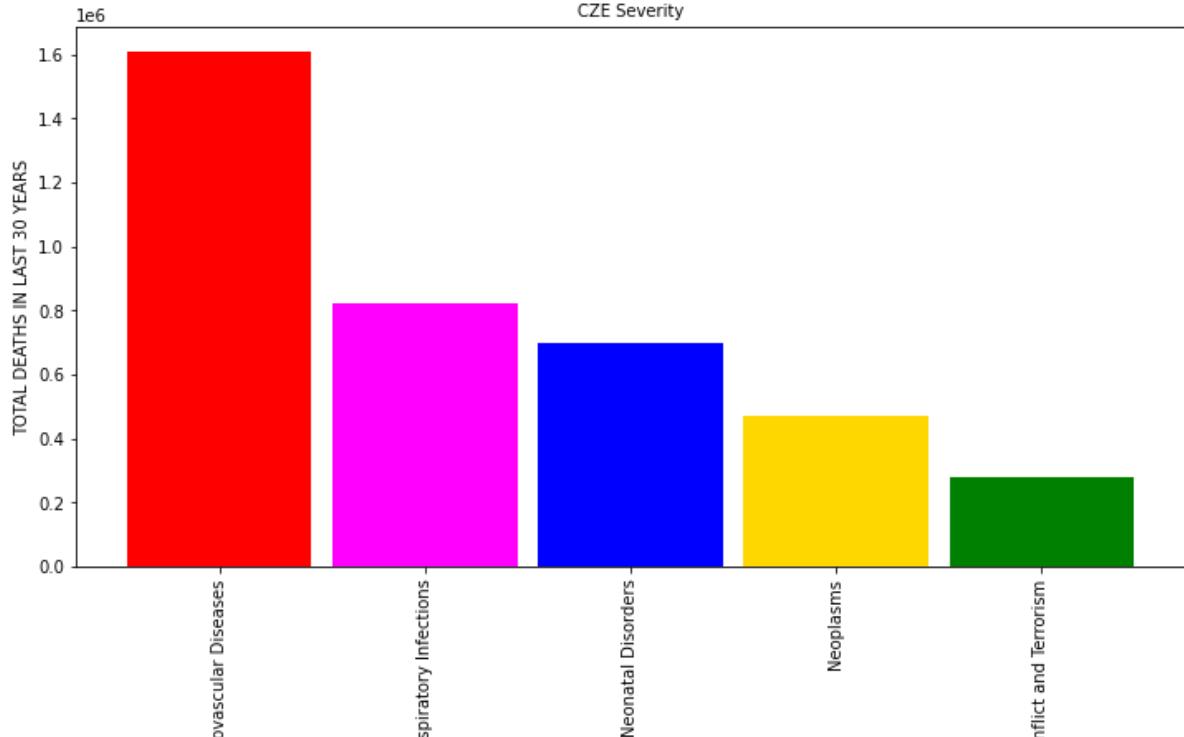
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



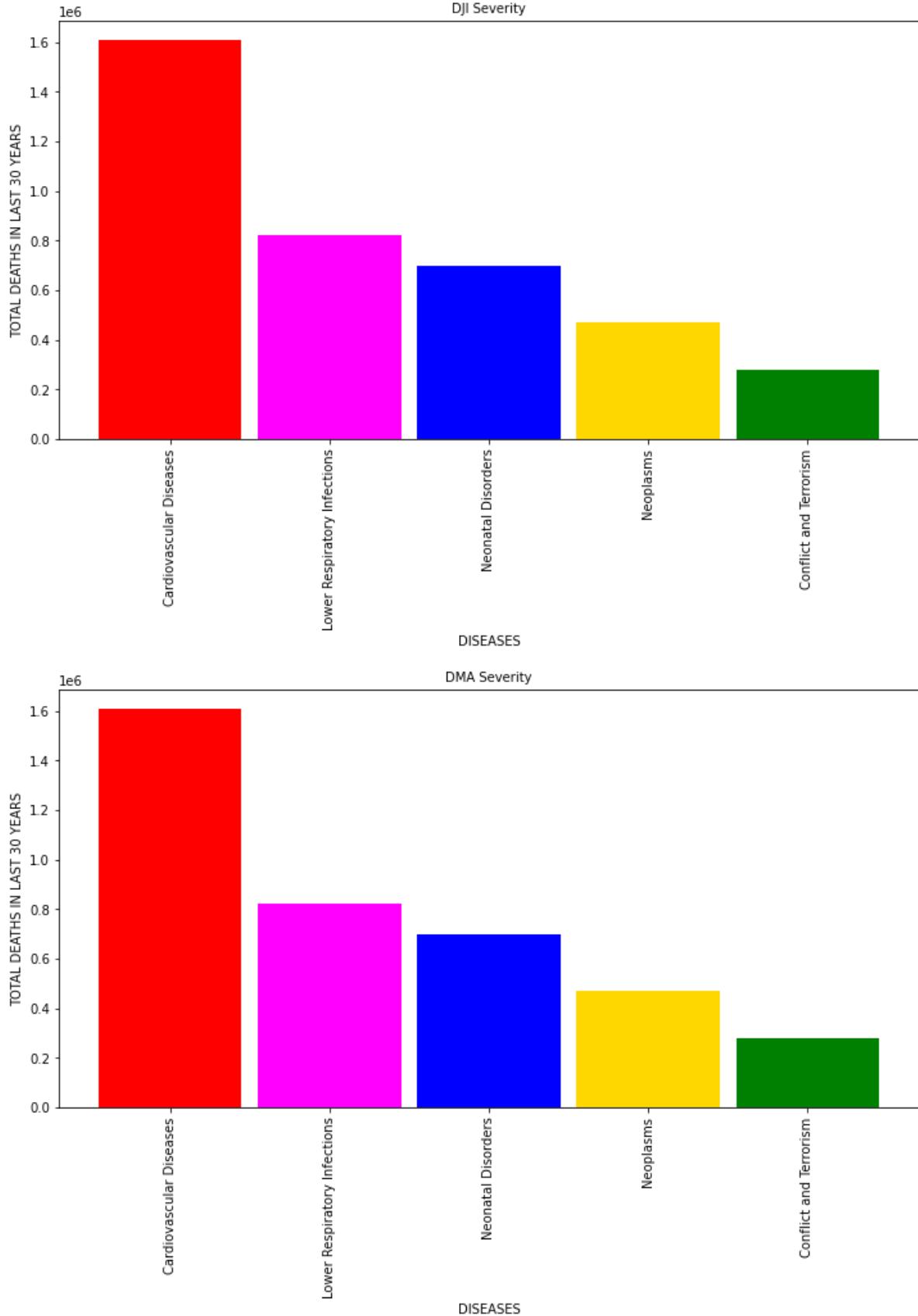
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death

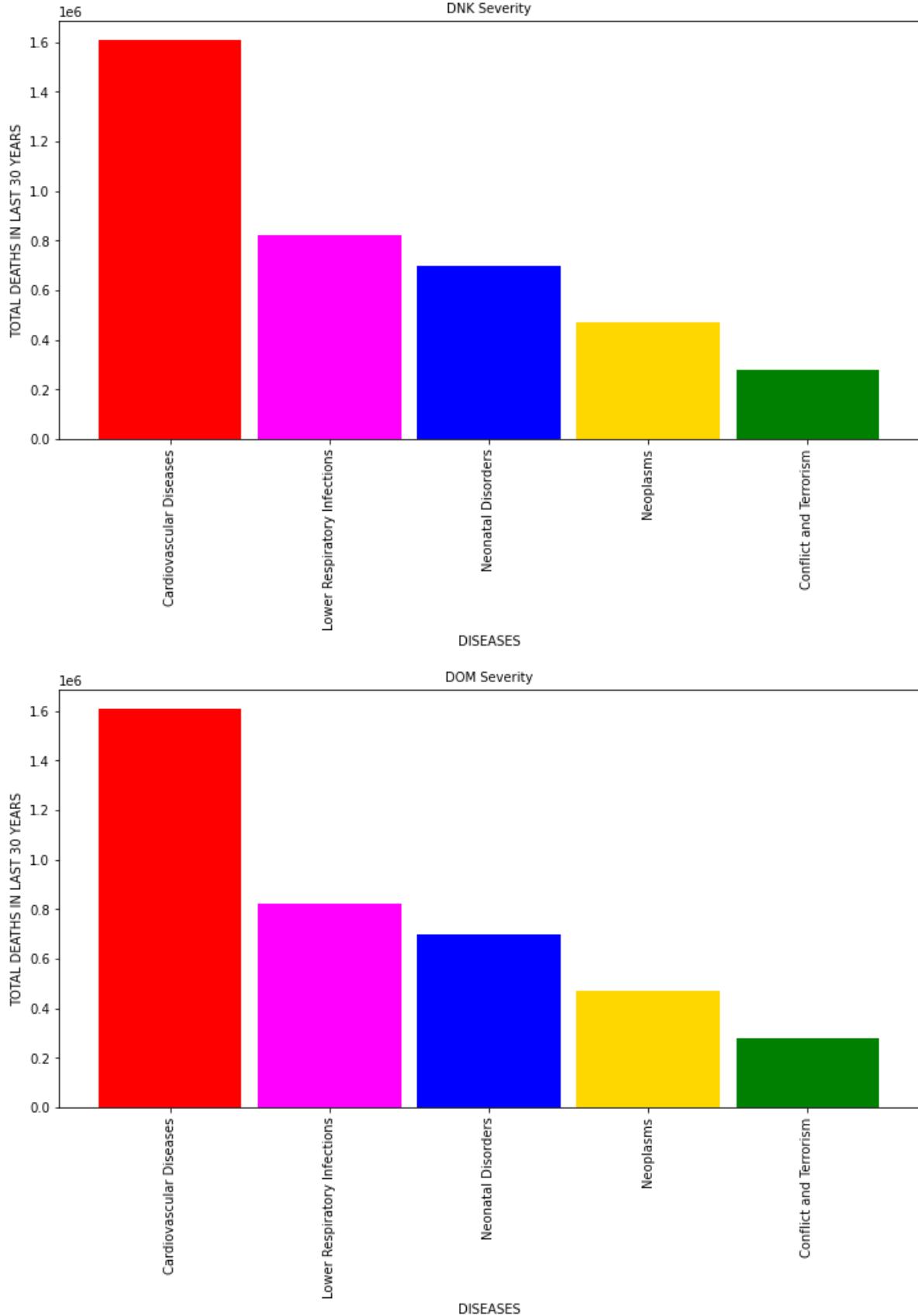


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

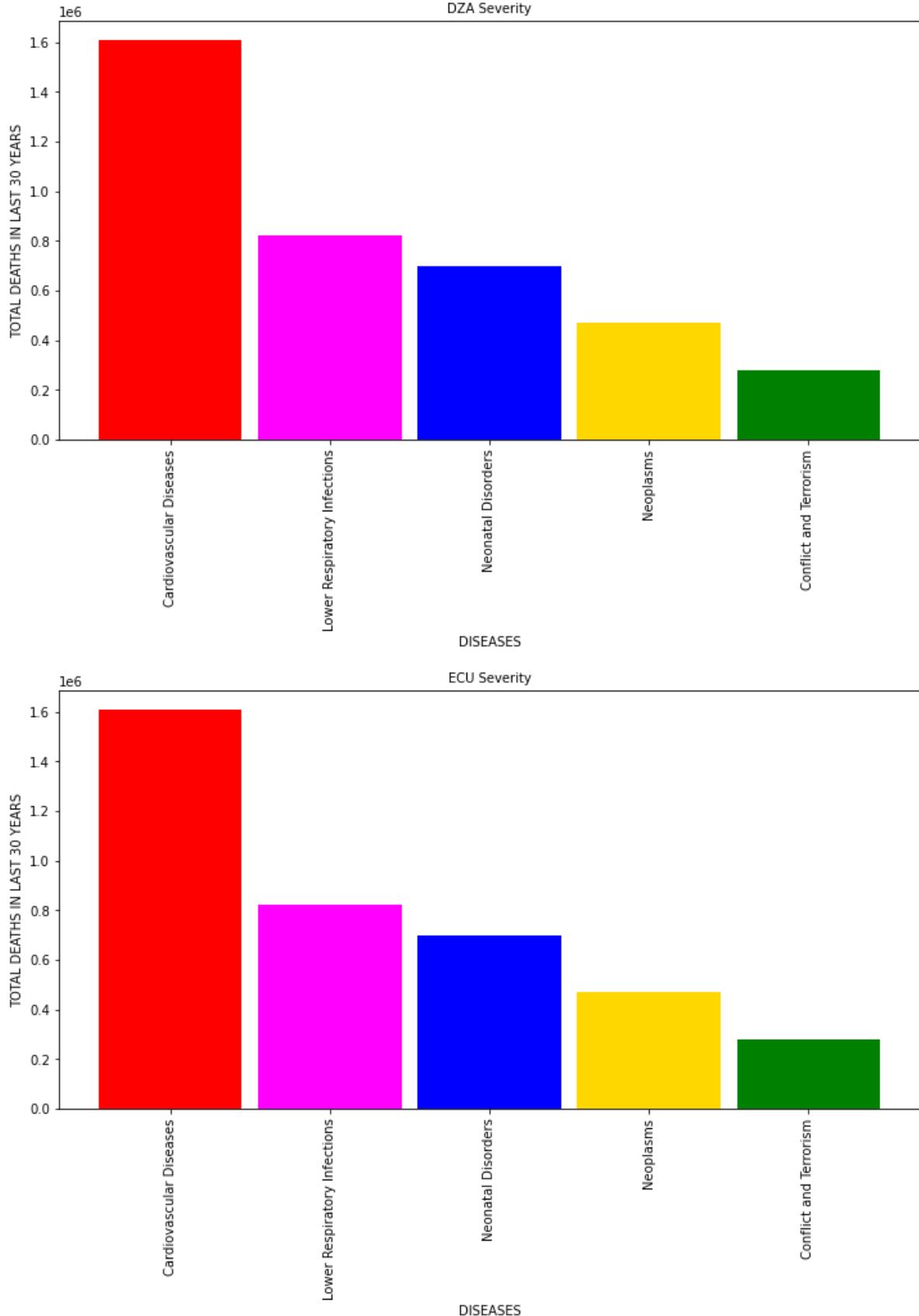
Cause Of Death



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

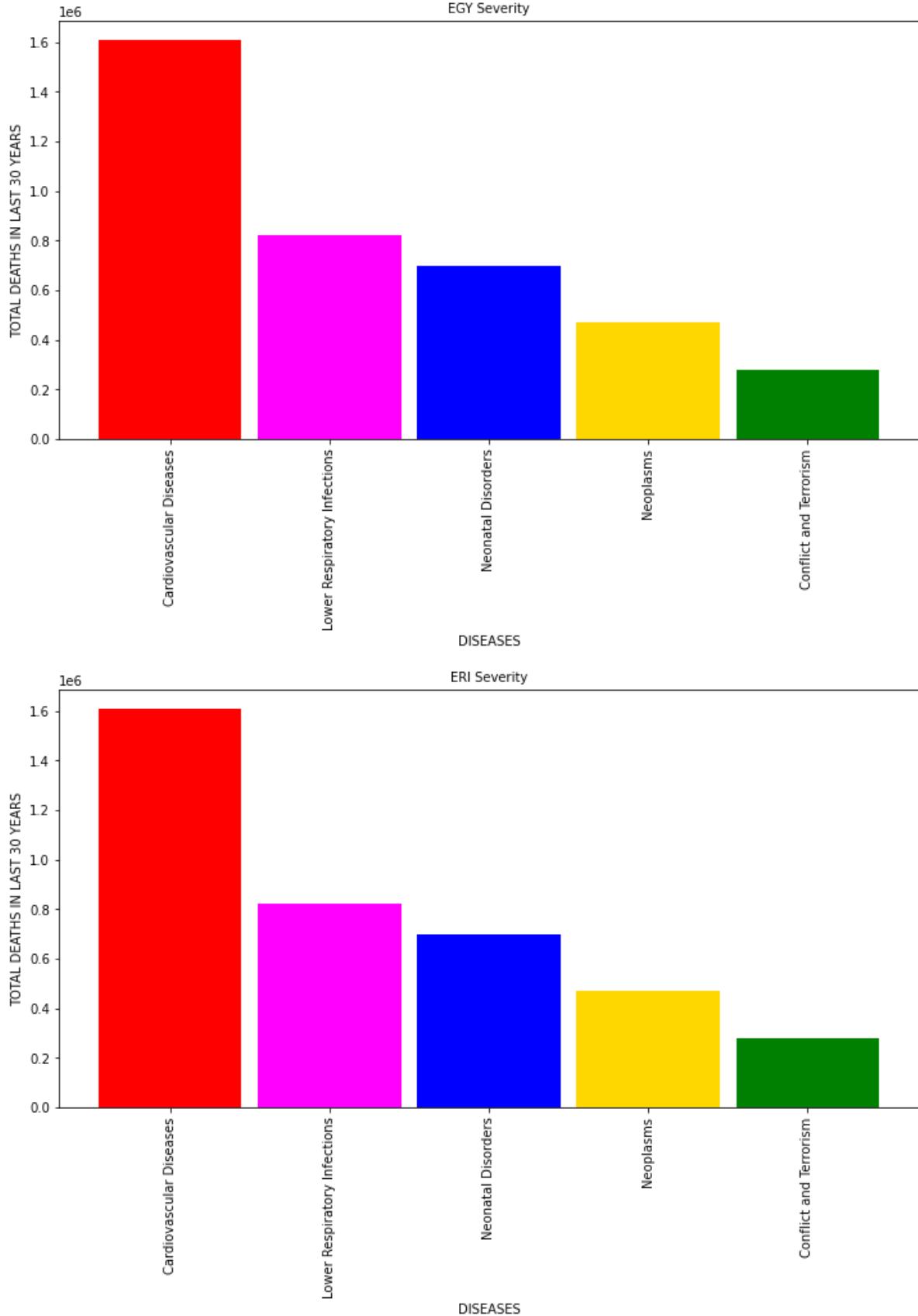


Cause Of Death



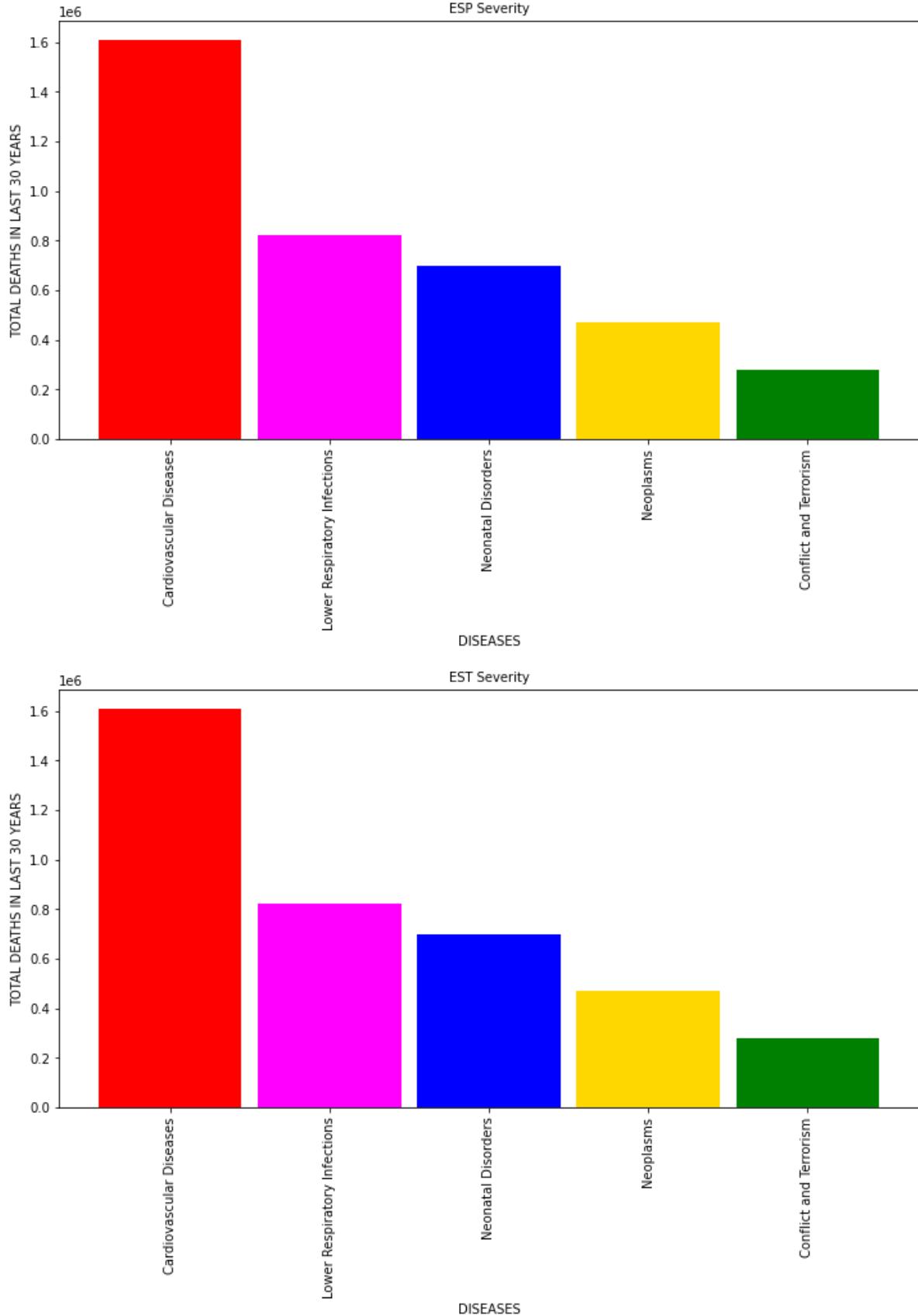
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



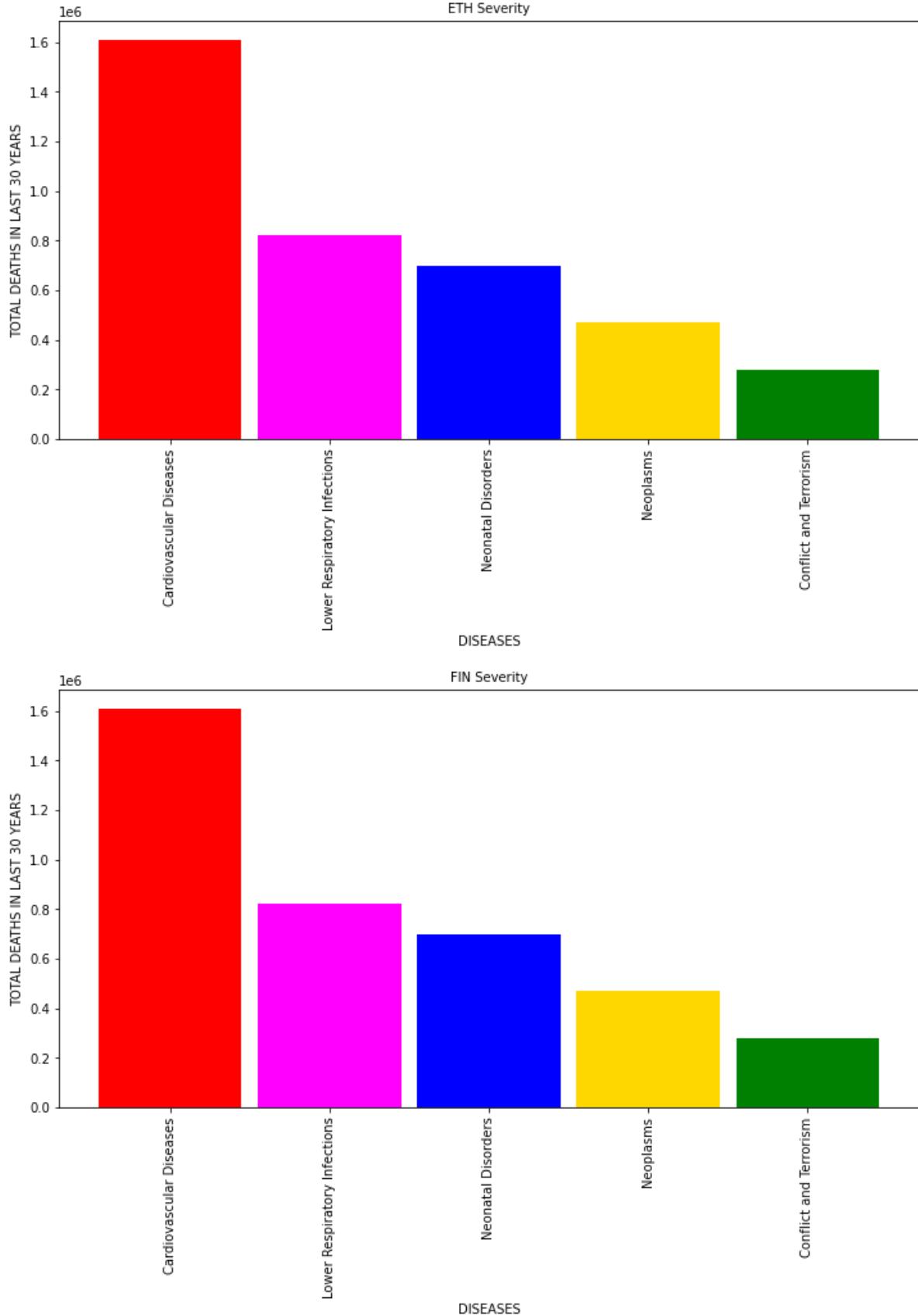
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



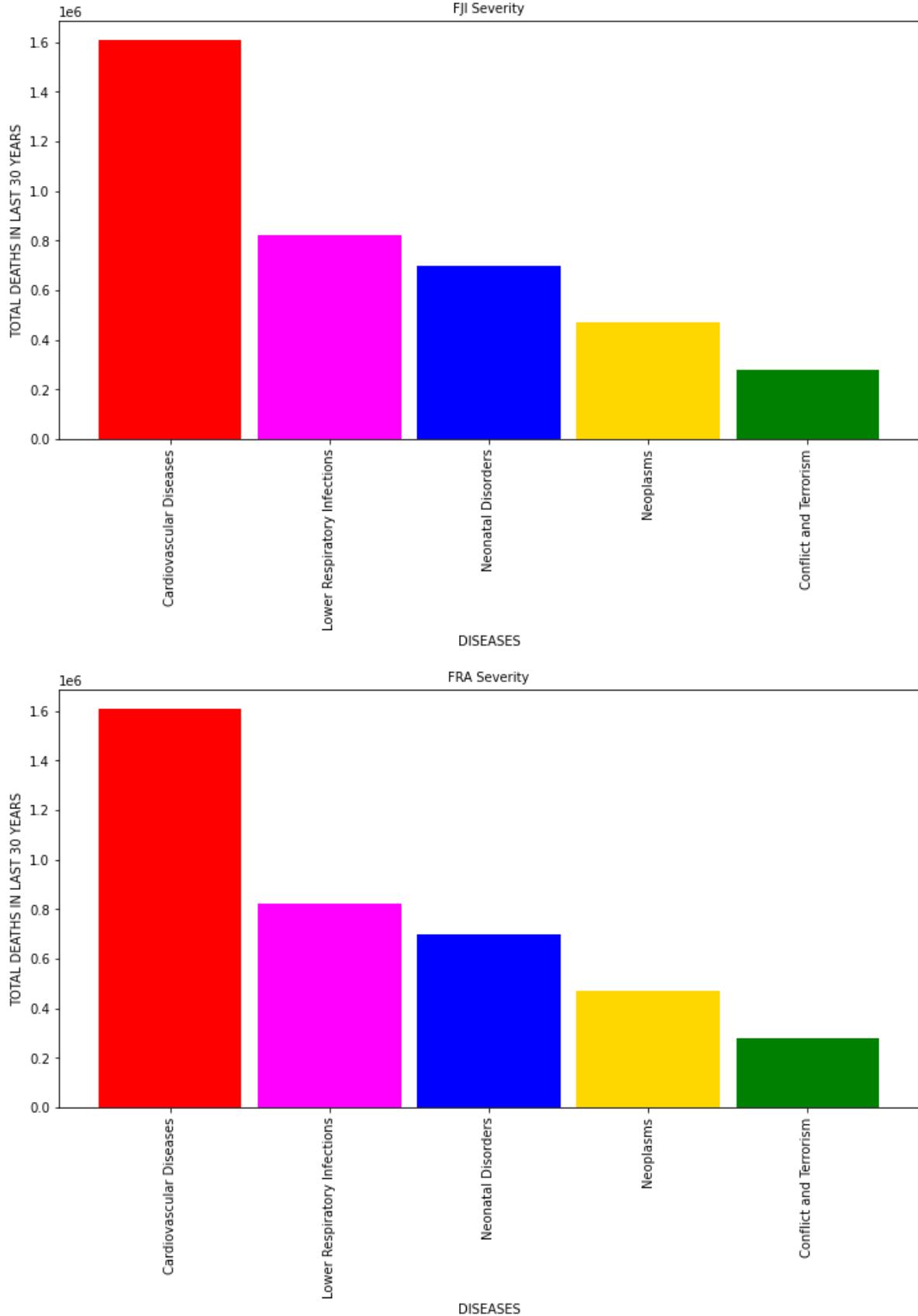
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



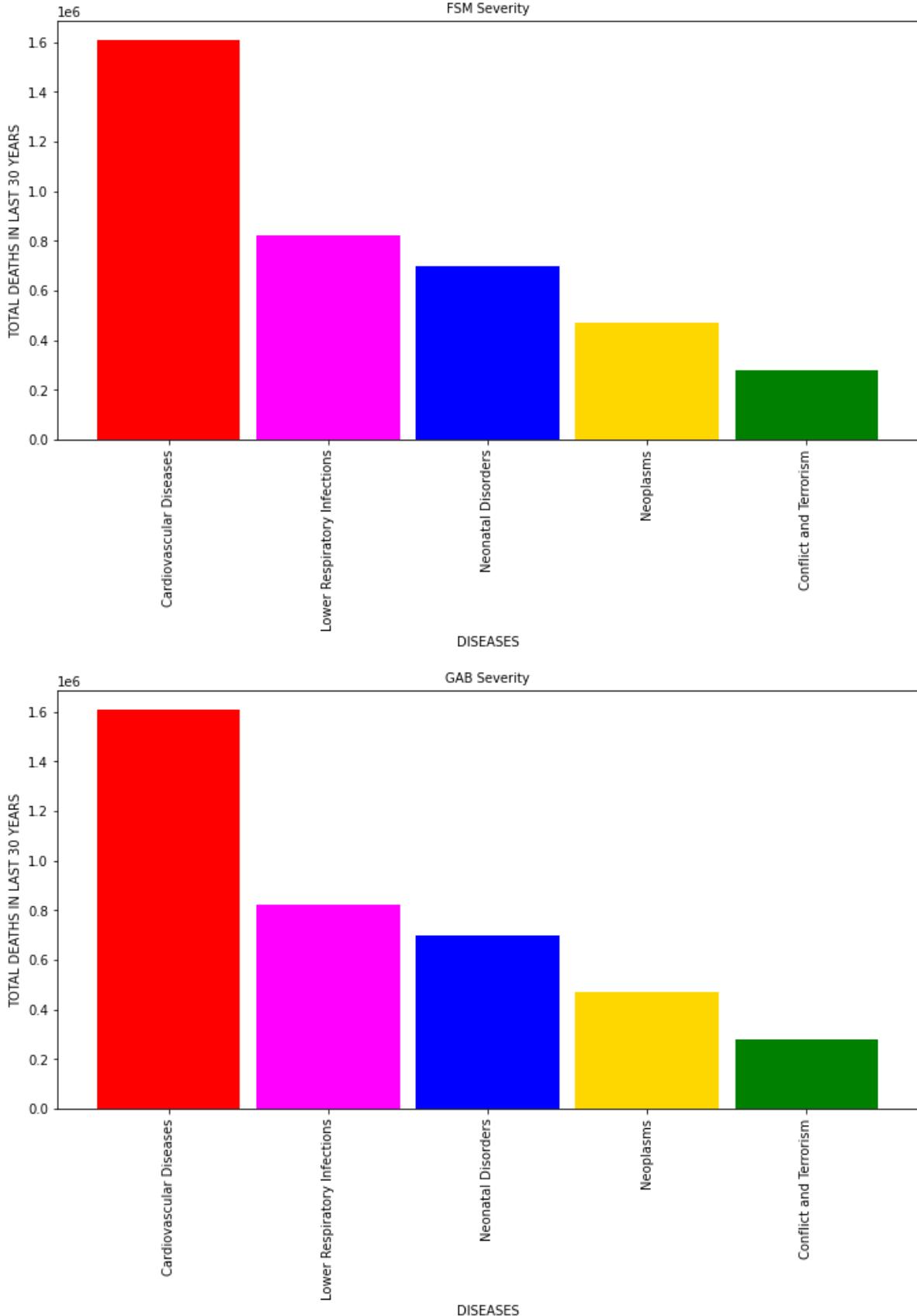
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



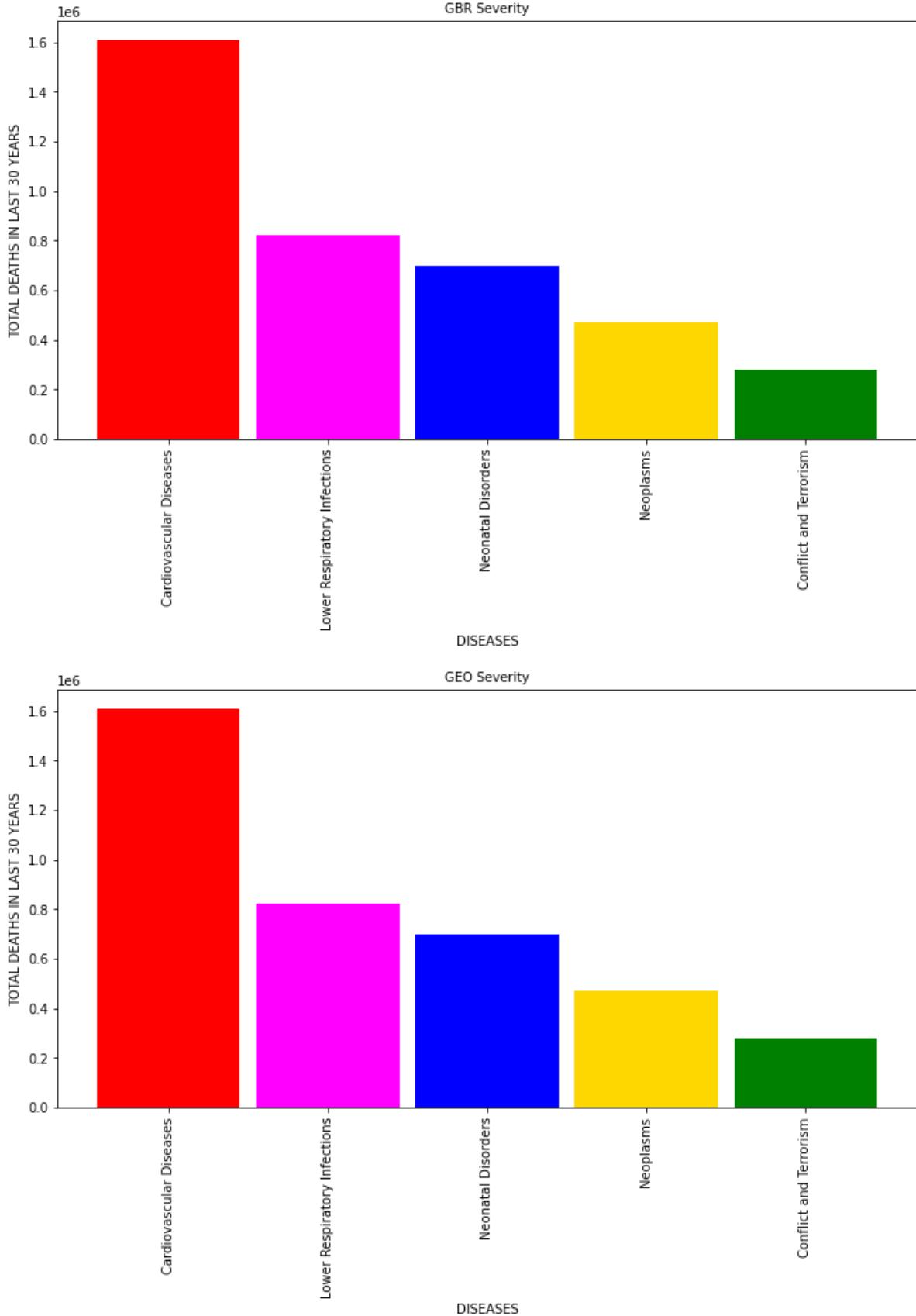
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



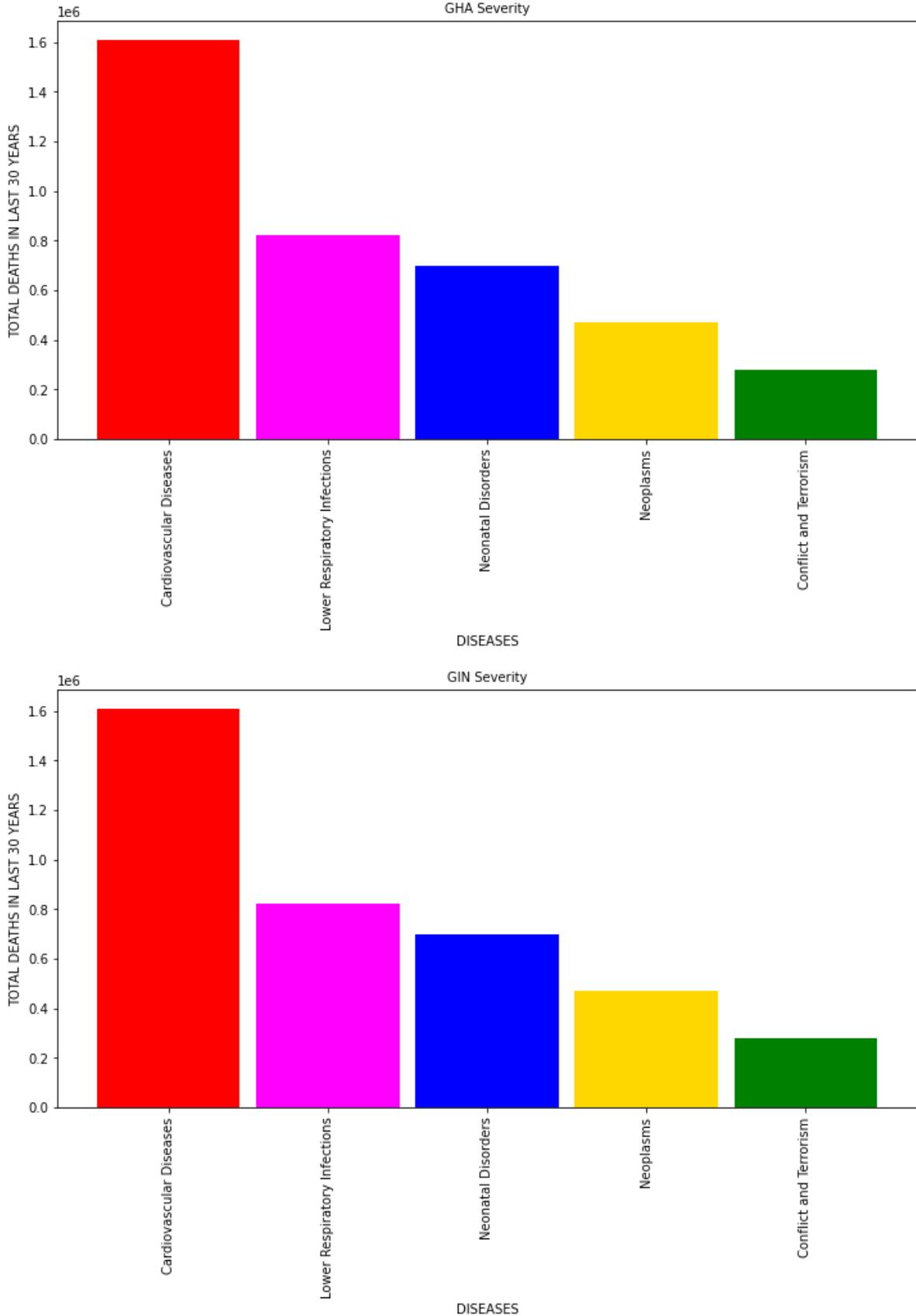
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



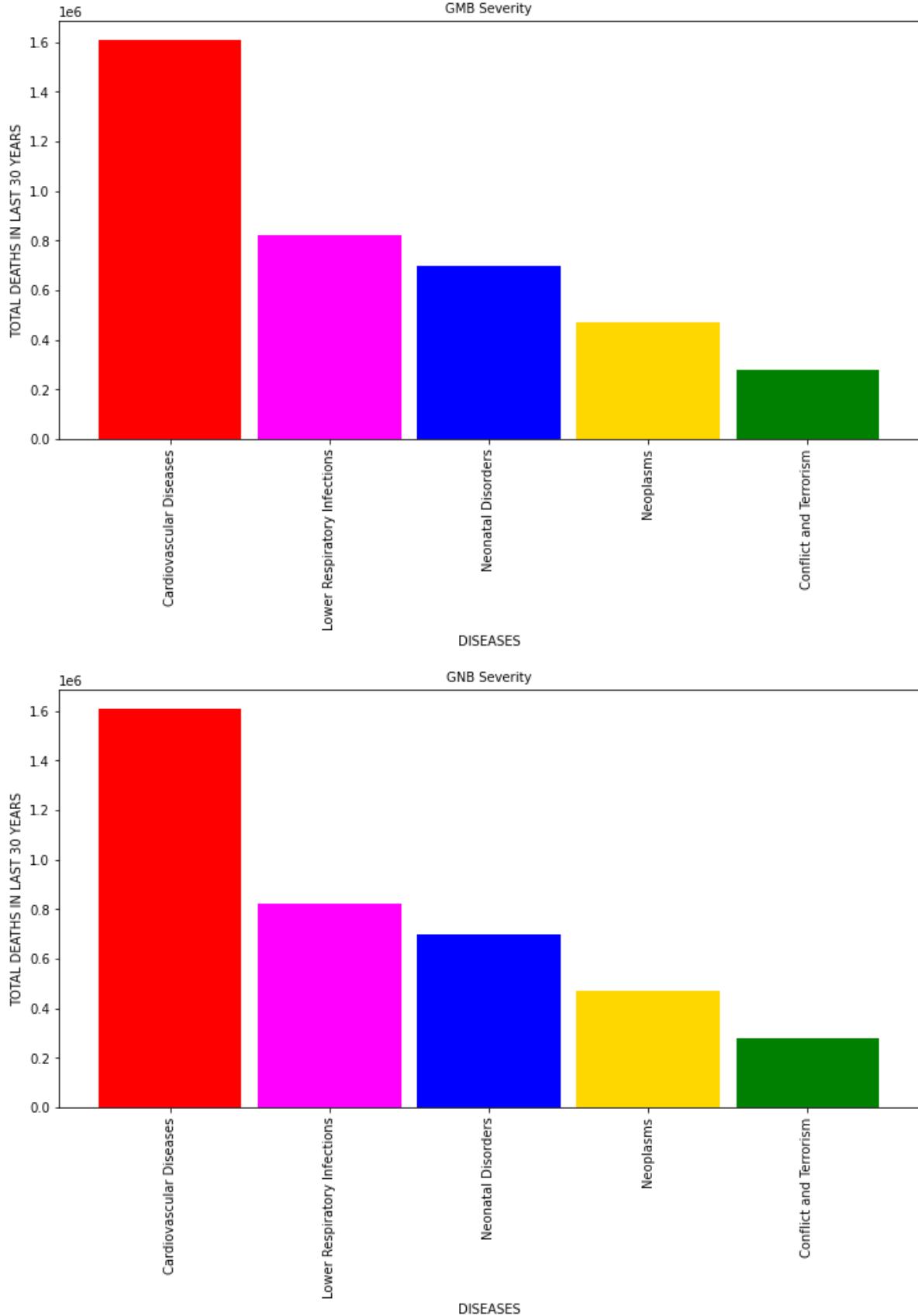
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



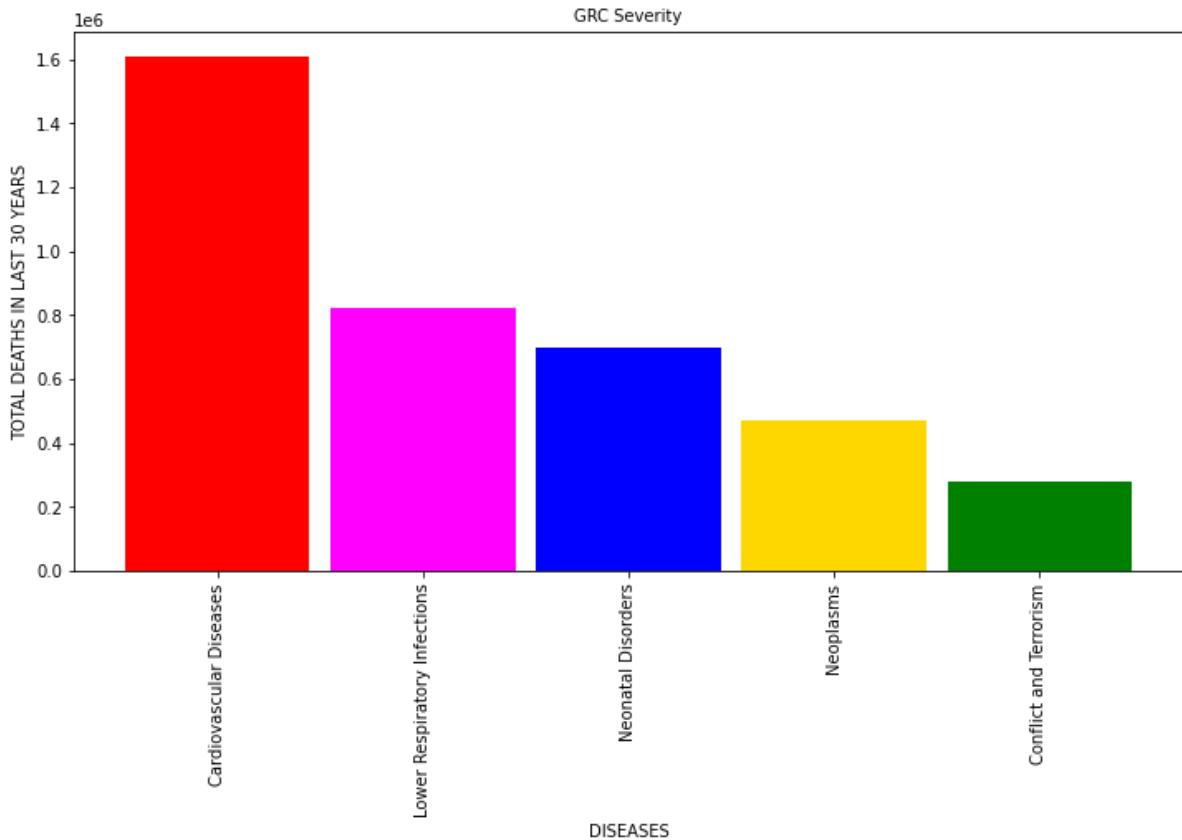
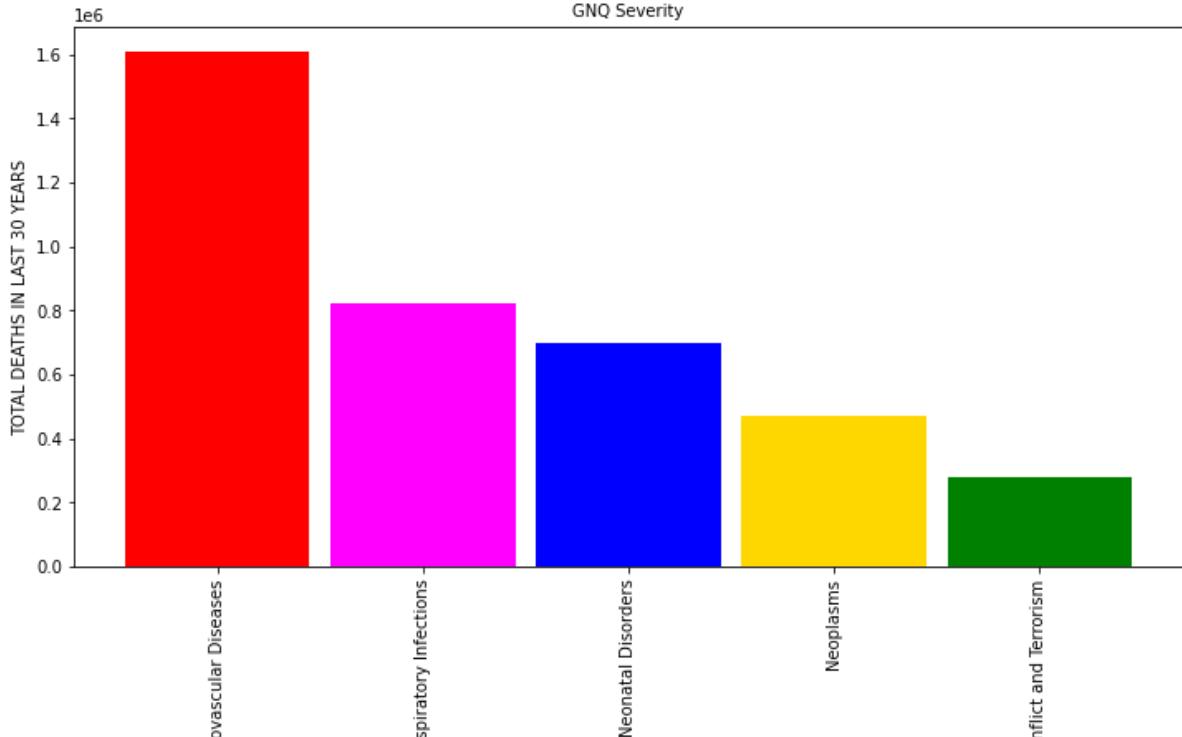
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



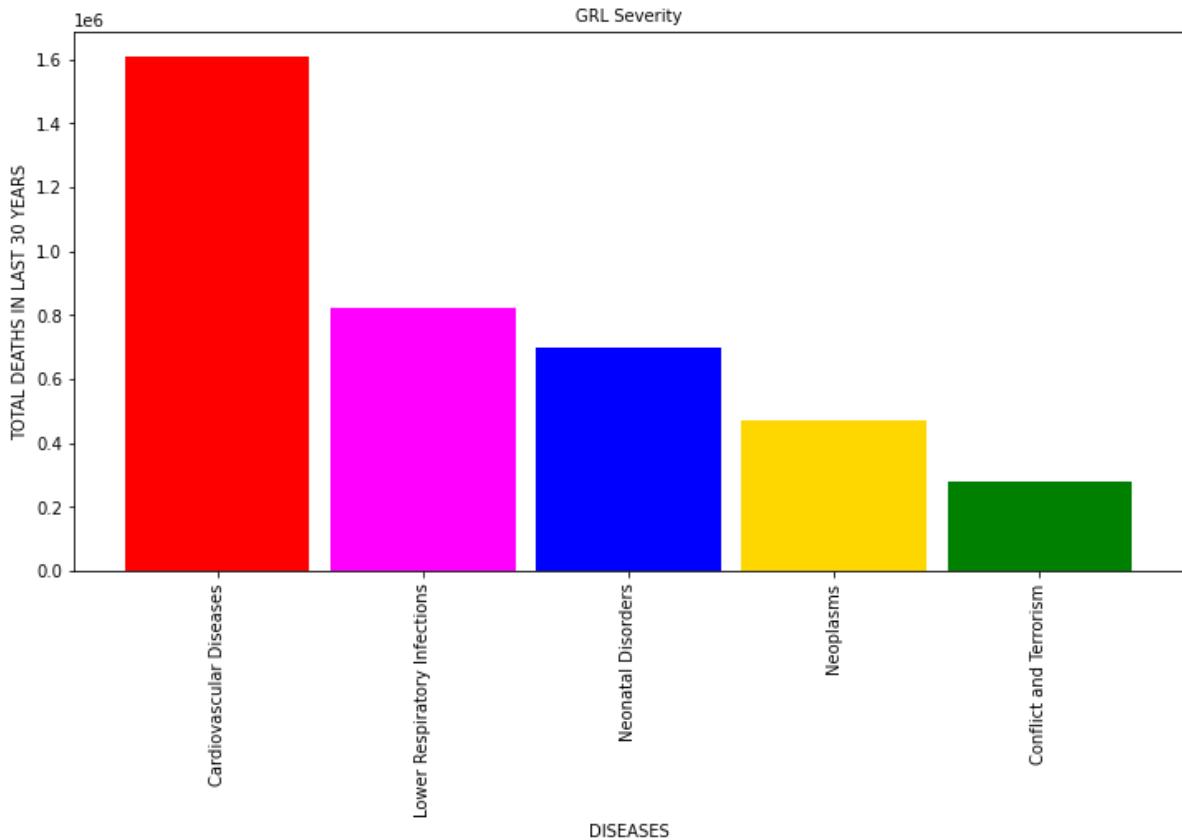
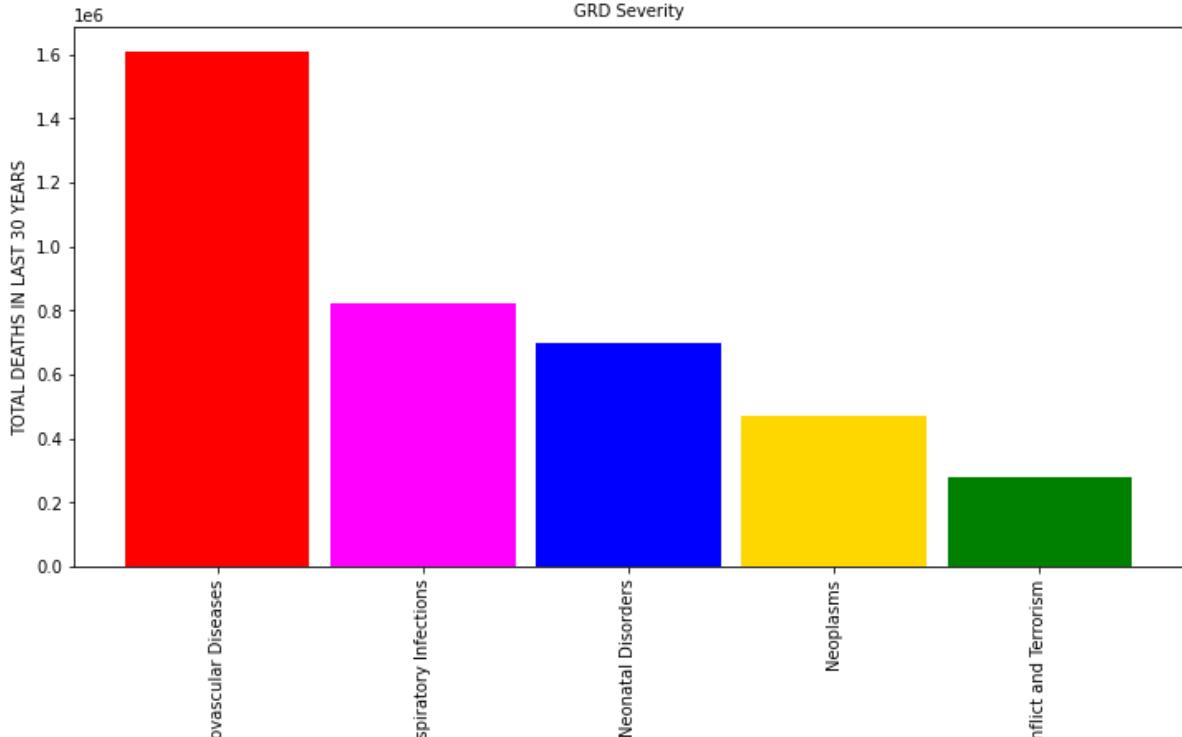
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



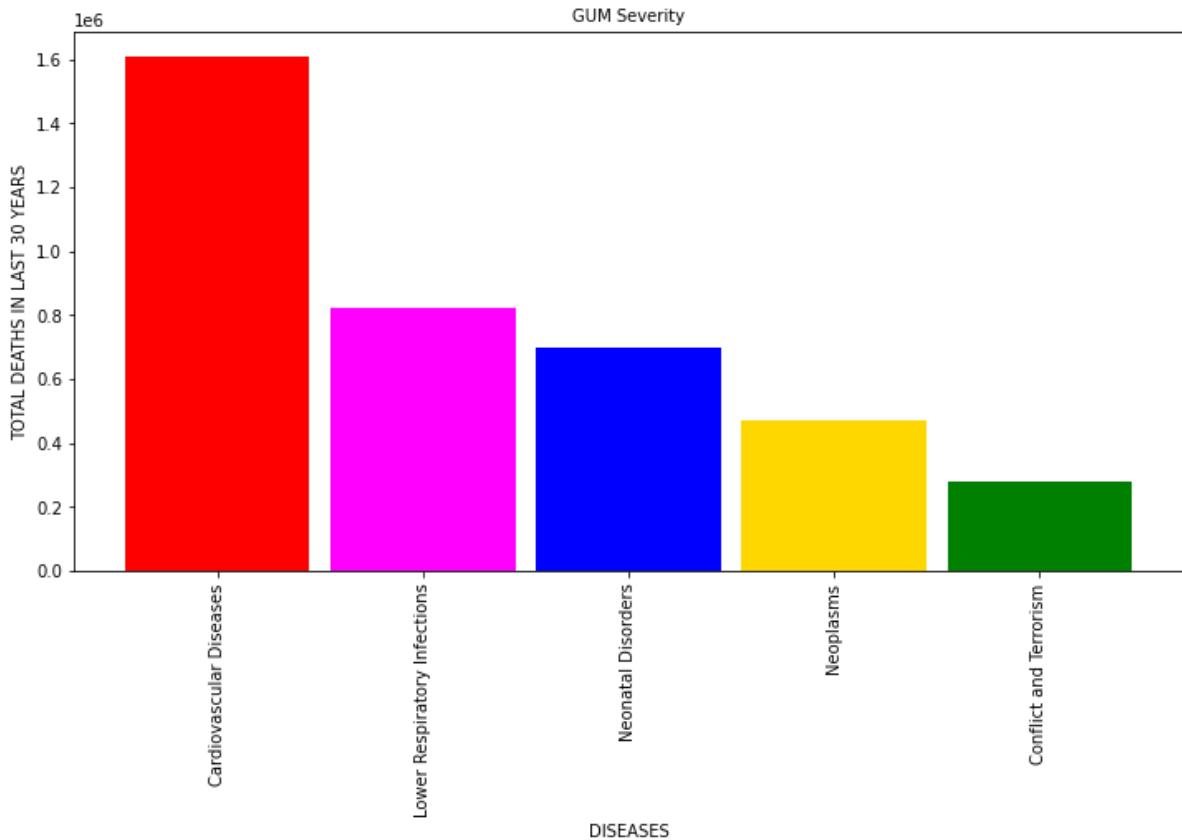
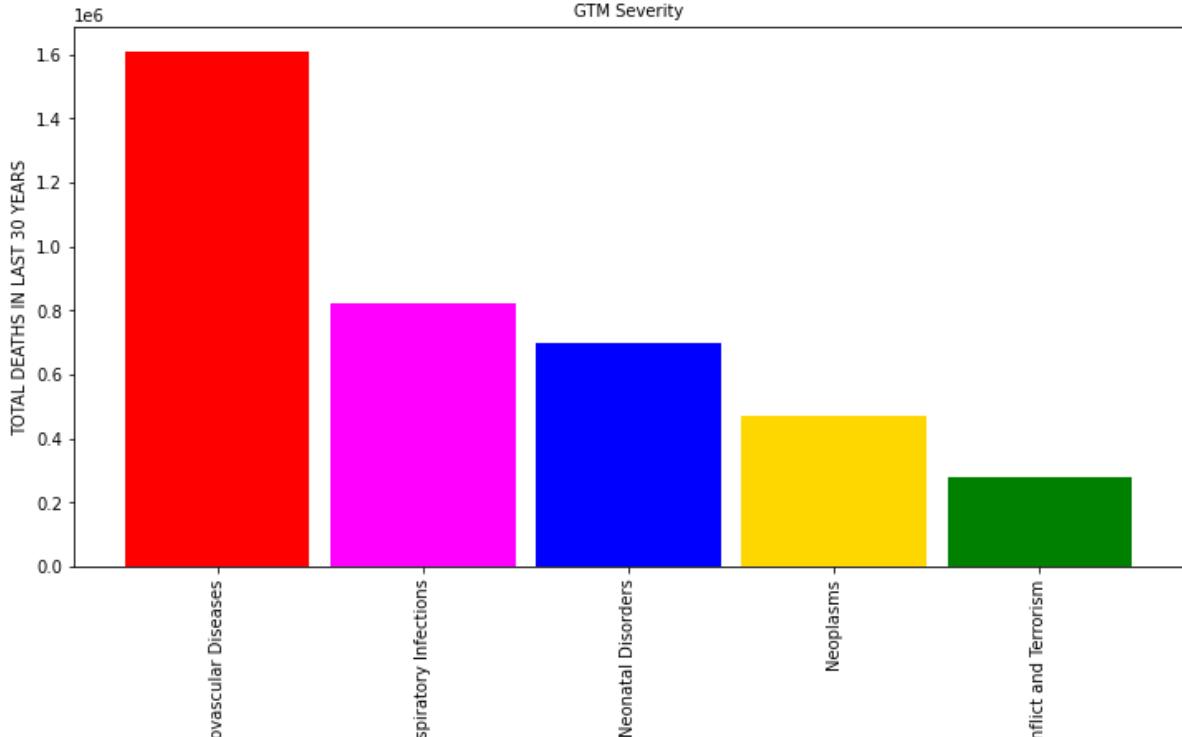
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



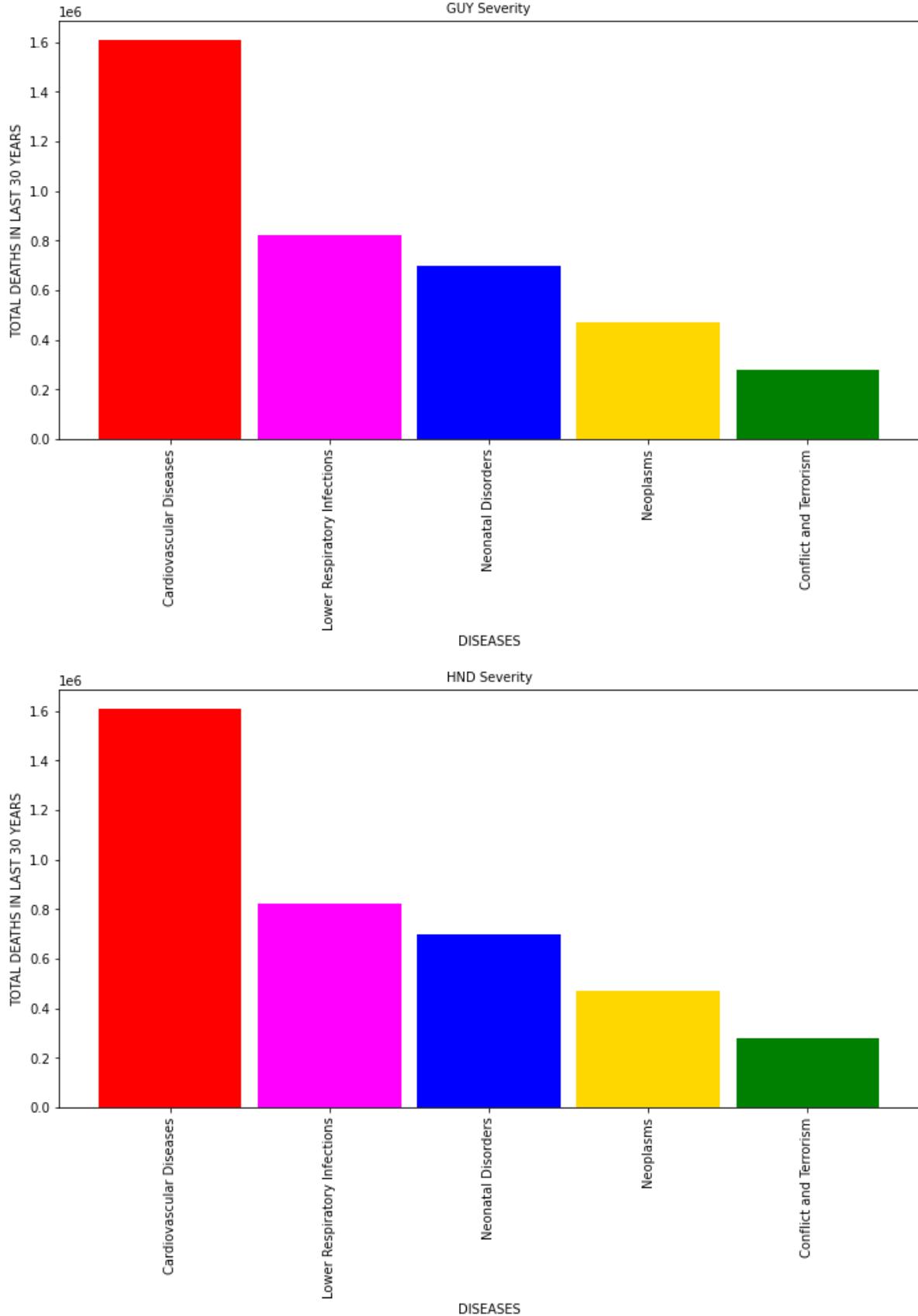
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death

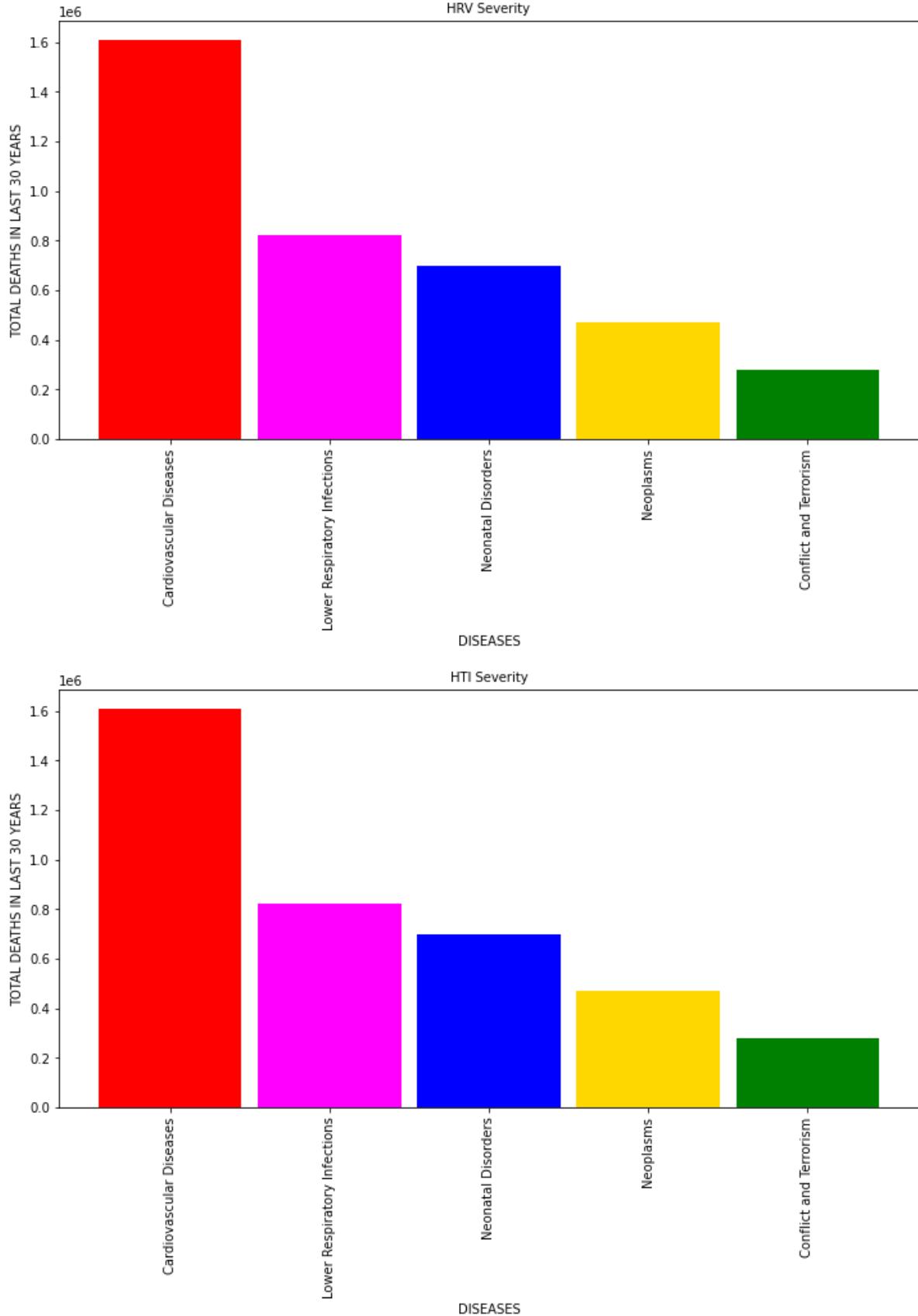


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

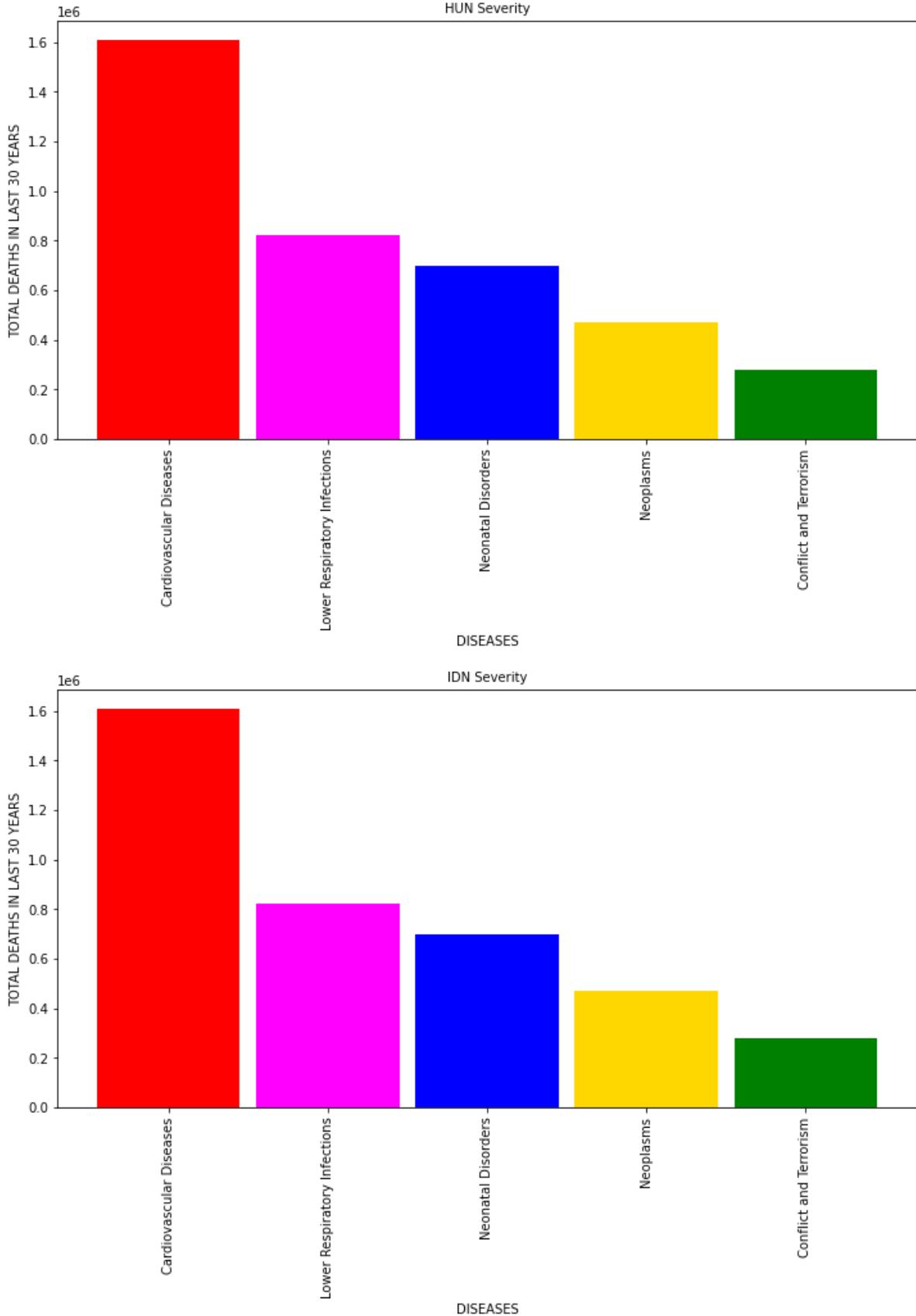
Cause Of Death



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

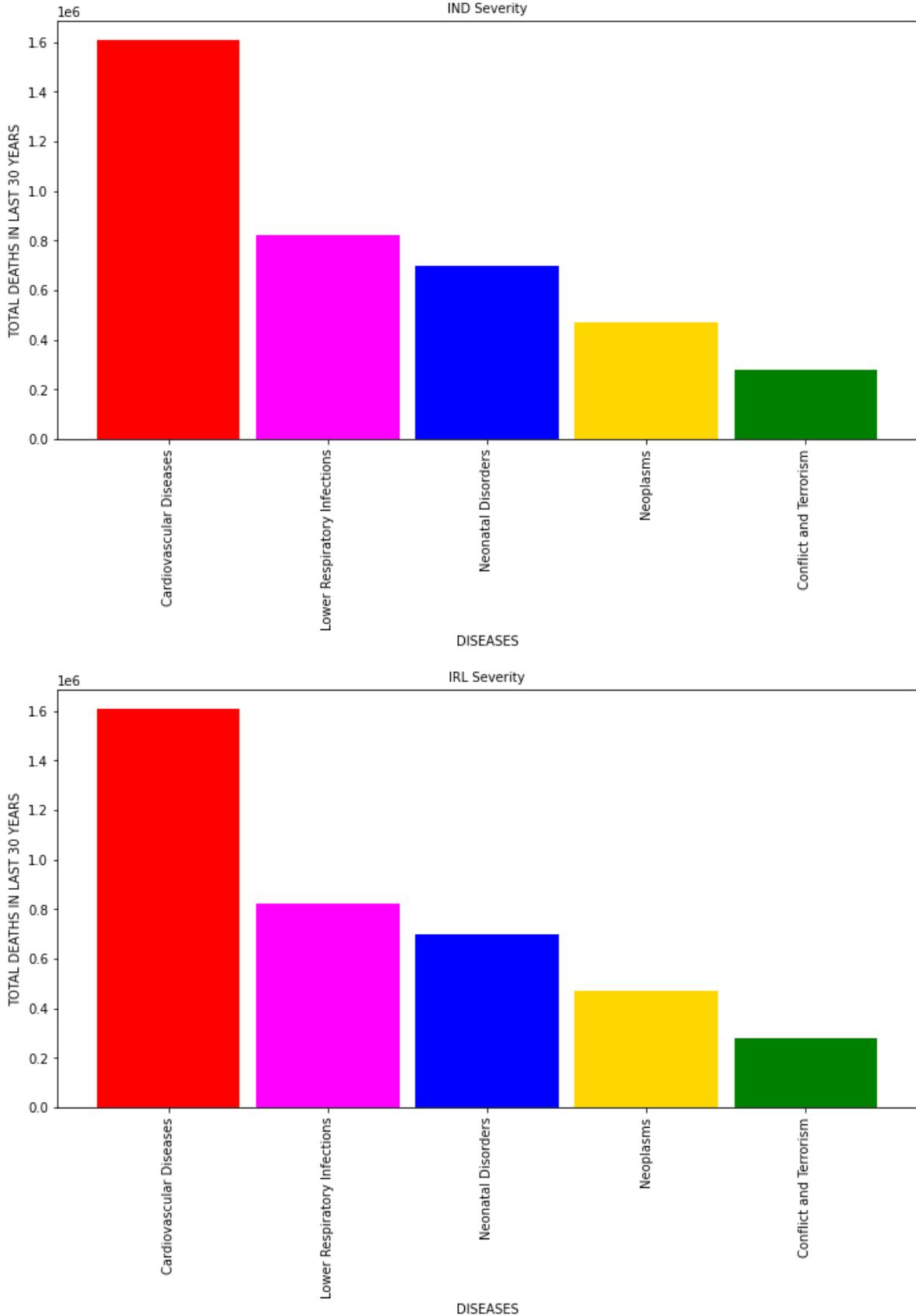


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



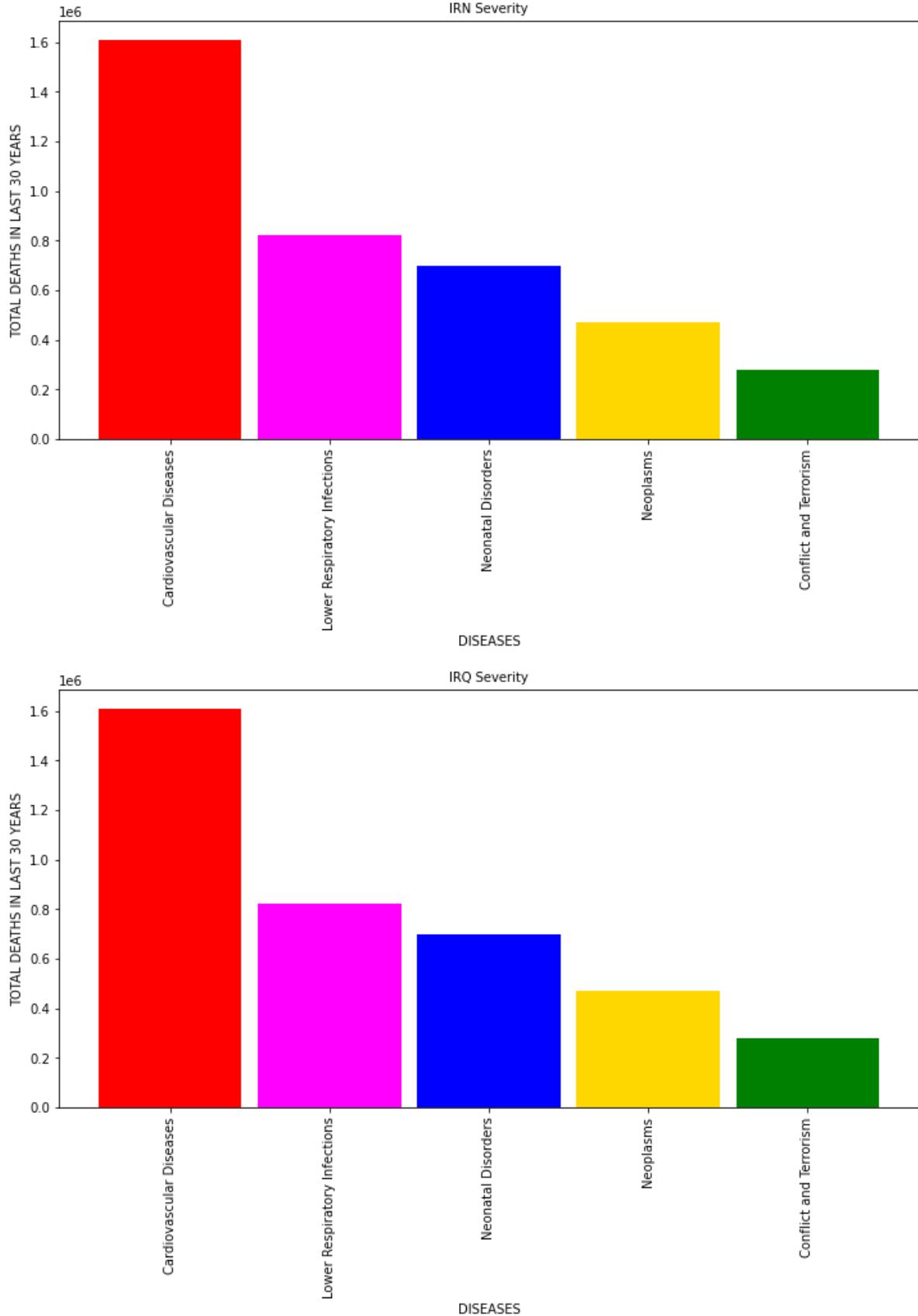
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



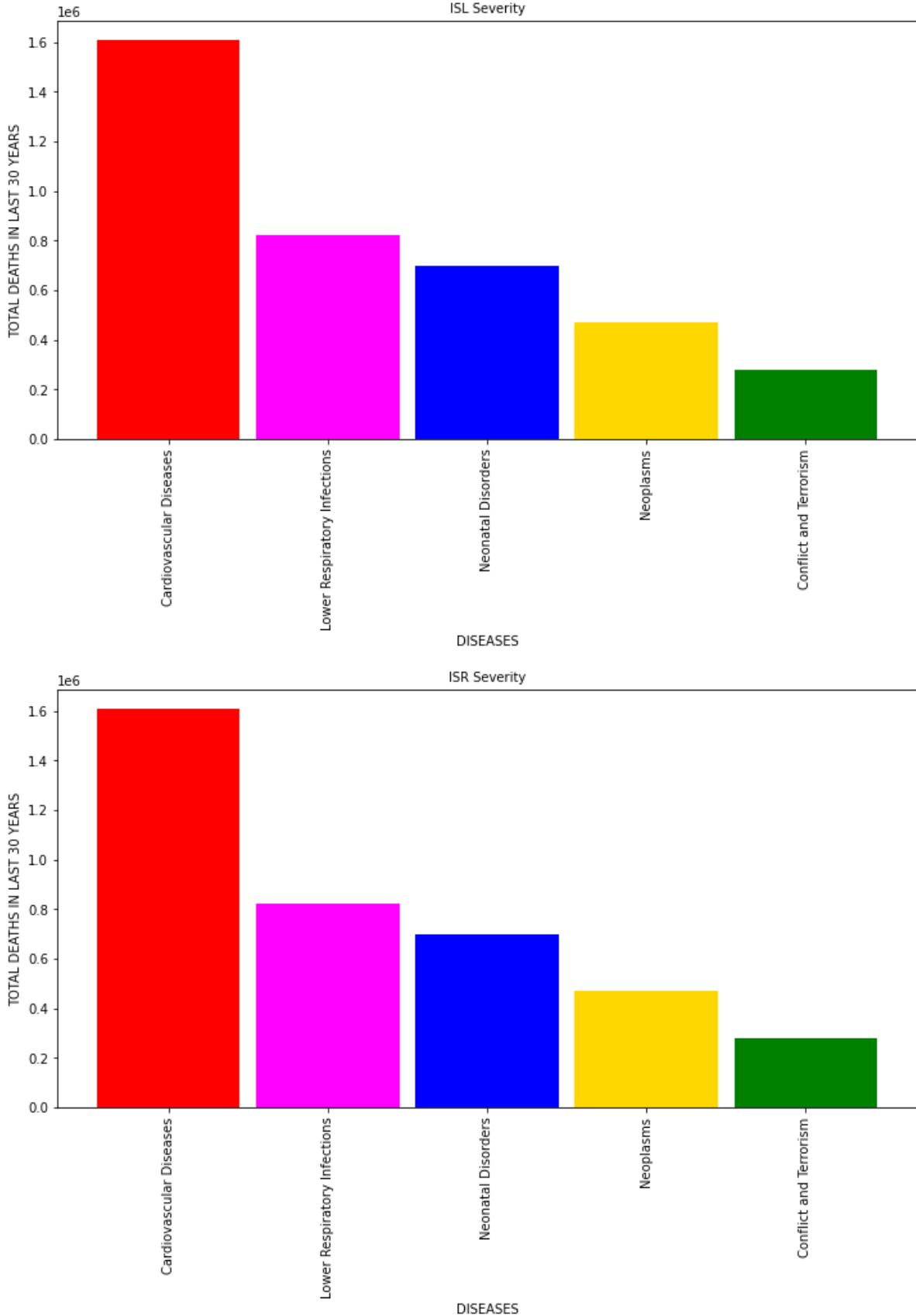
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



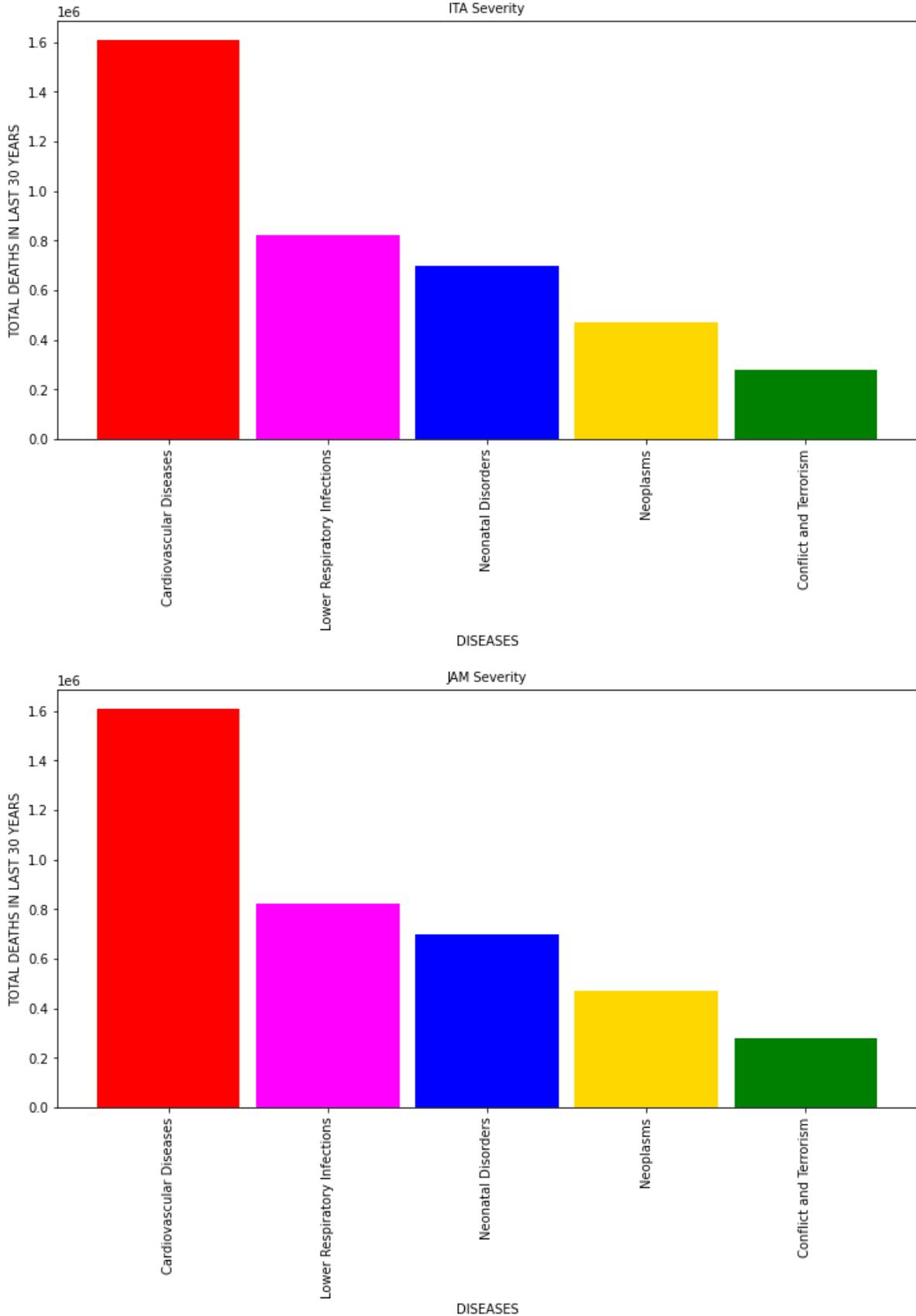
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



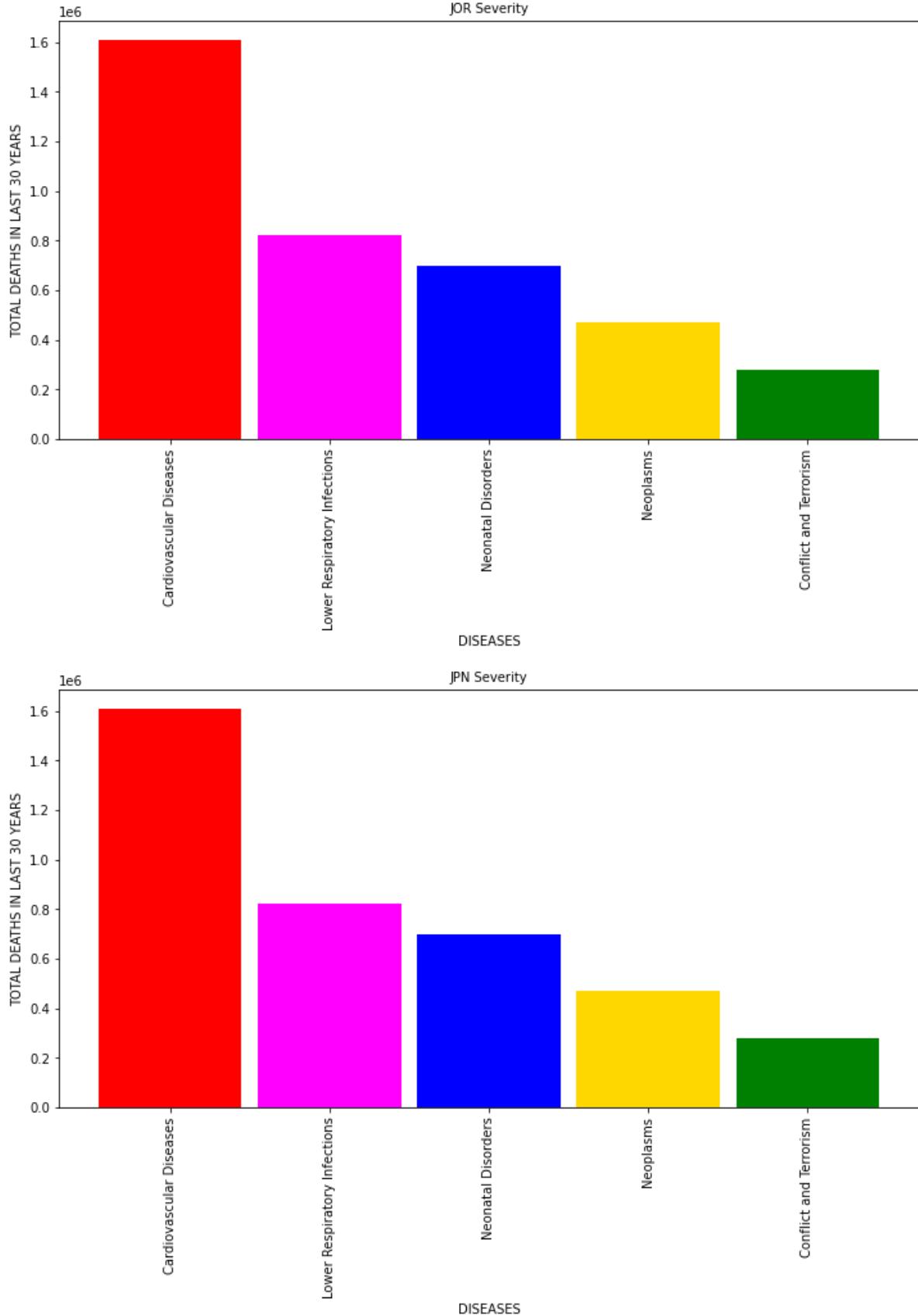
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death

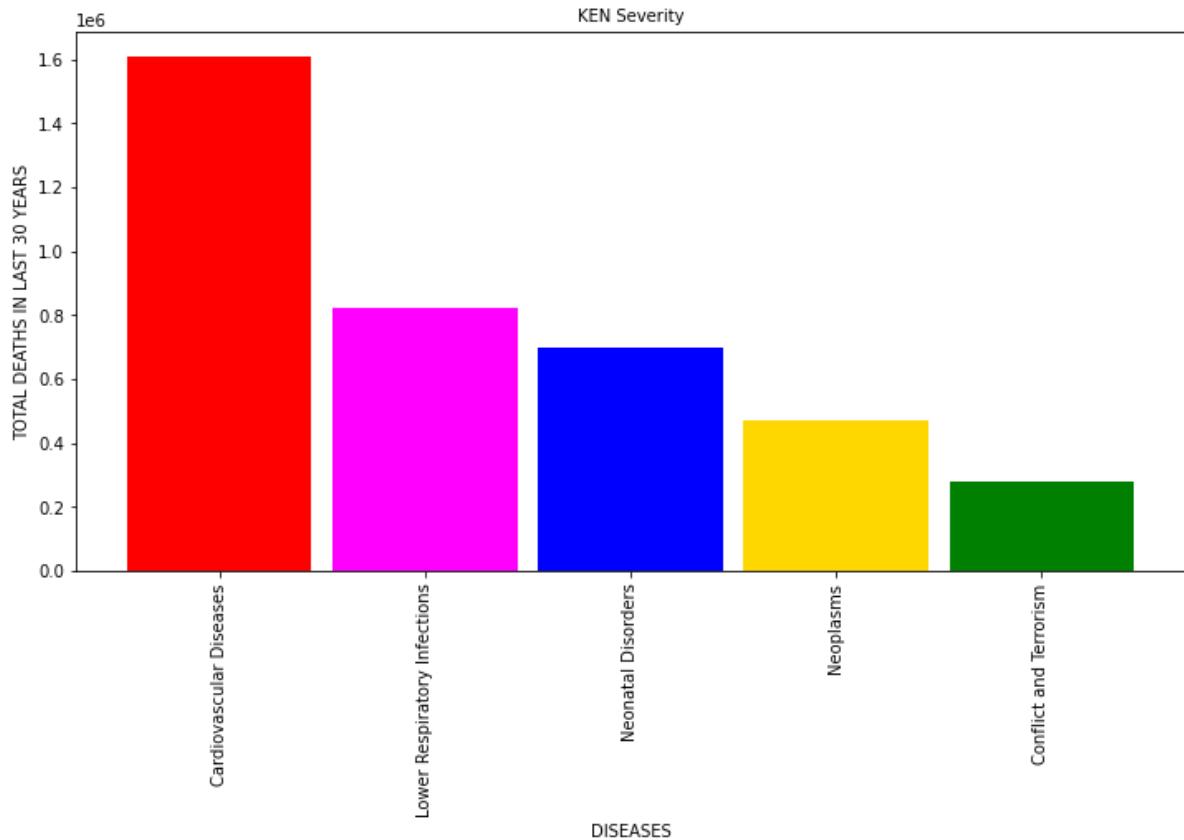
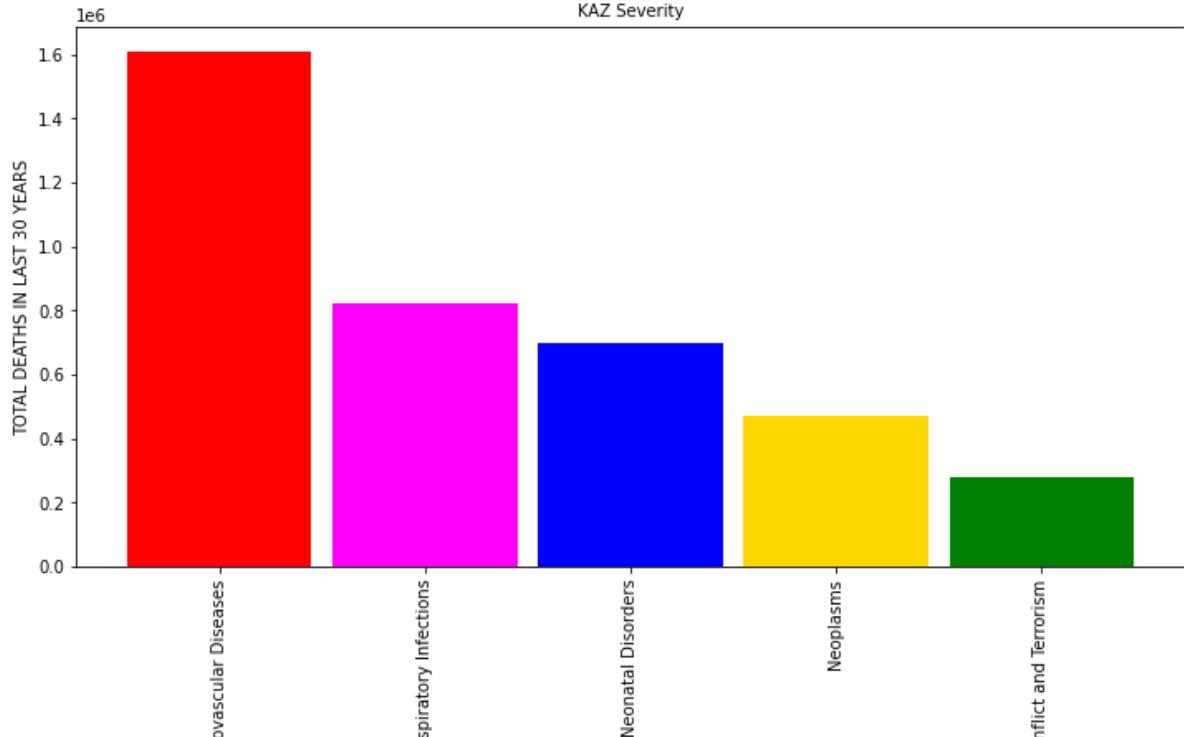


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

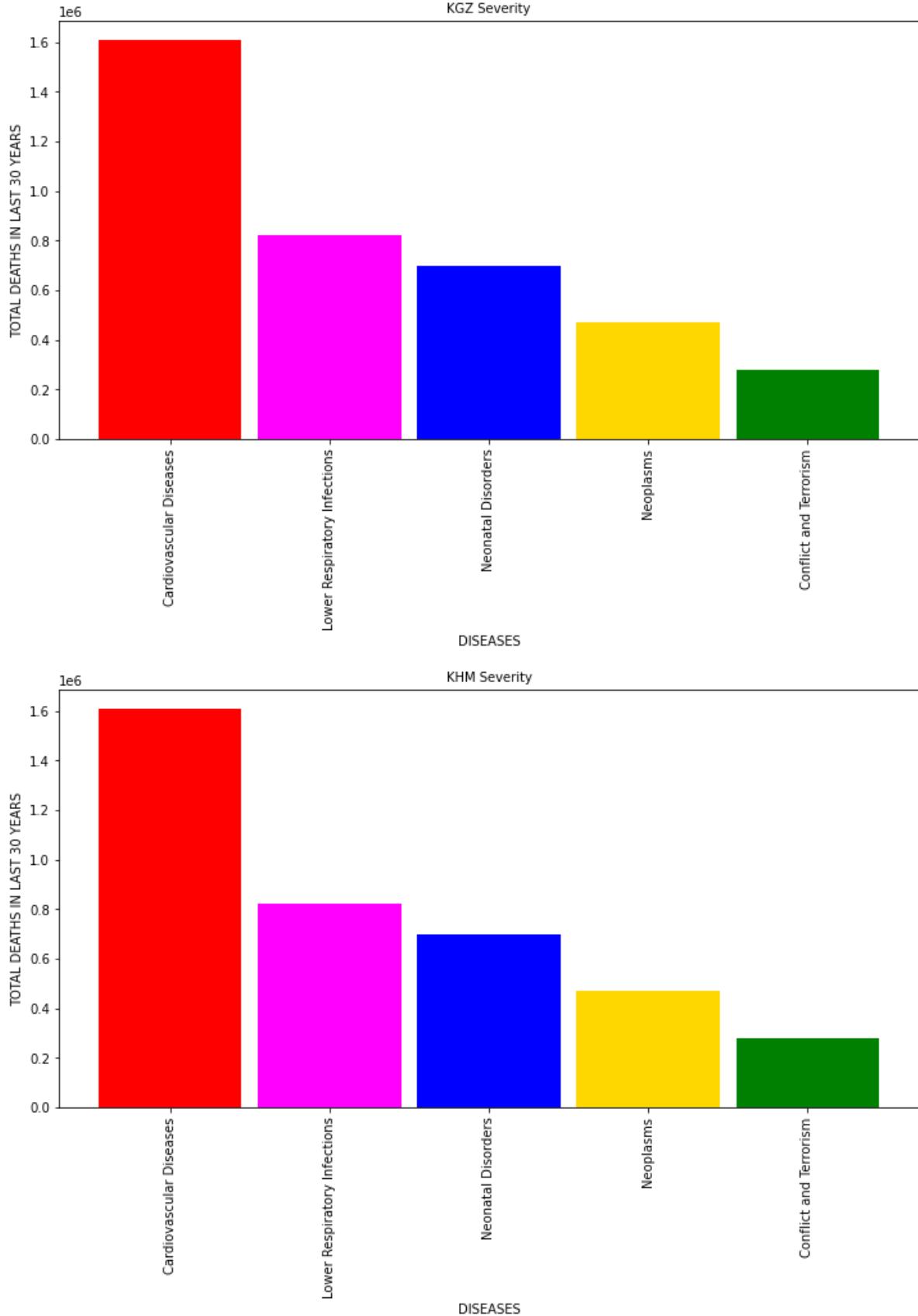
Cause Of Death



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

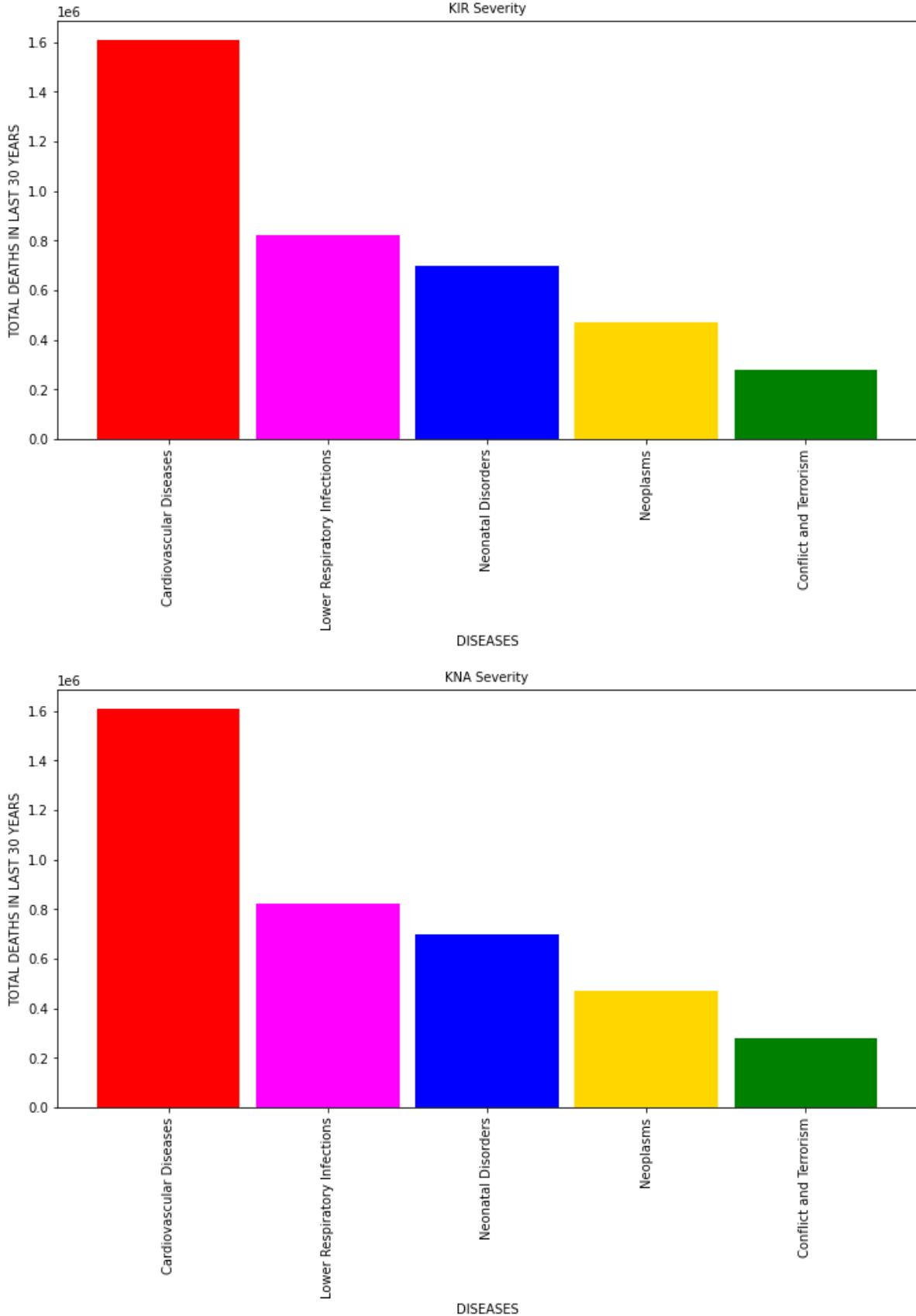


Cause Of Death



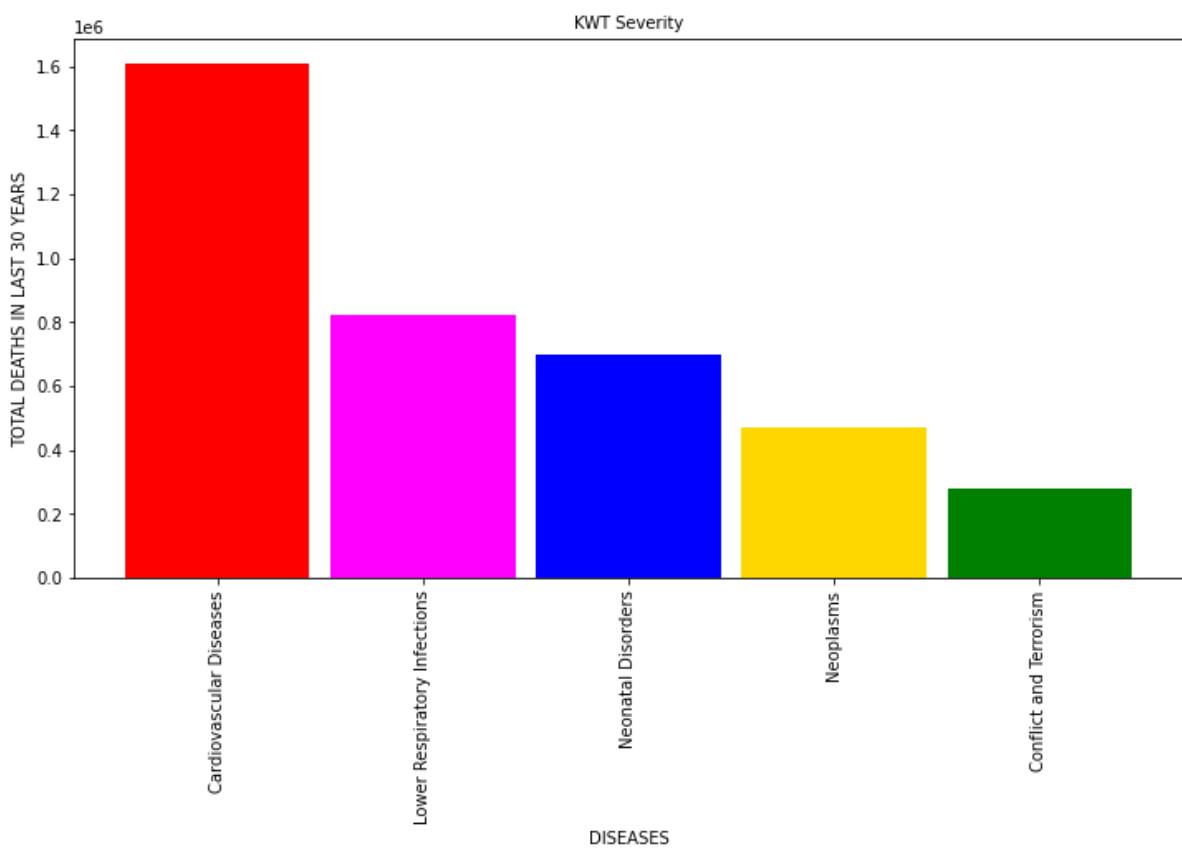
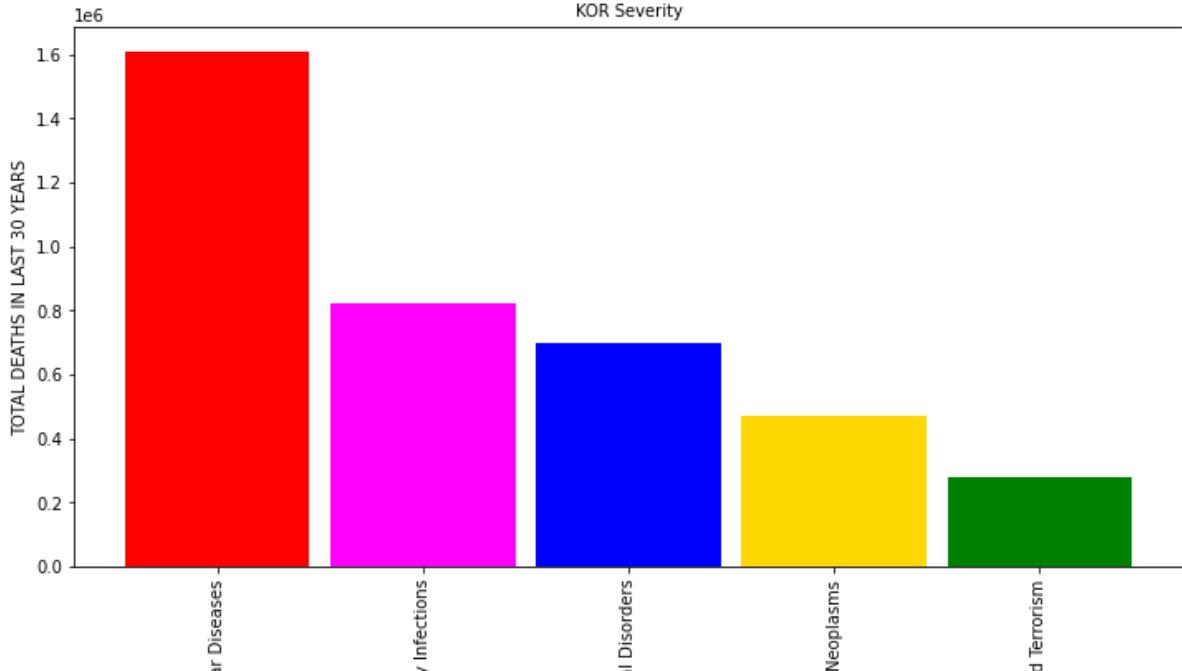
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



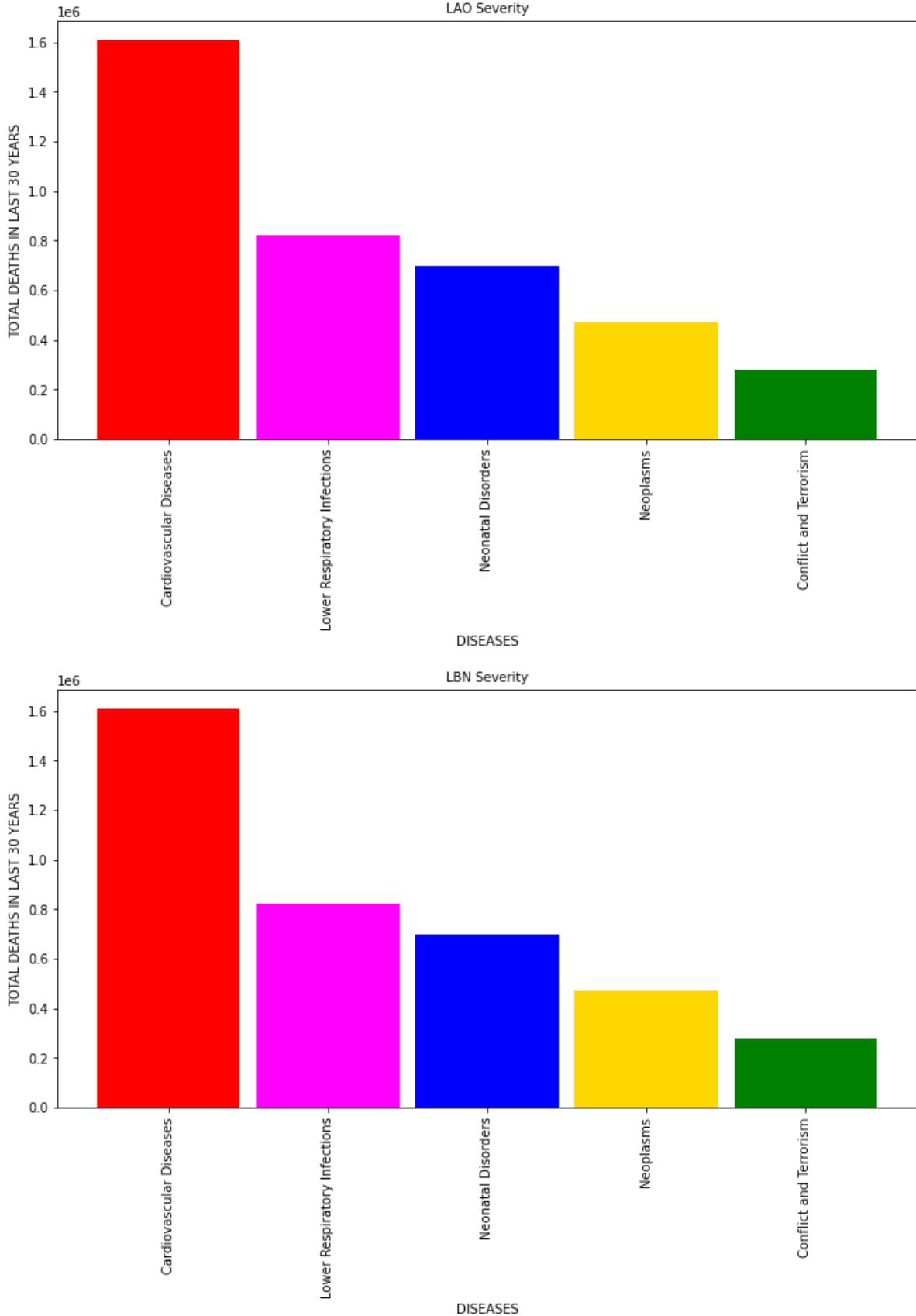
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death

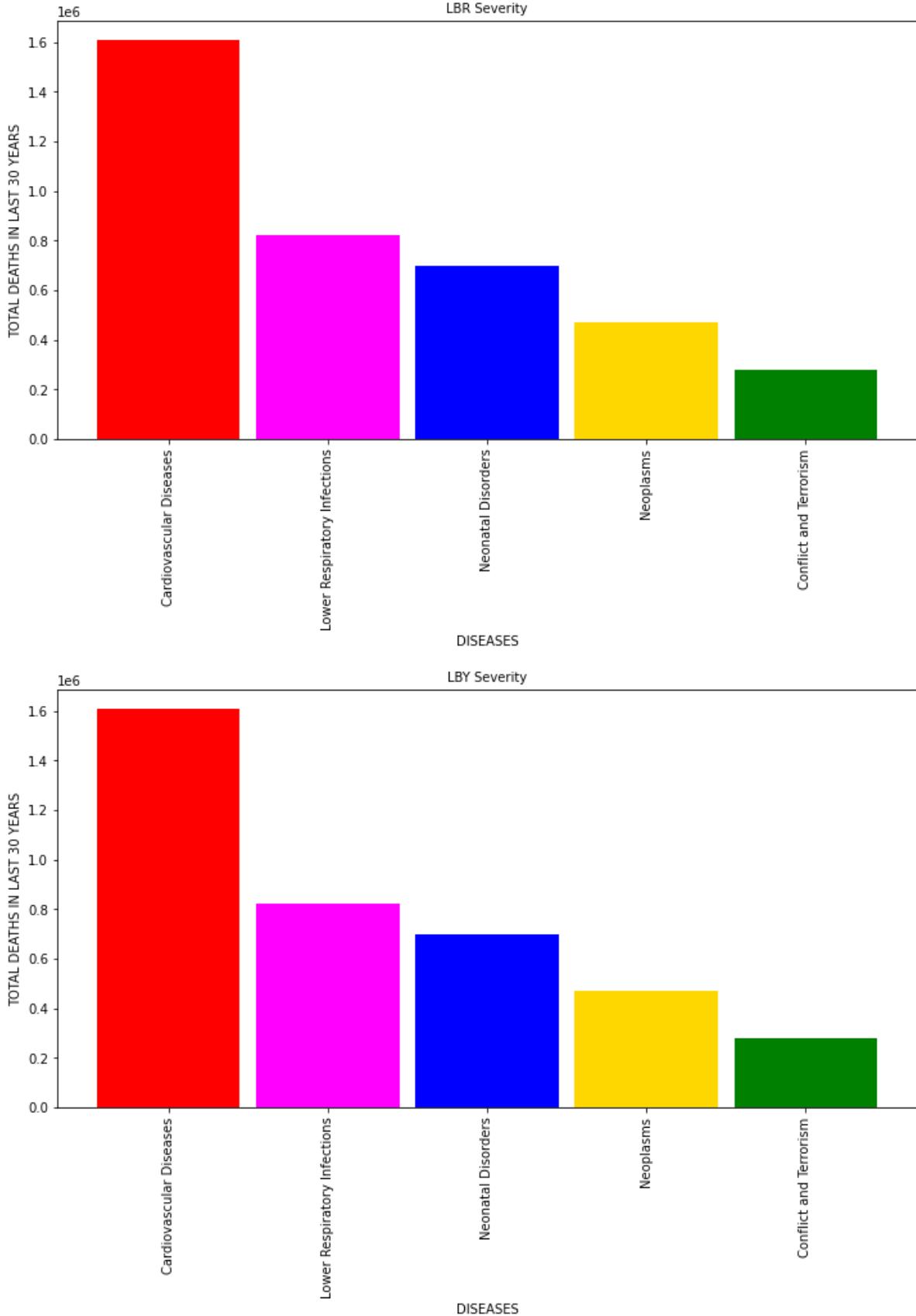


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

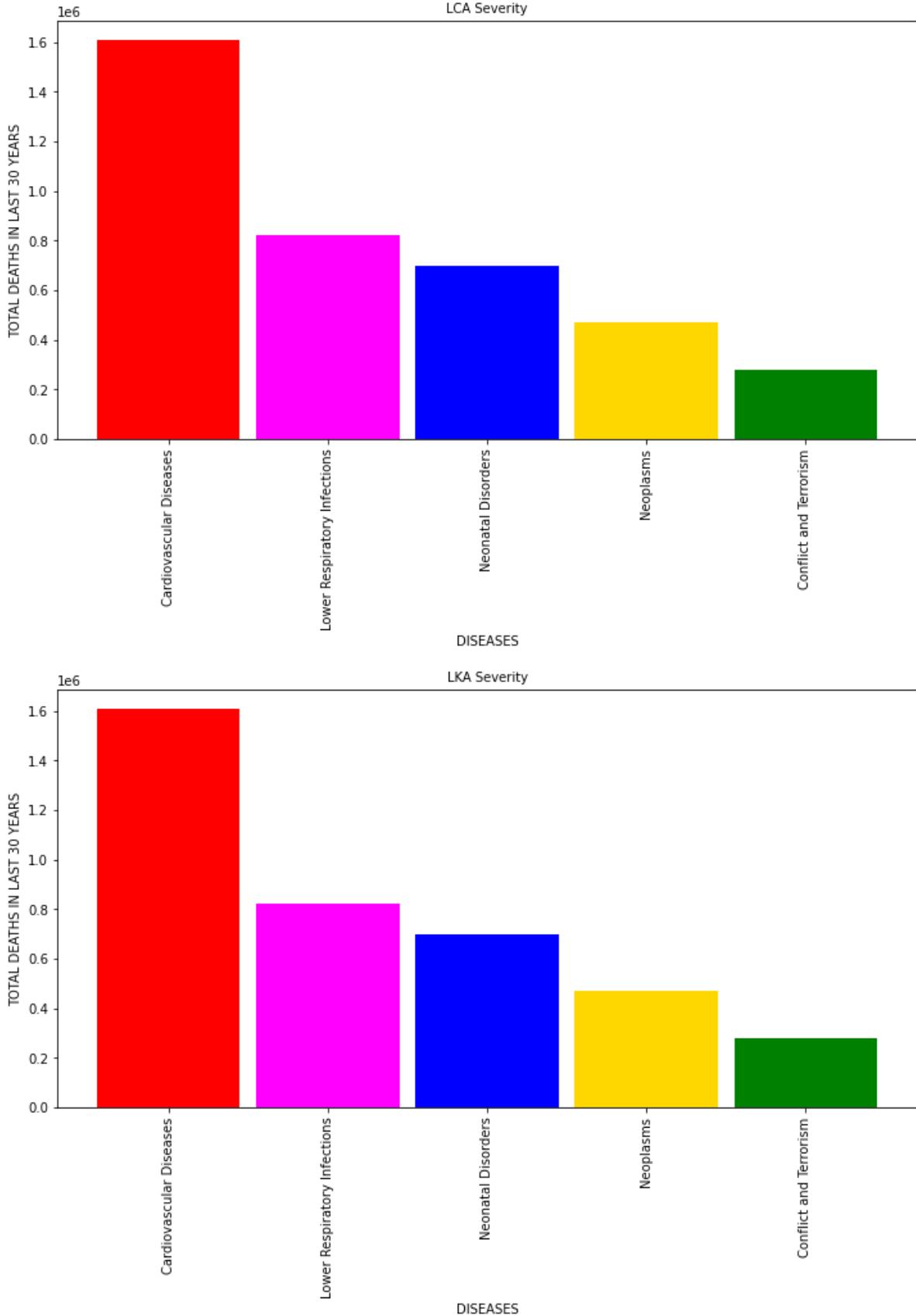
Cause Of Death



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

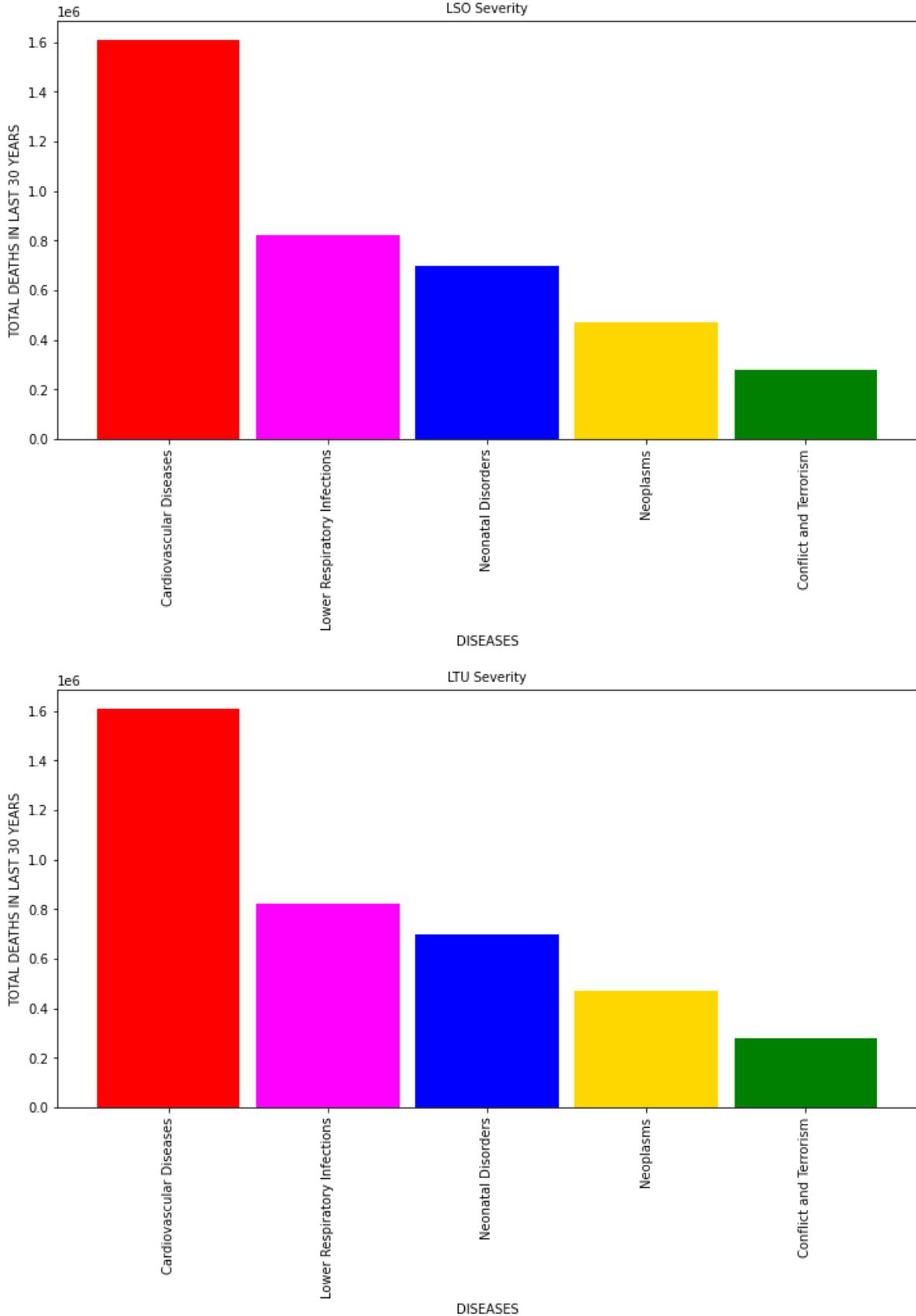


Cause Of Death

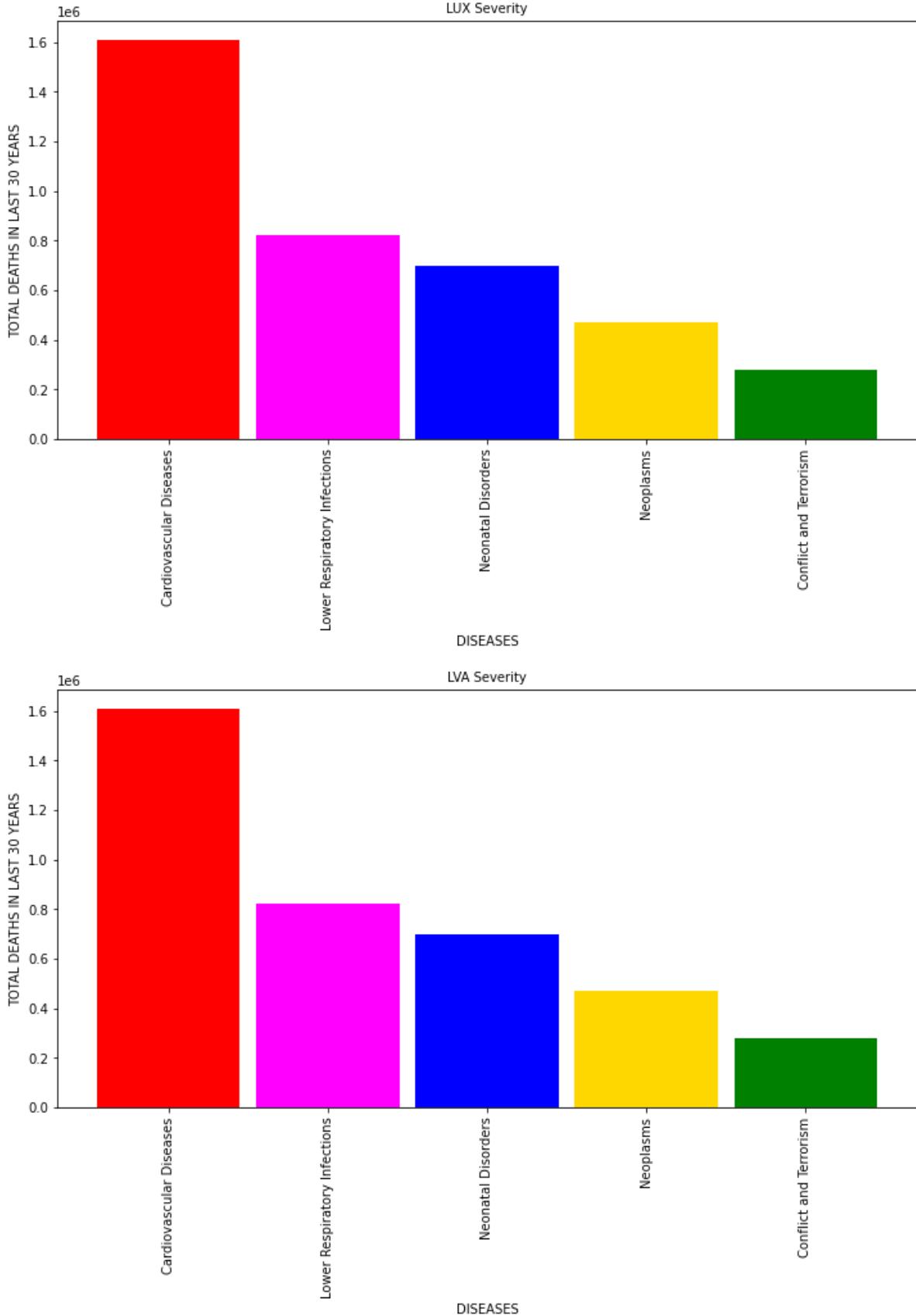


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

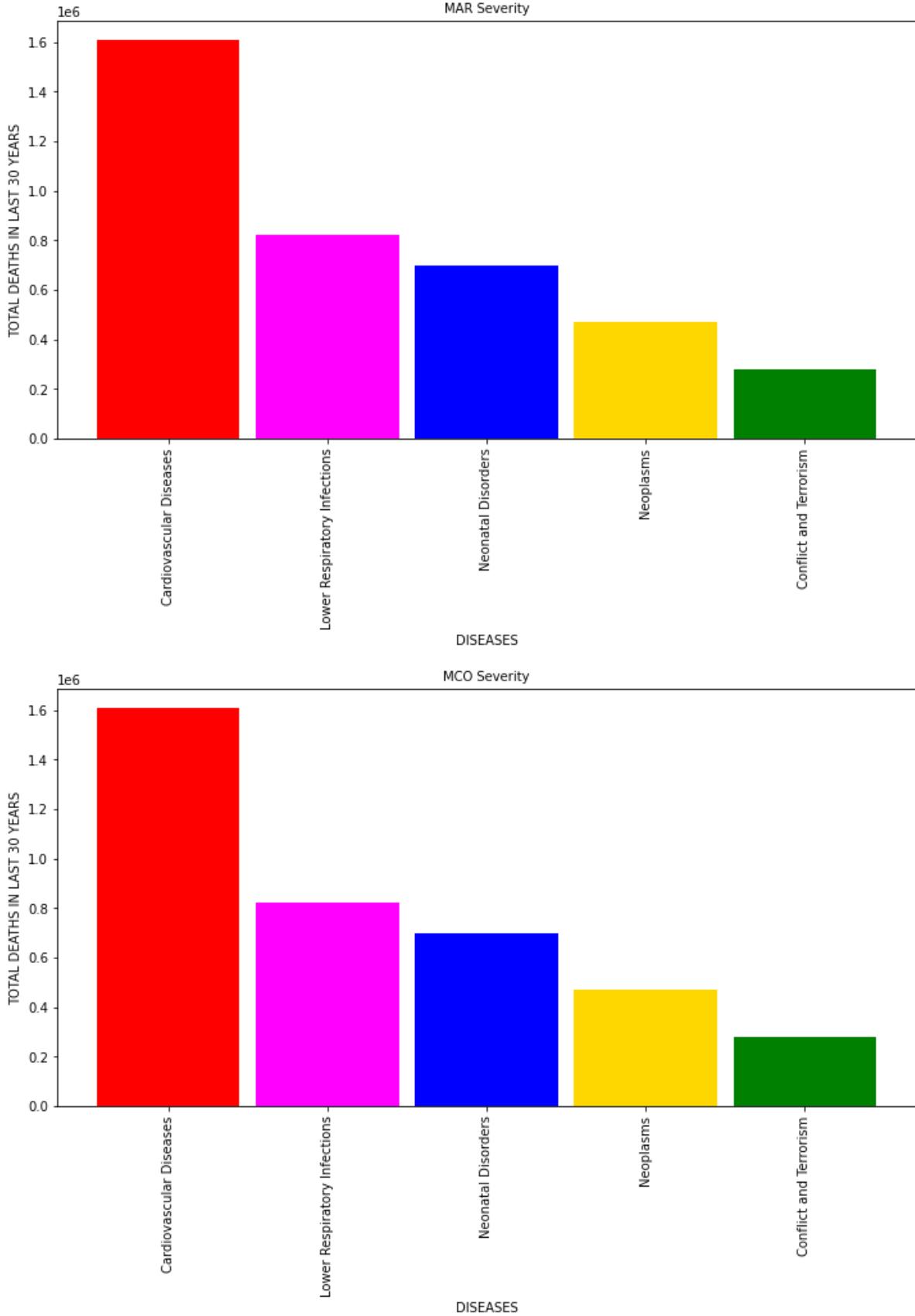
Cause Of Death



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

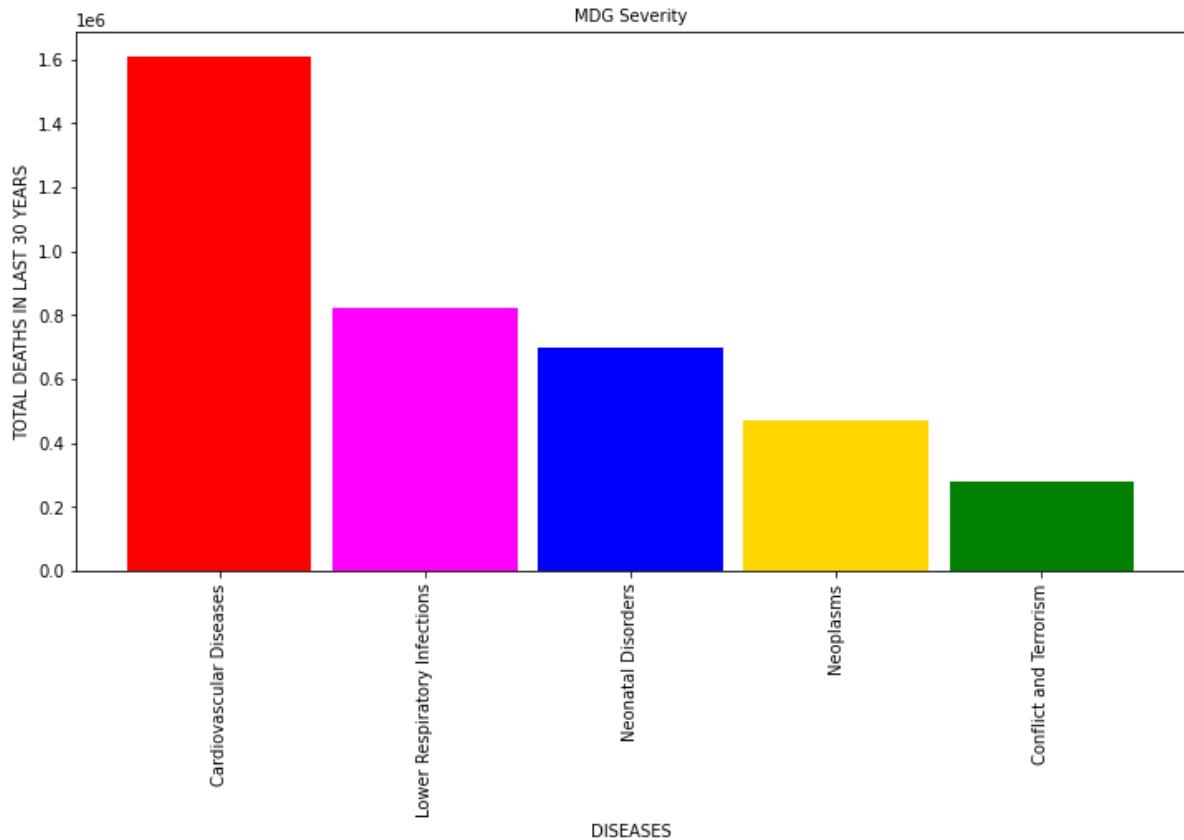
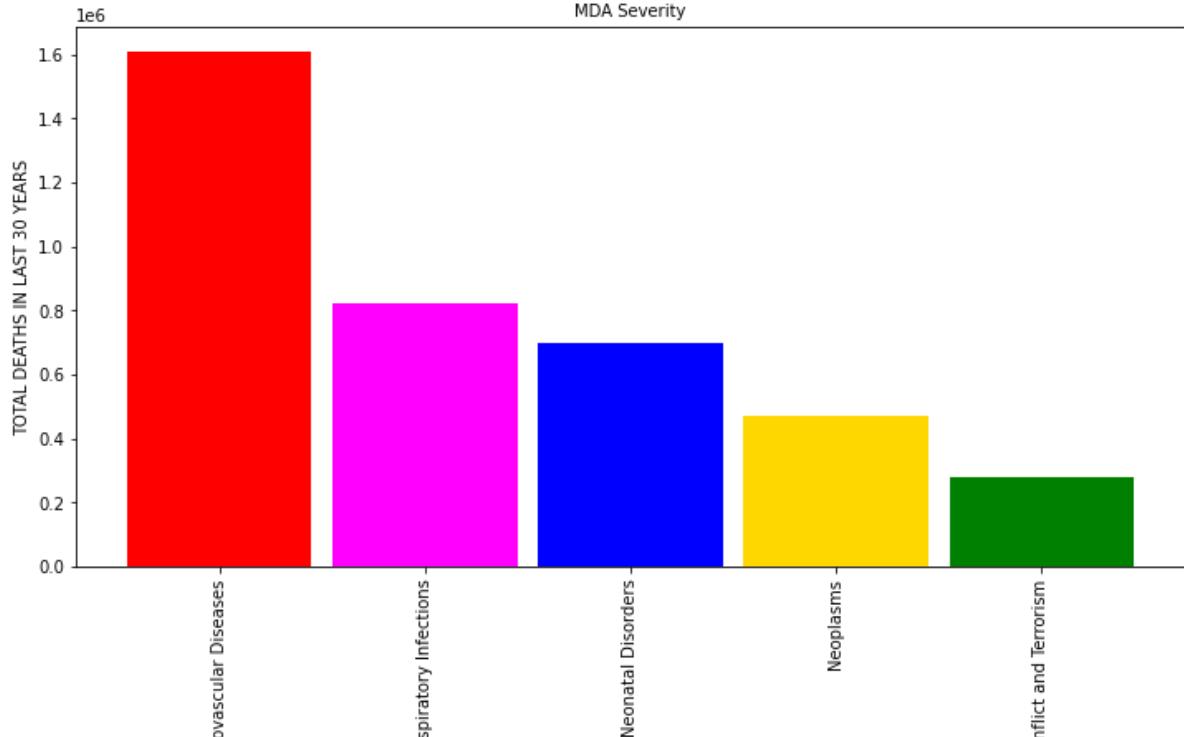


Cause Of Death



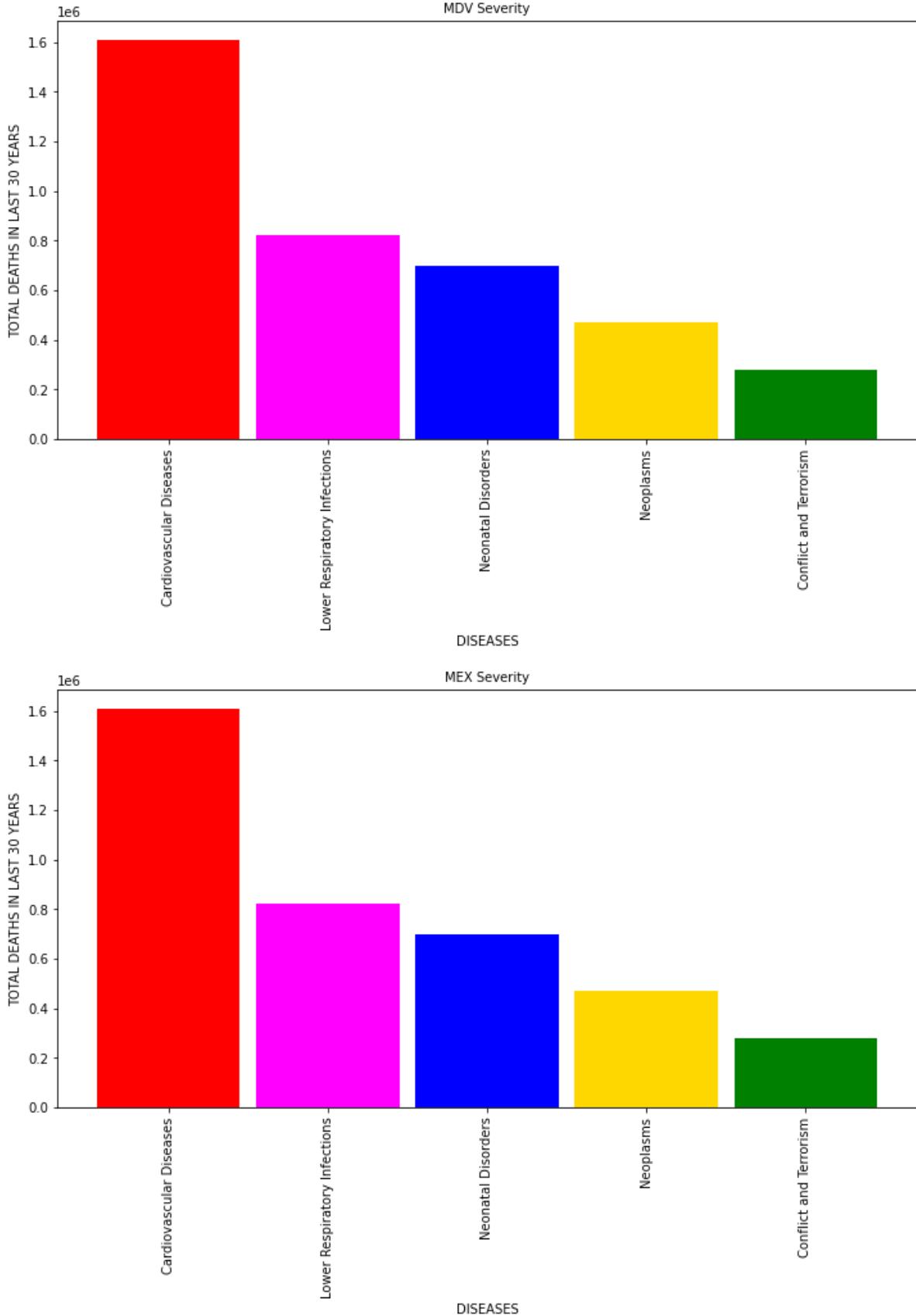
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death

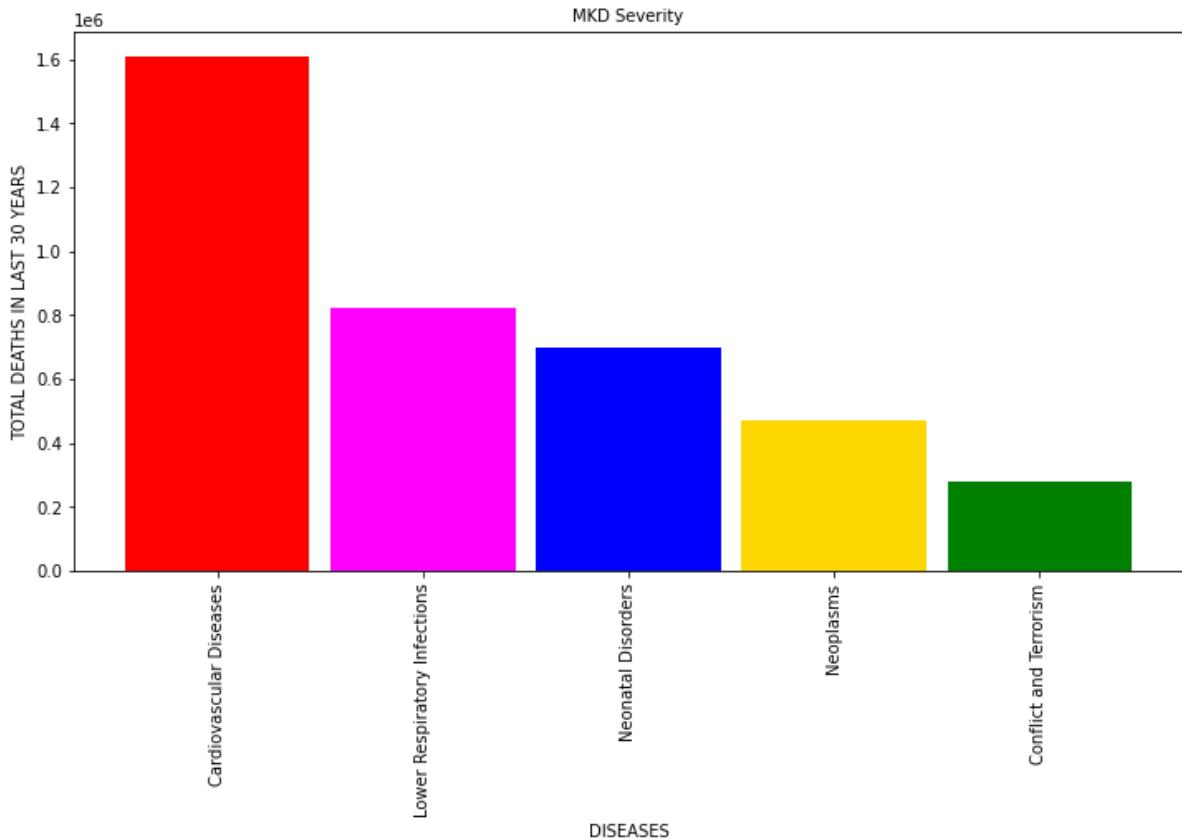
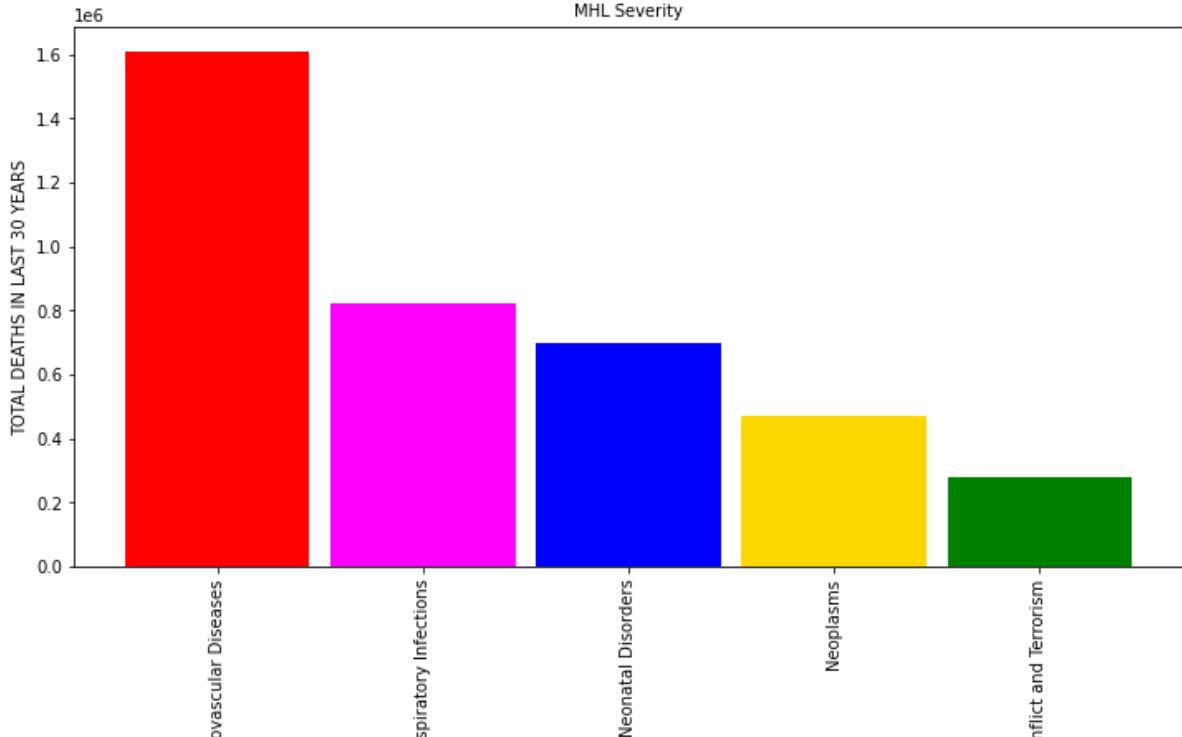


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

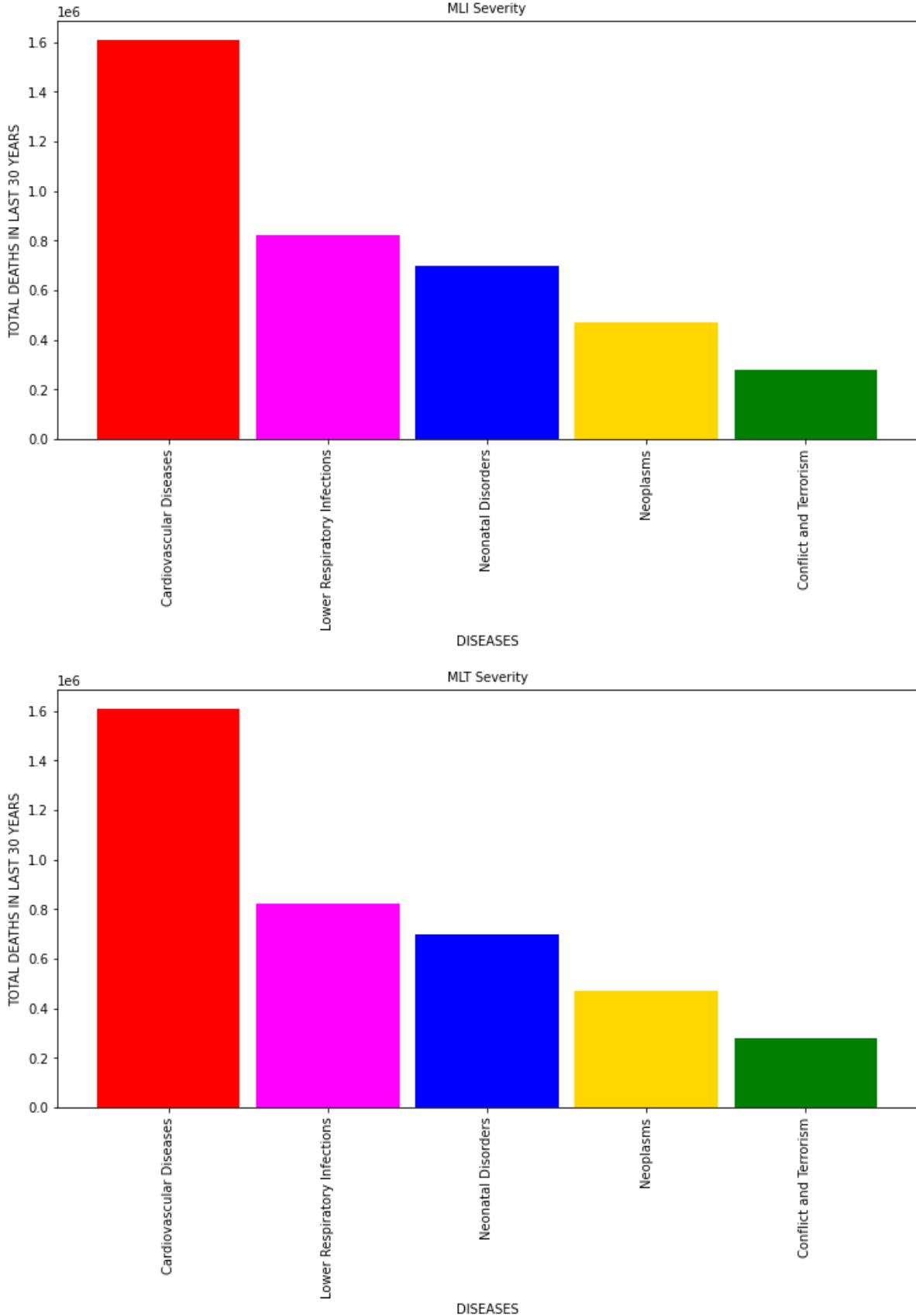
Cause Of Death



Cause Of Death

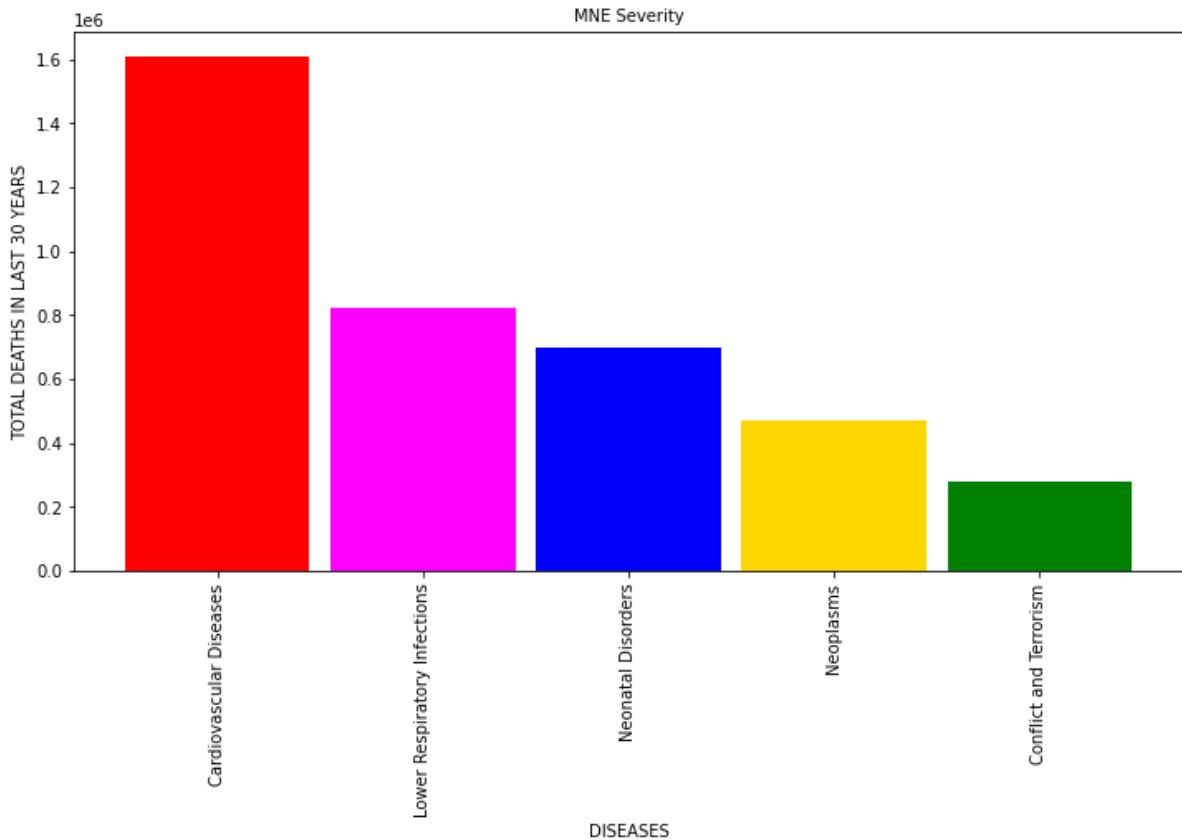
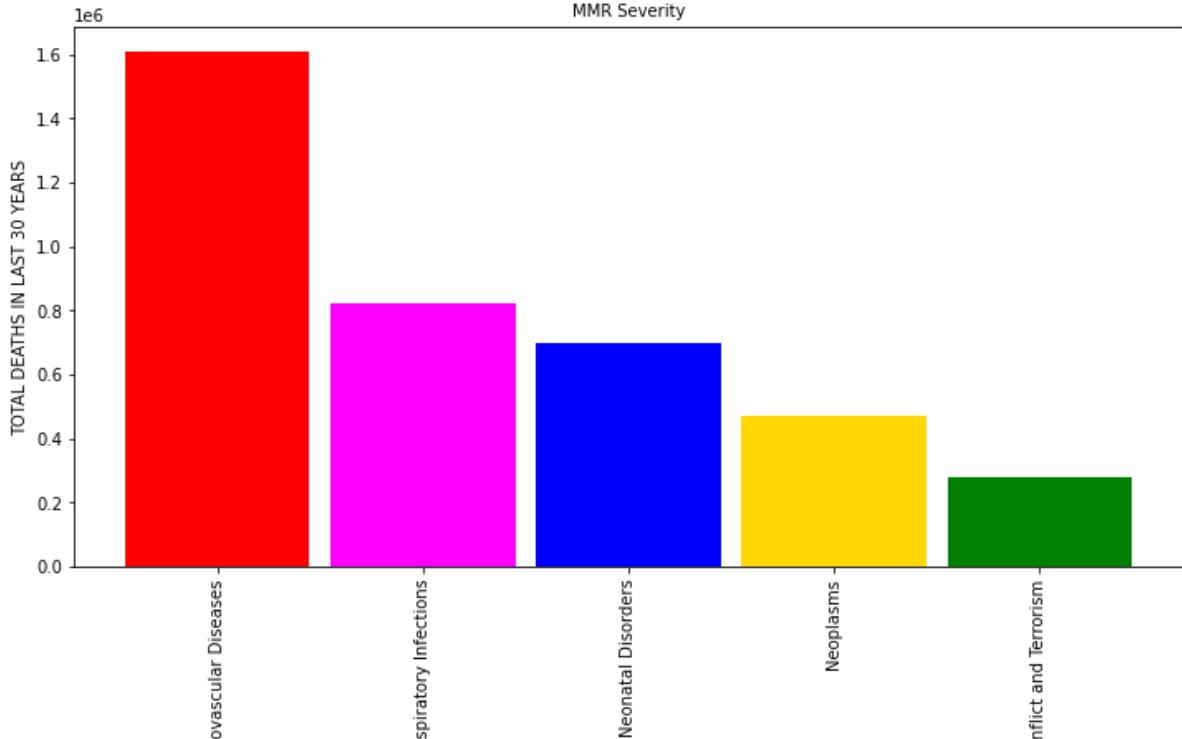


Cause Of Death



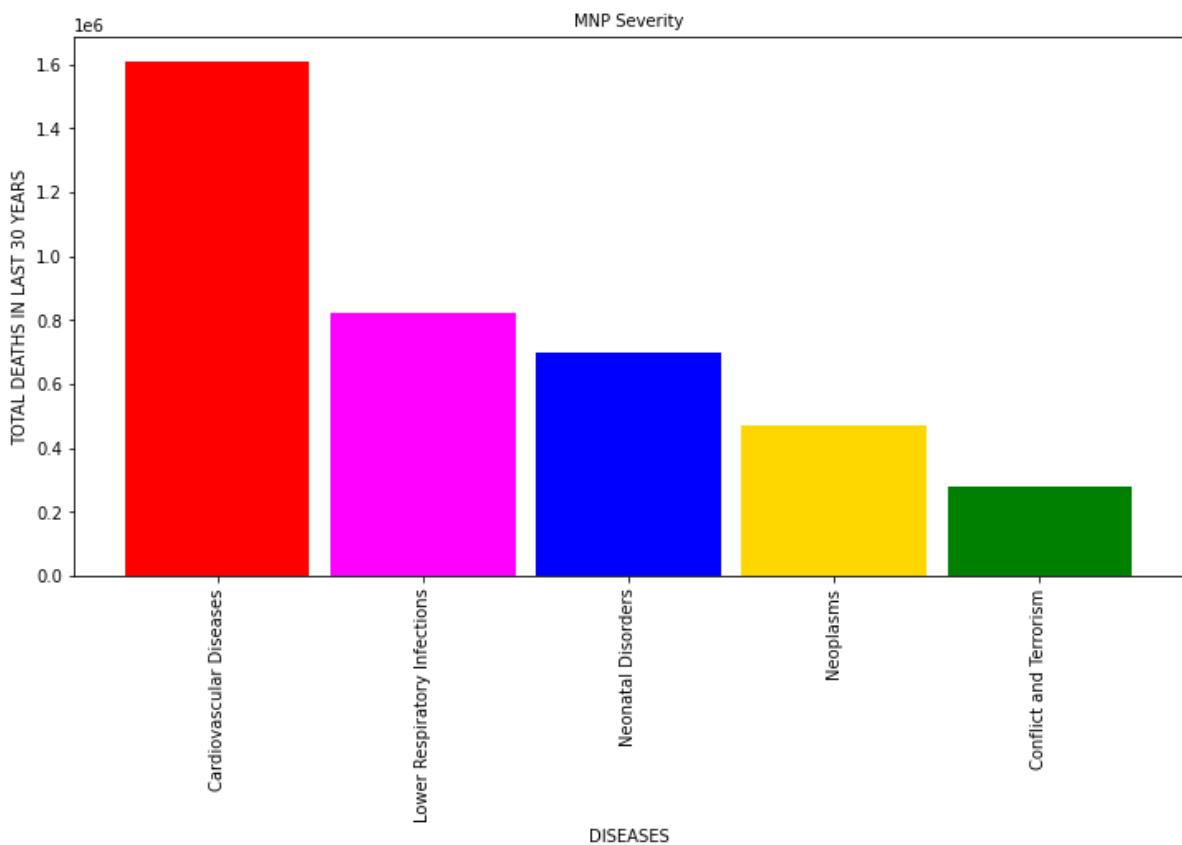
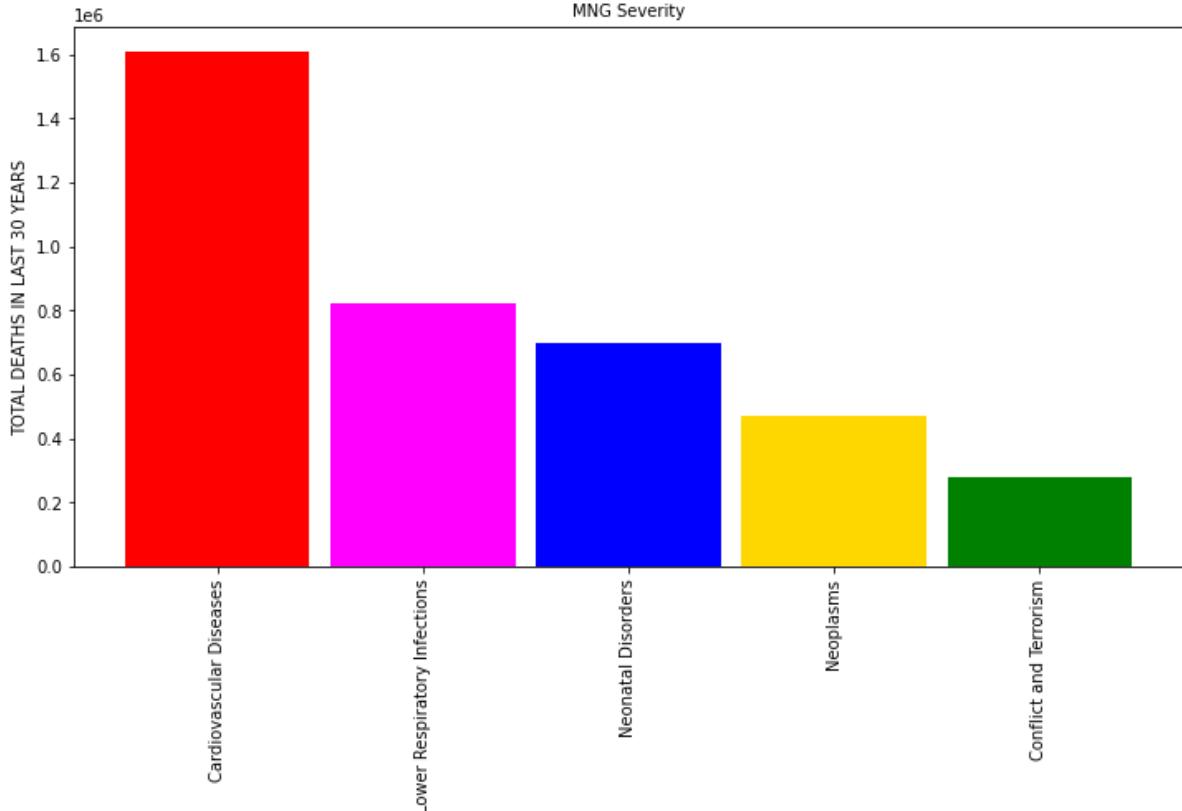
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death

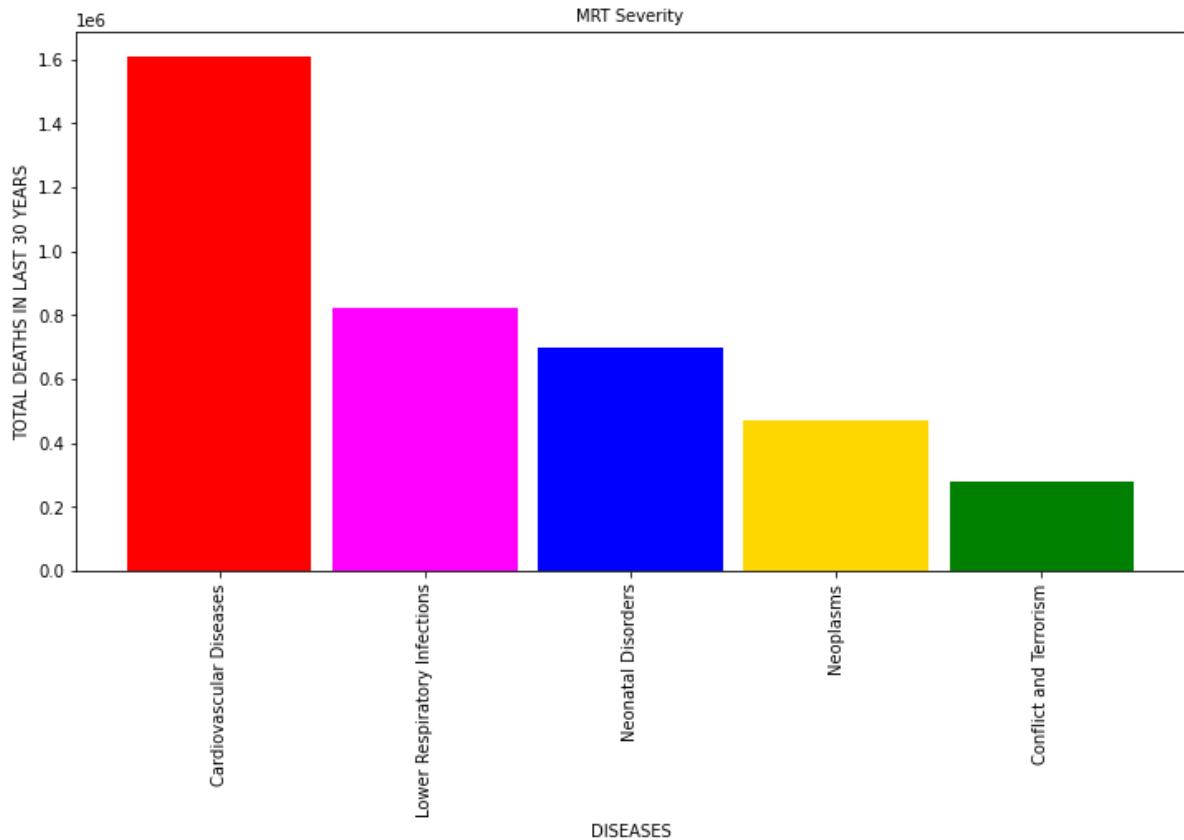
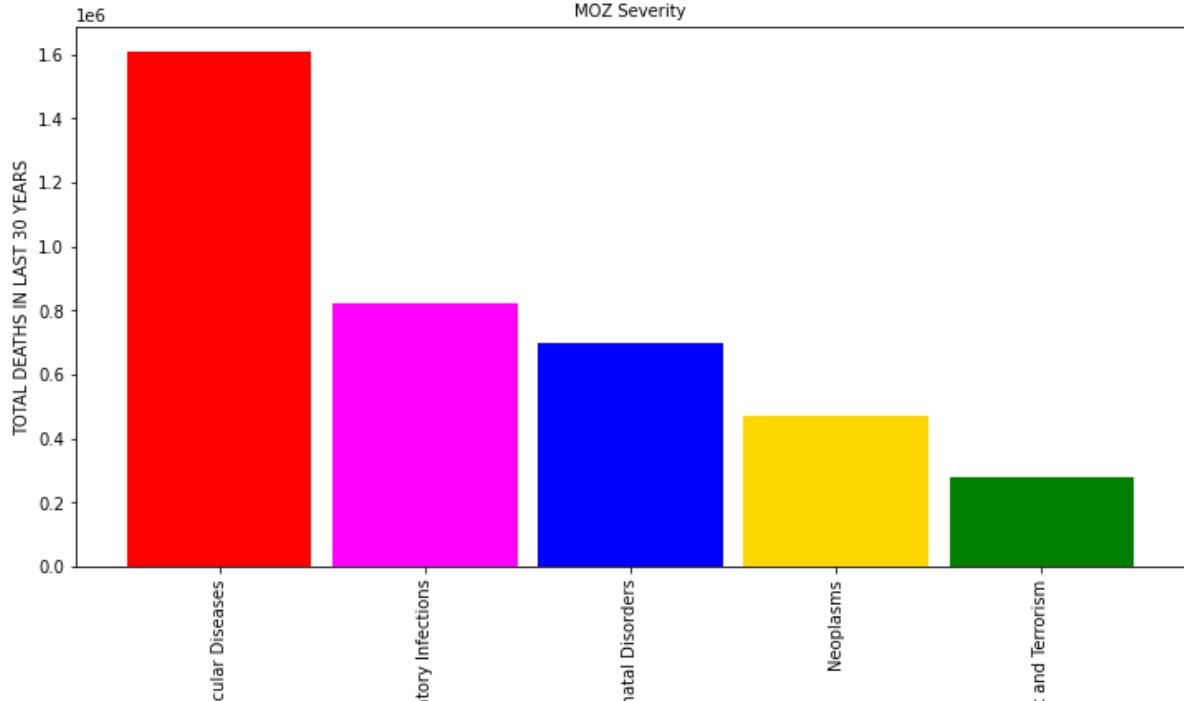


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death

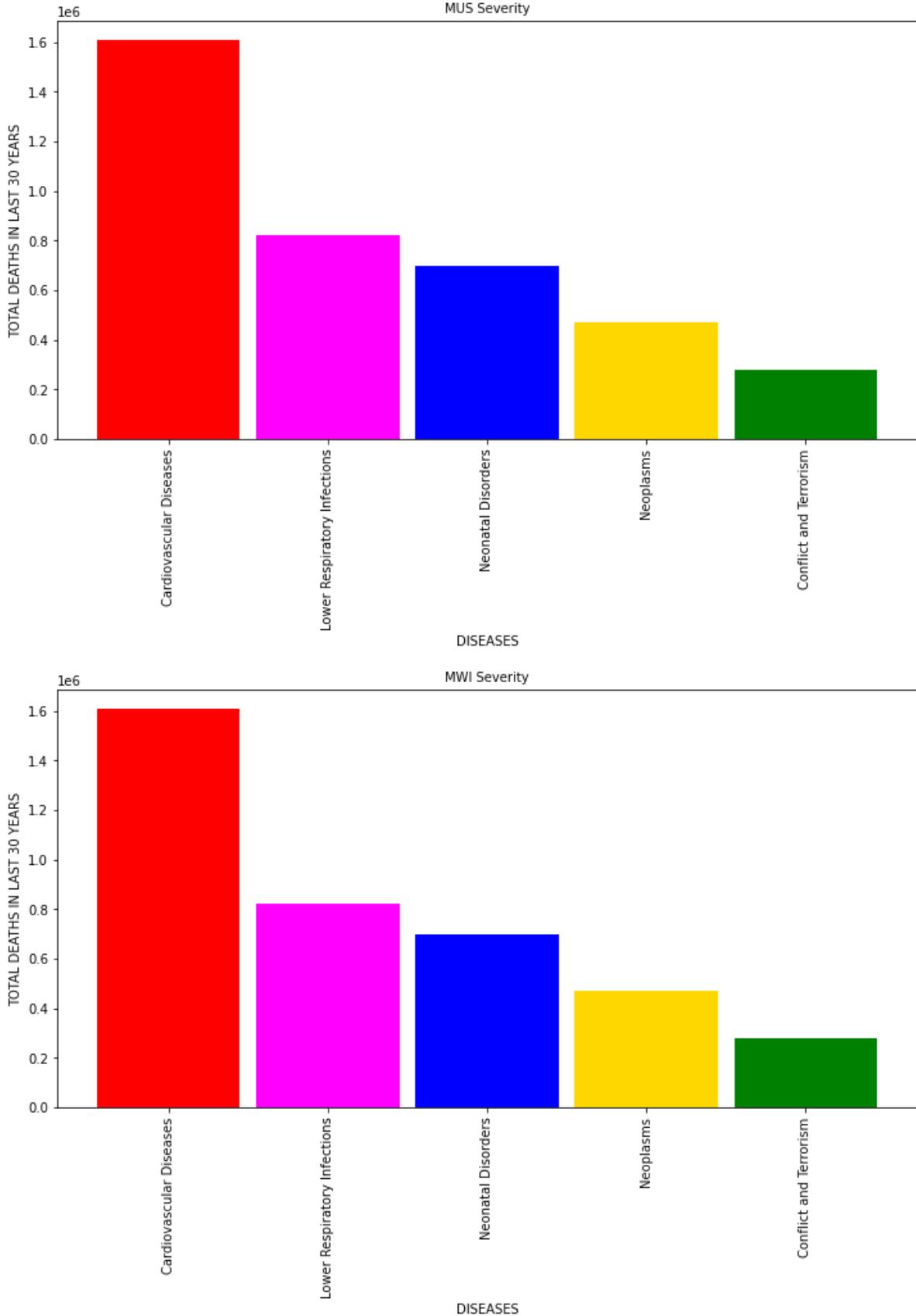


Cause Of Death



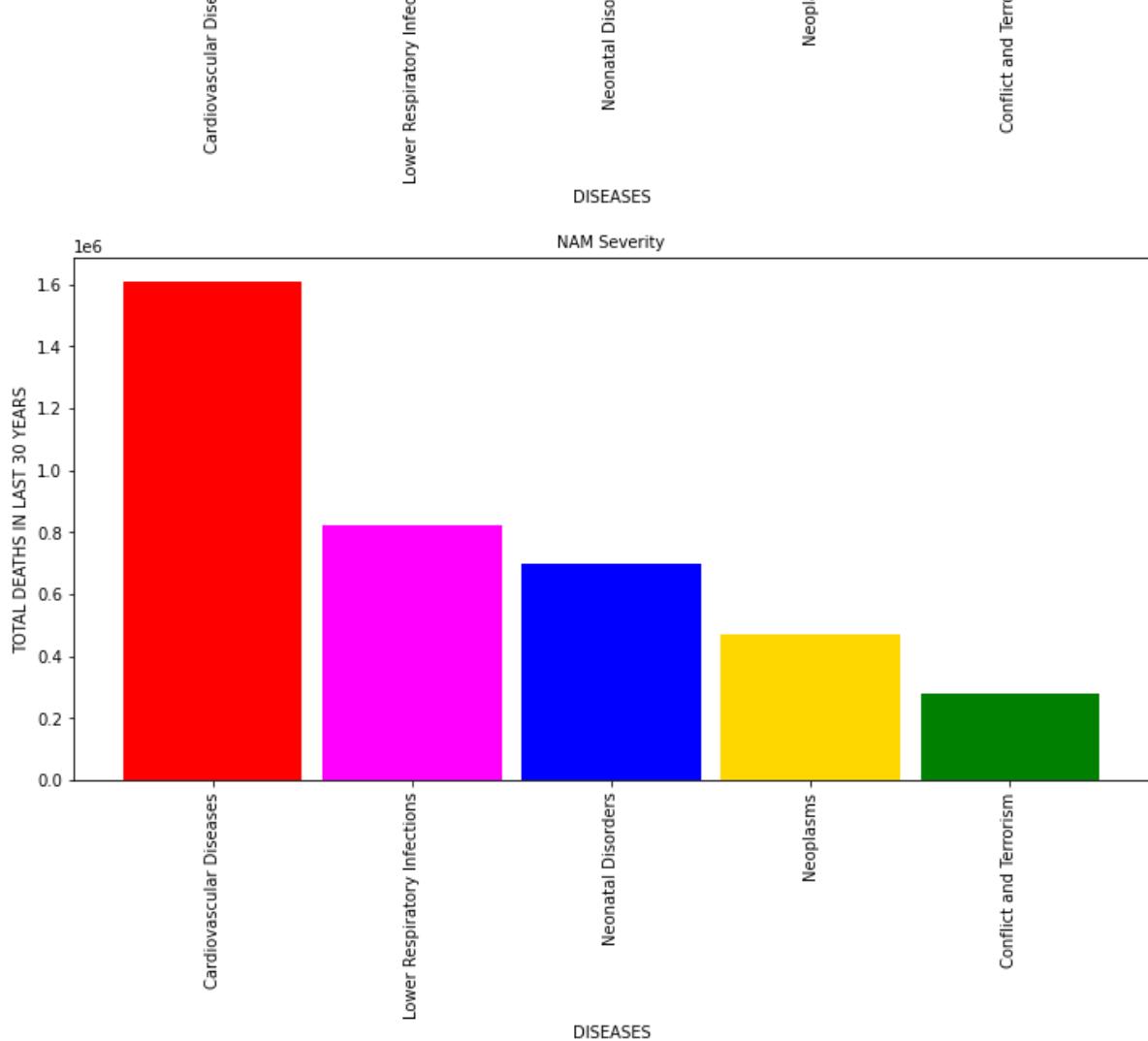
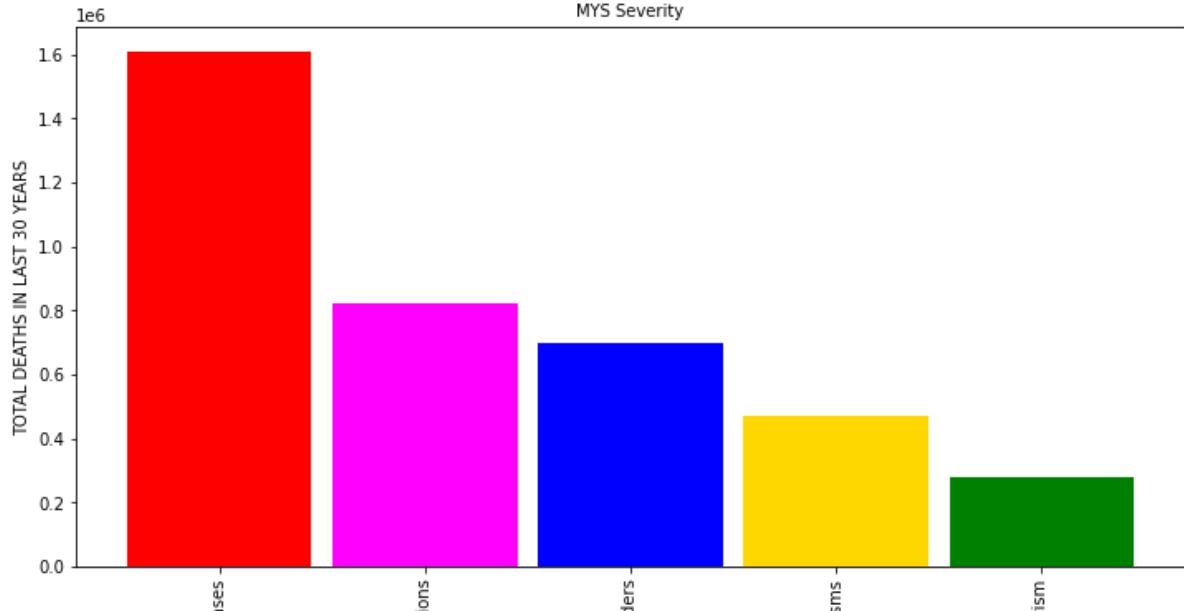
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



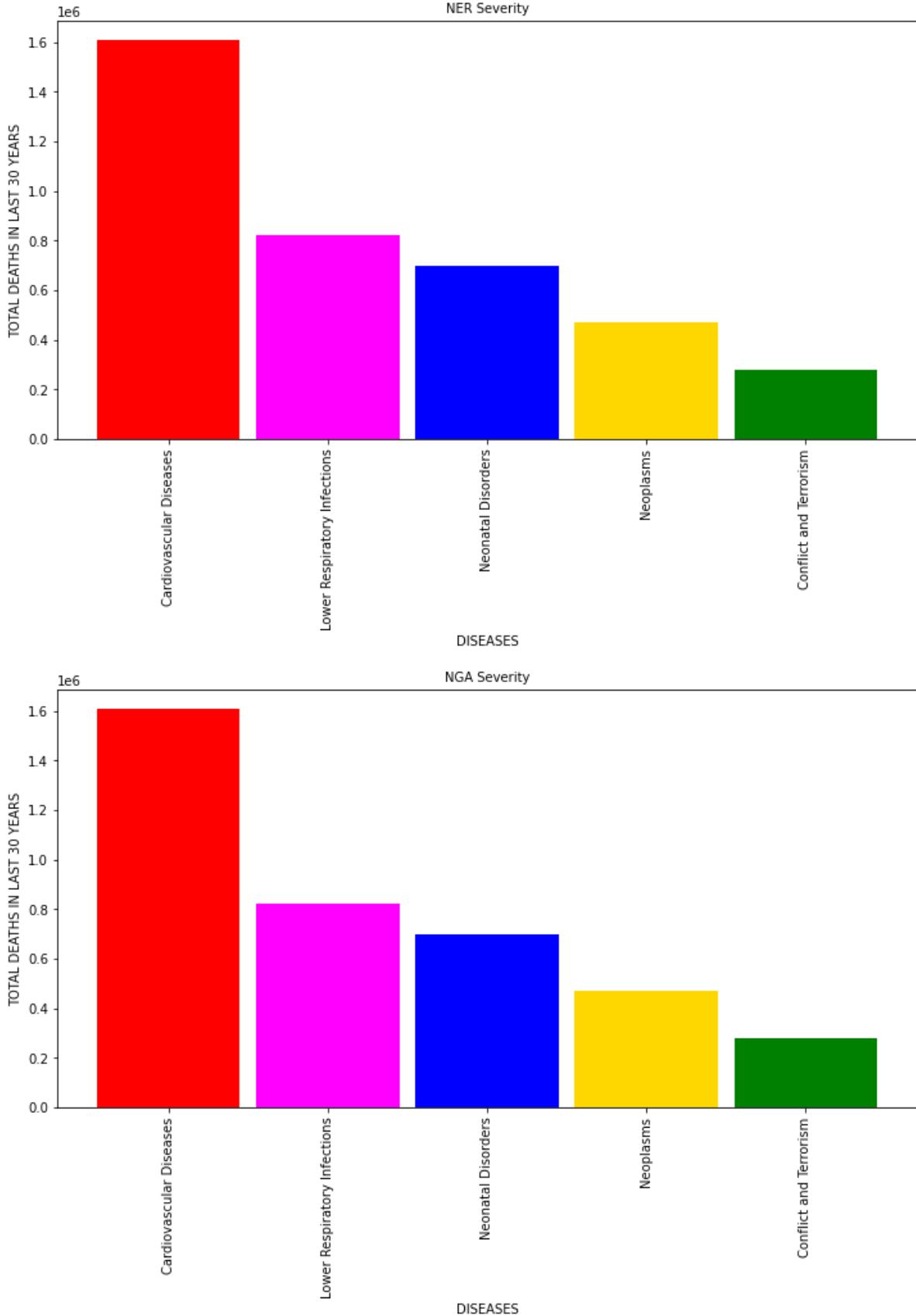
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



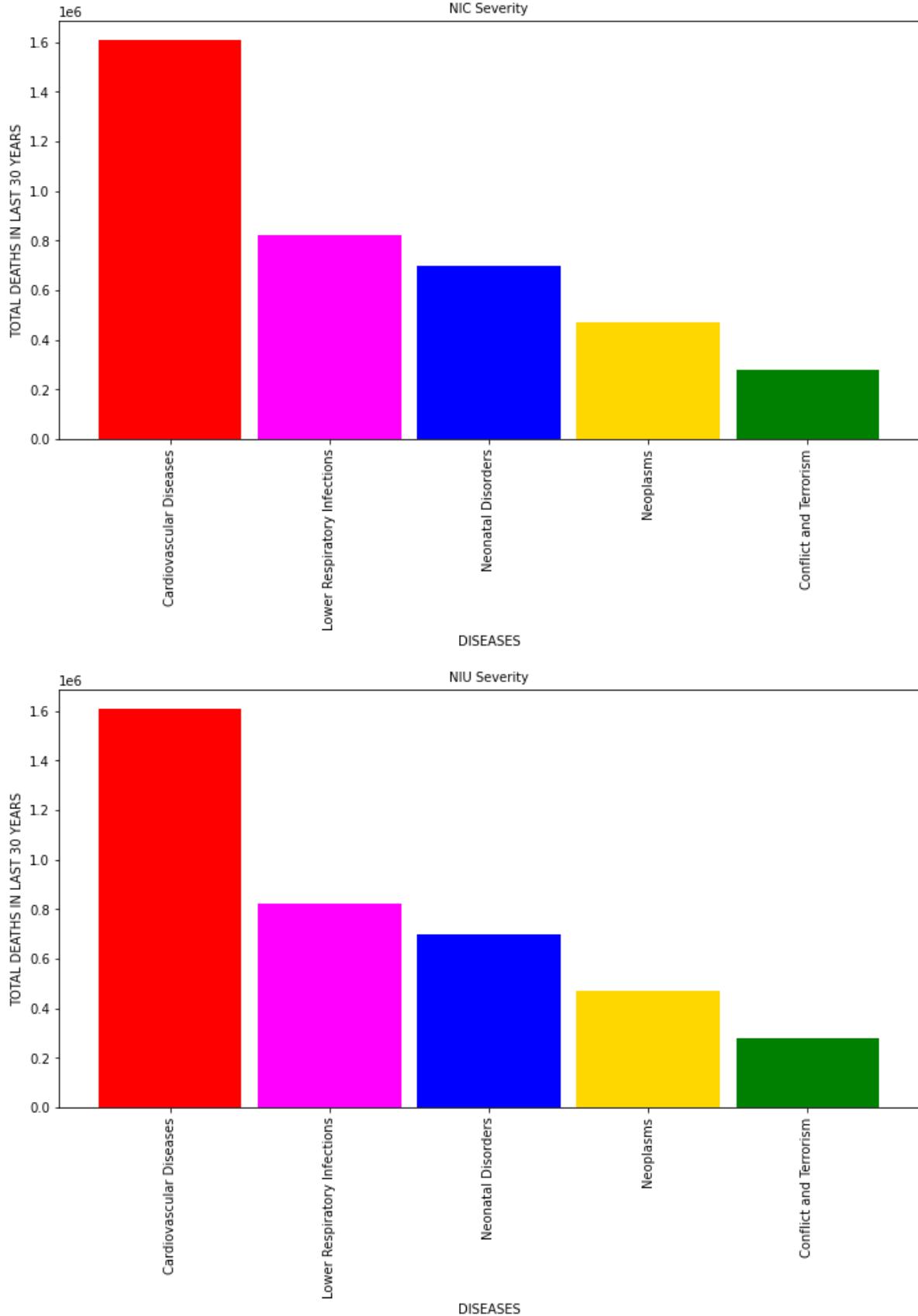
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



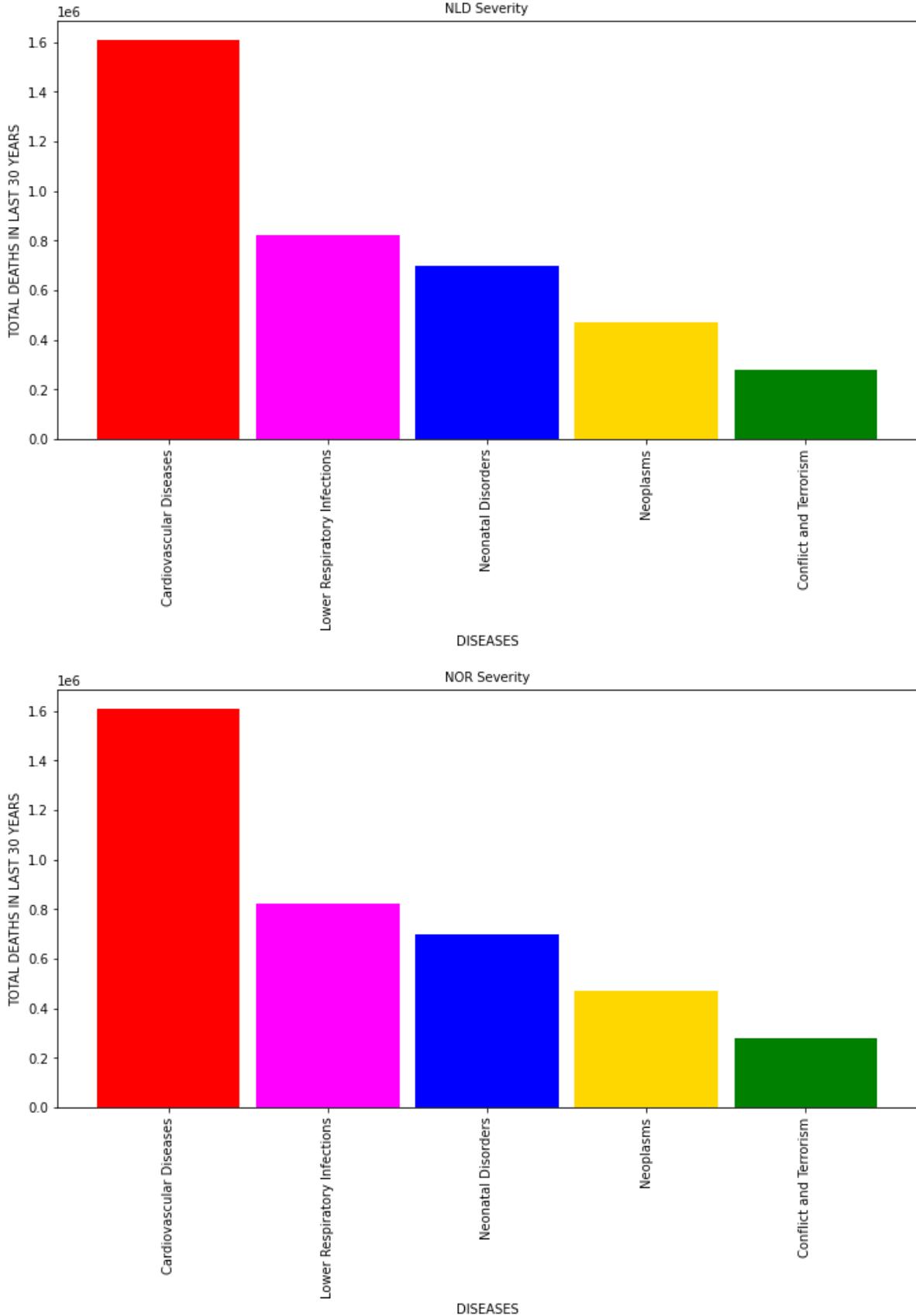
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



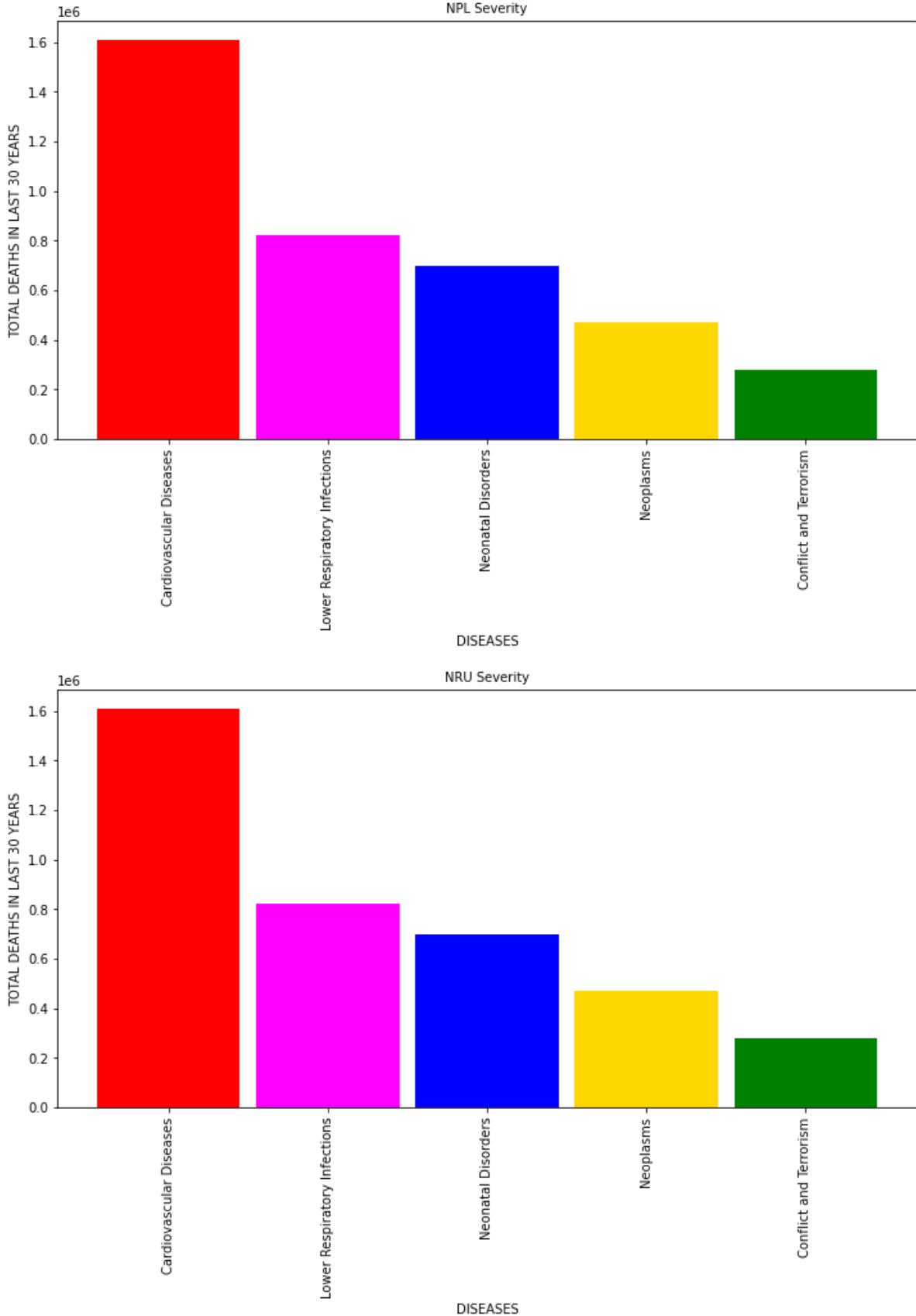
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



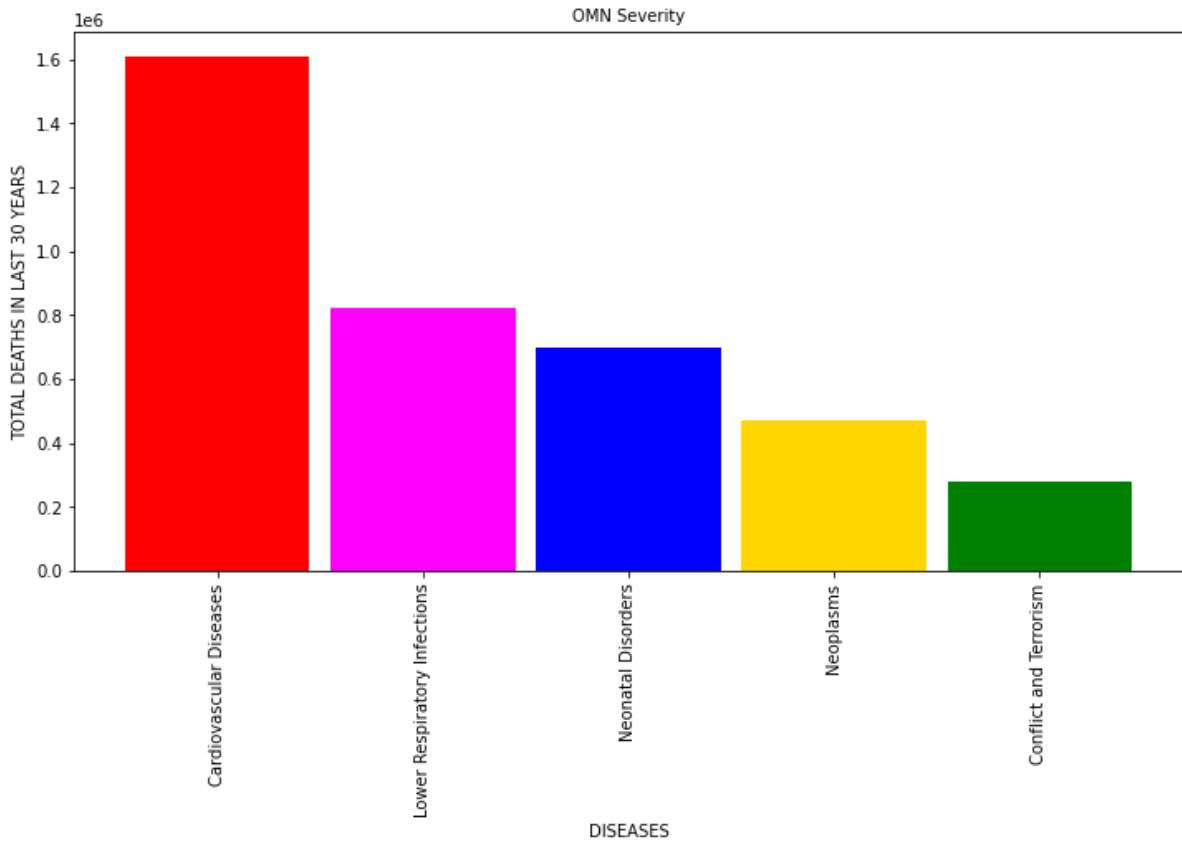
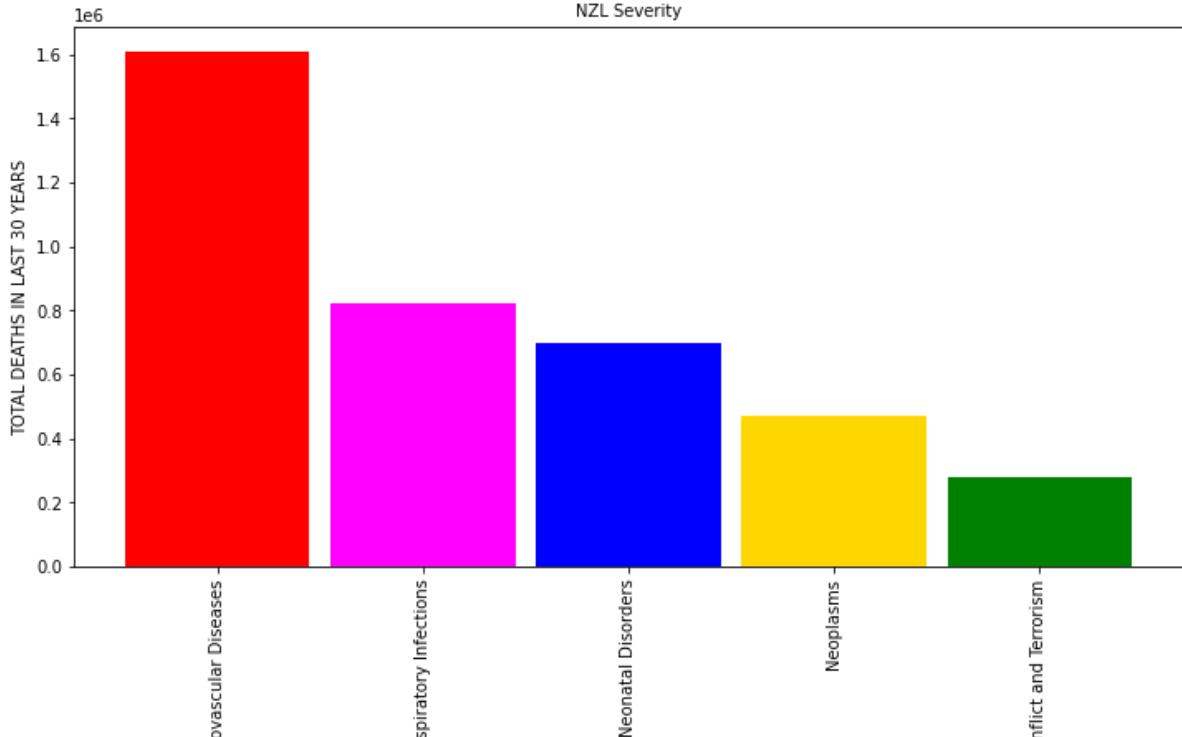
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death

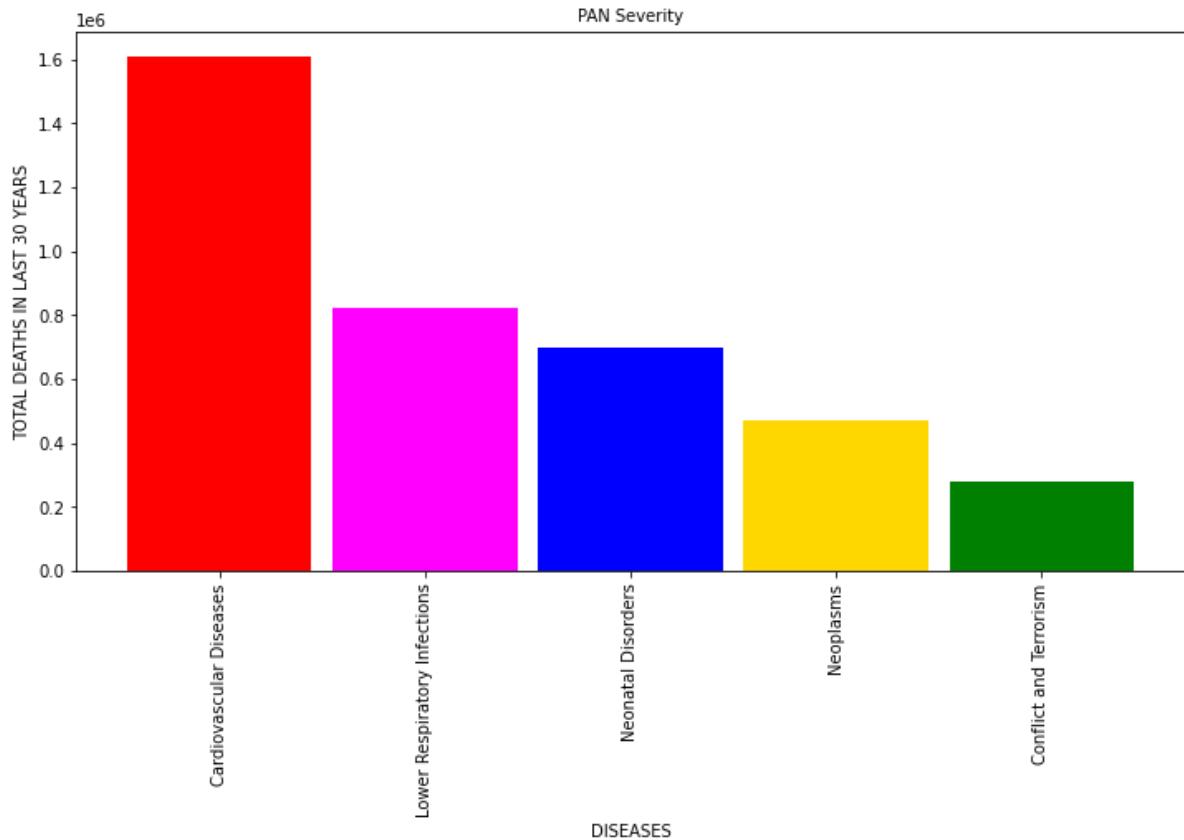
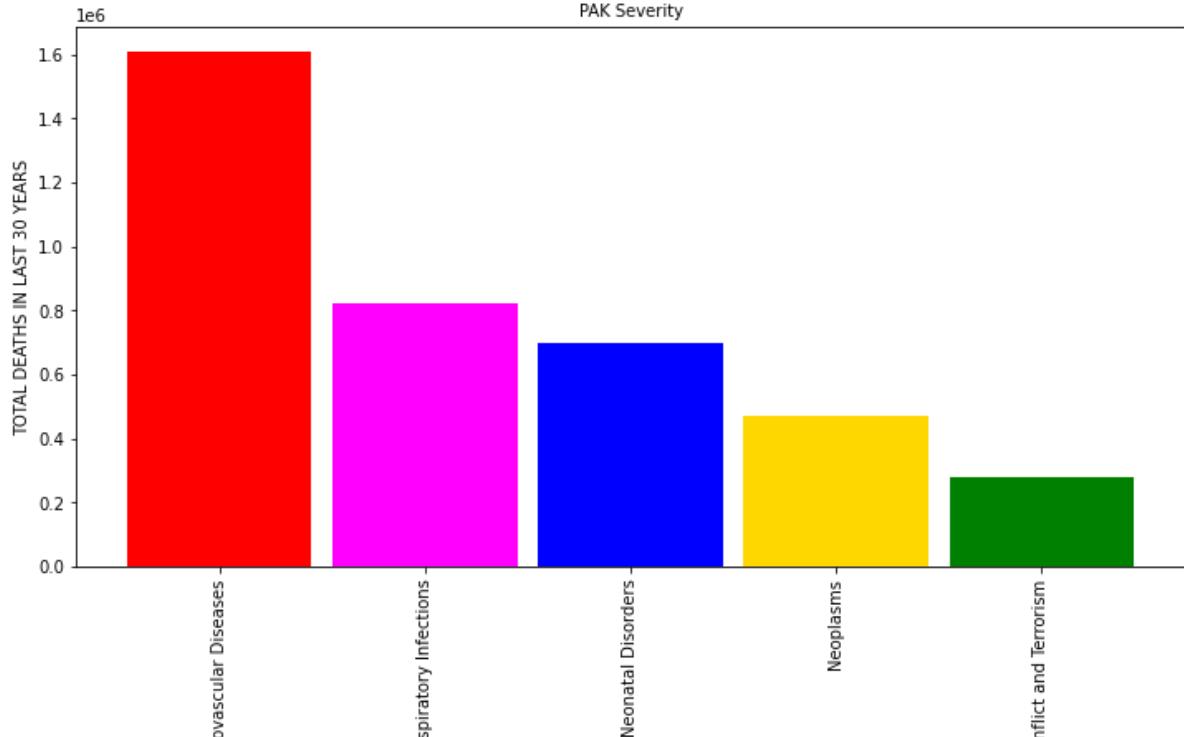


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

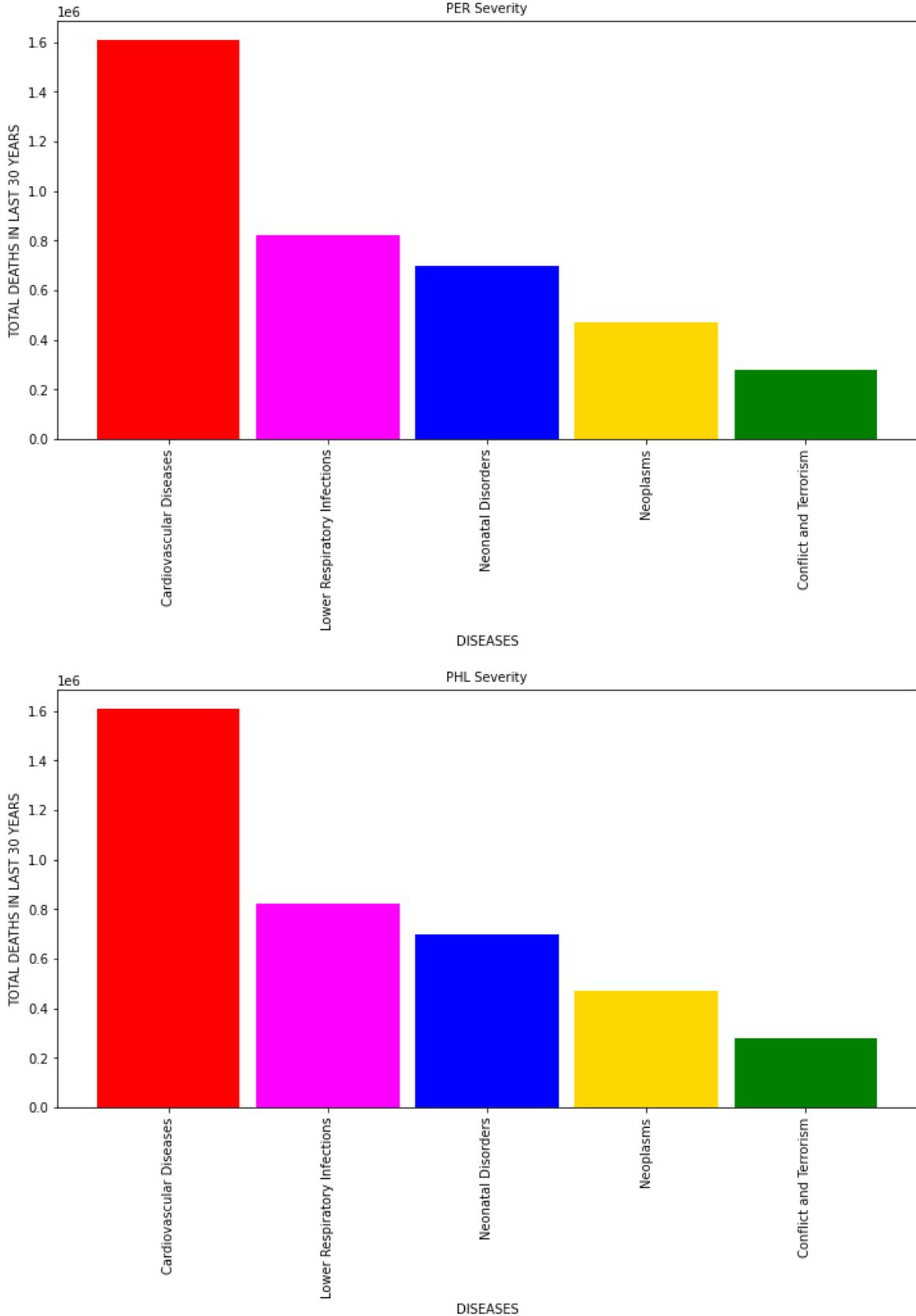
Cause Of Death



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

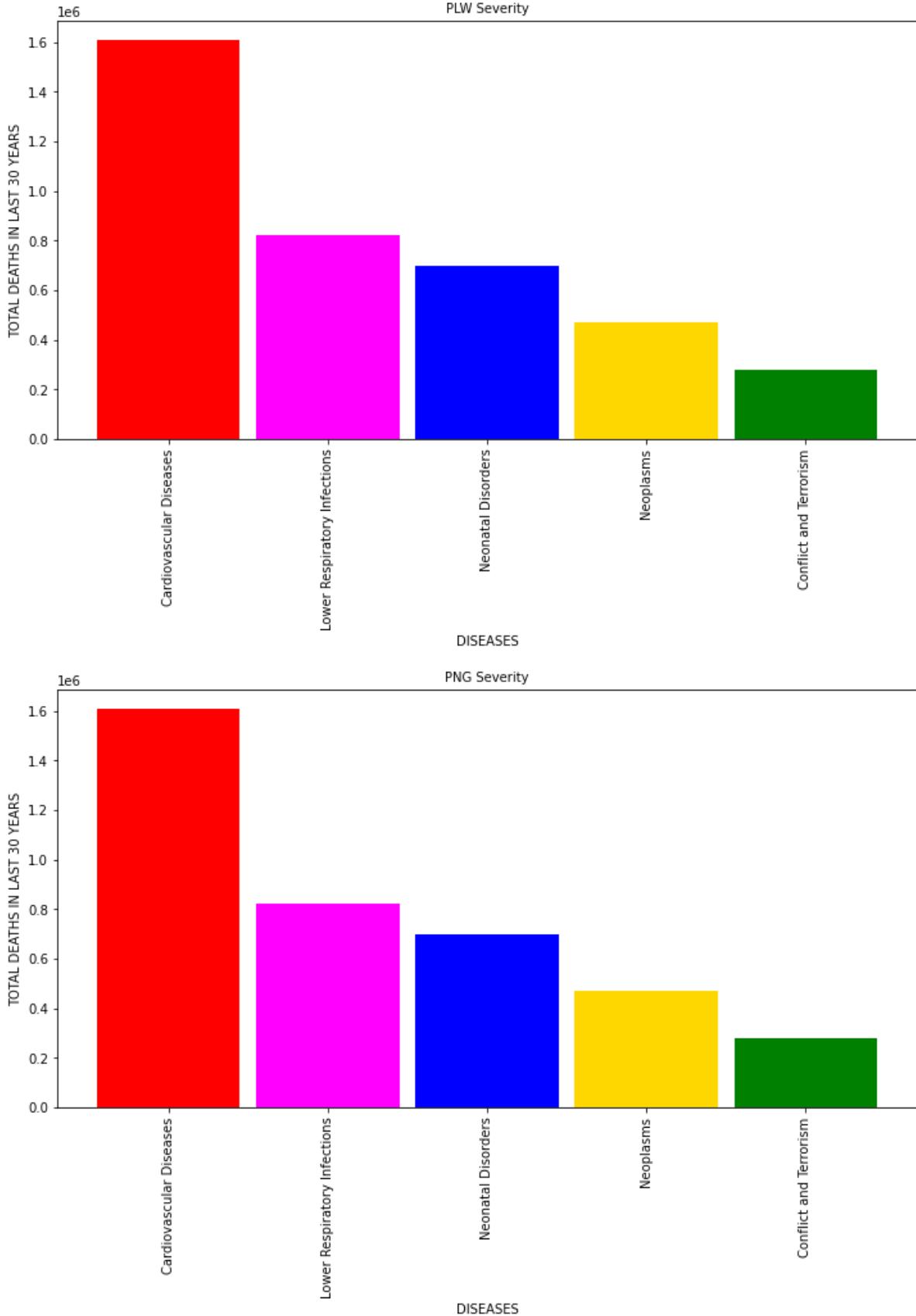


Cause Of Death



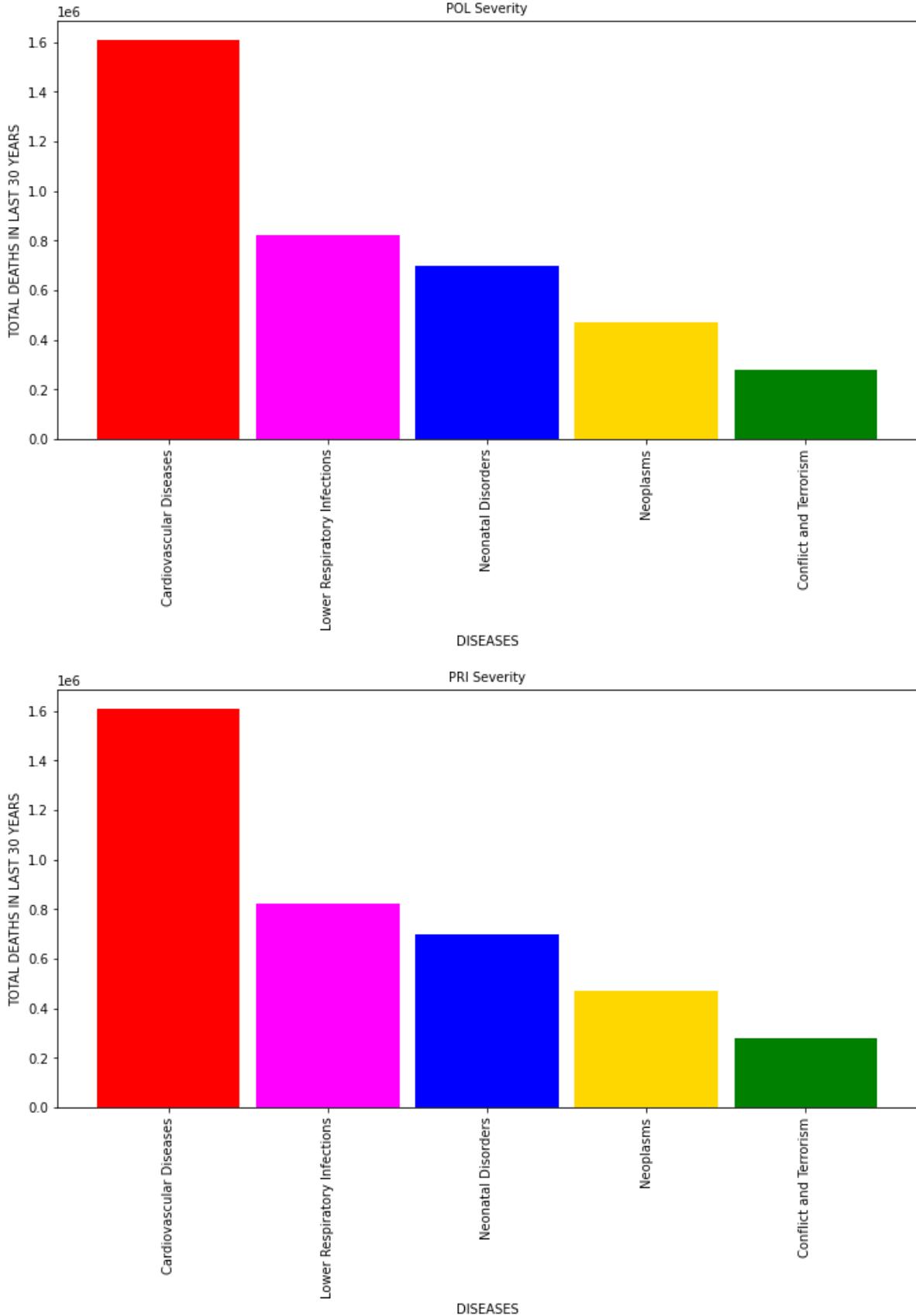
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death

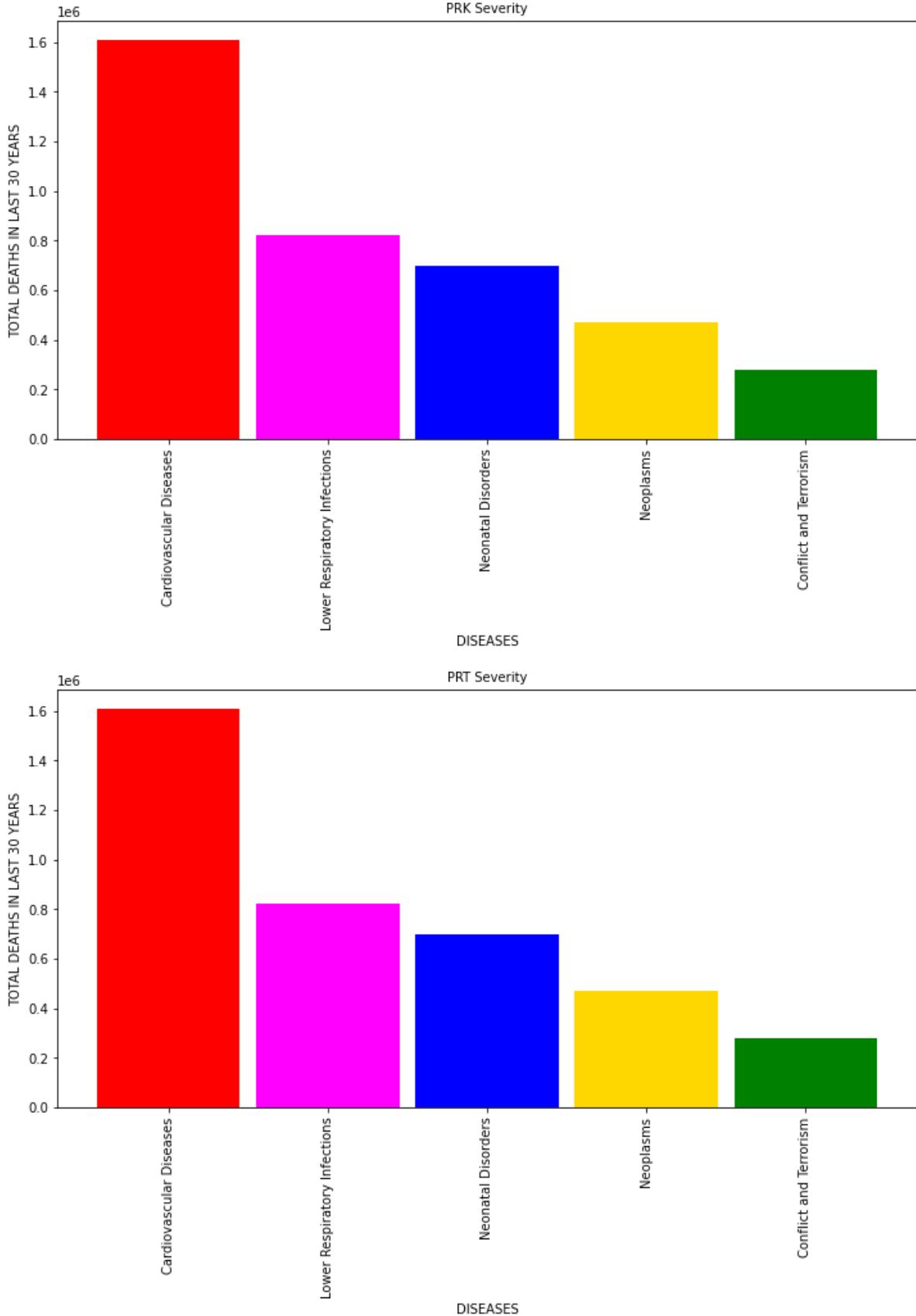


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death

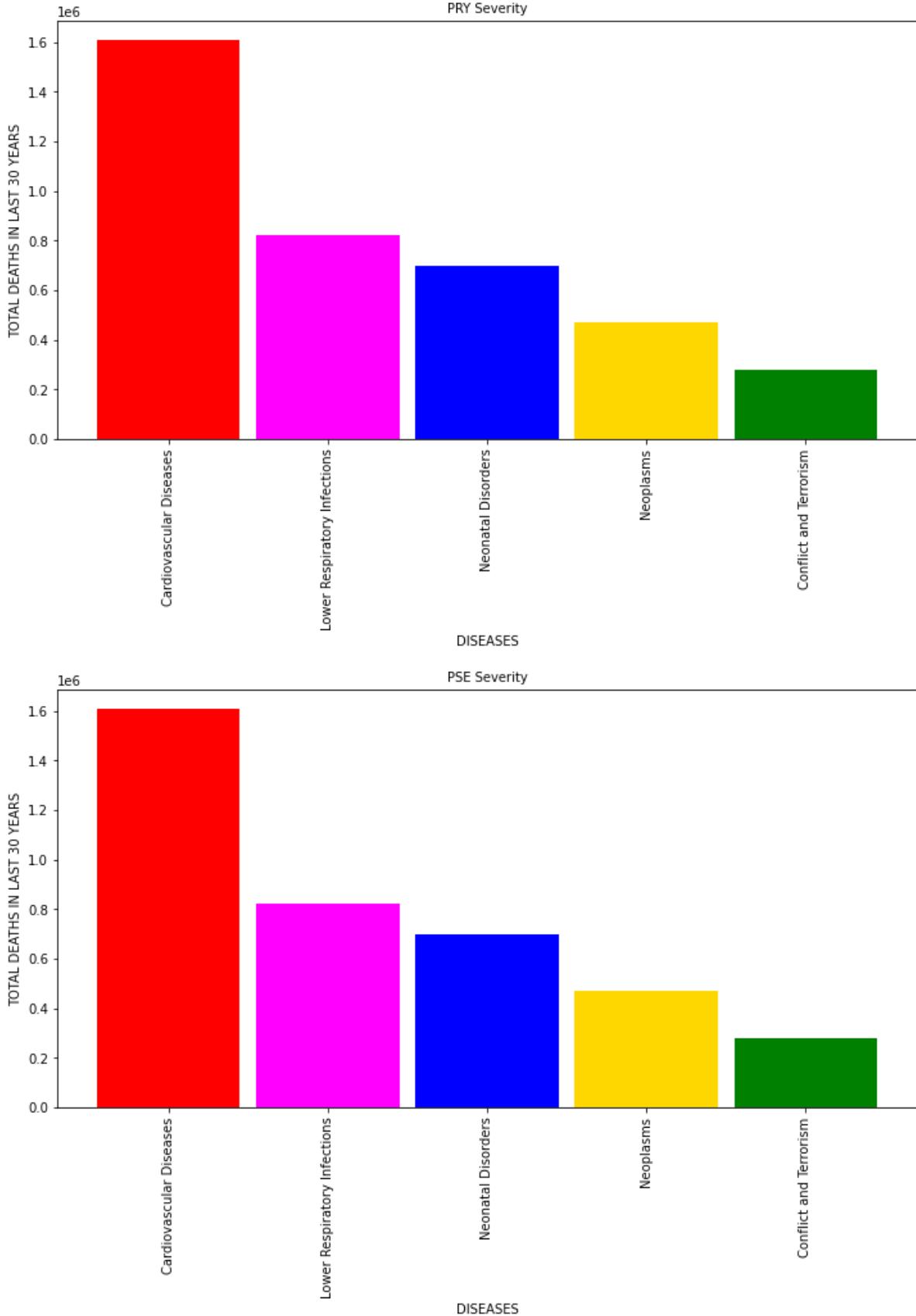


Cause Of Death



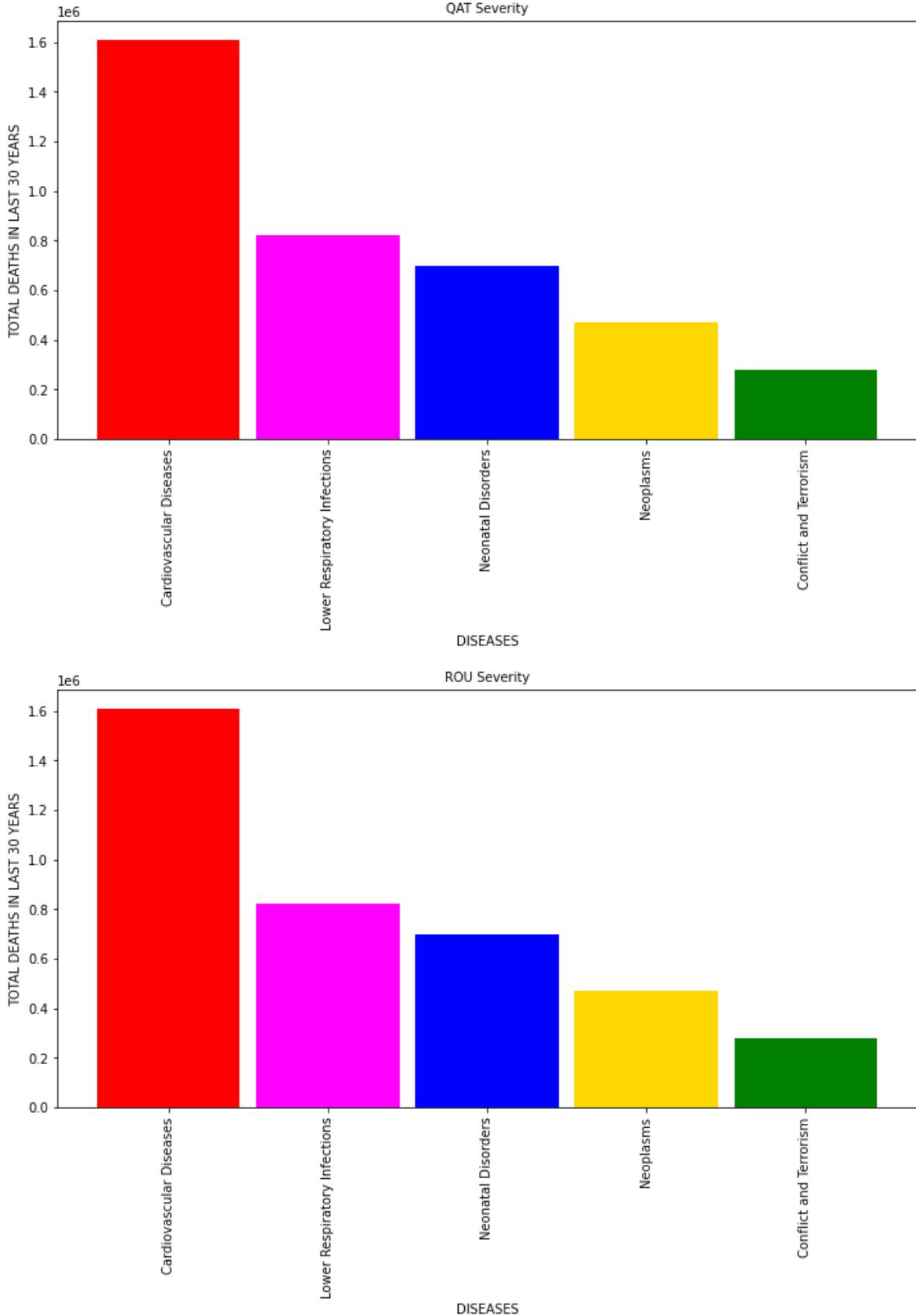
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



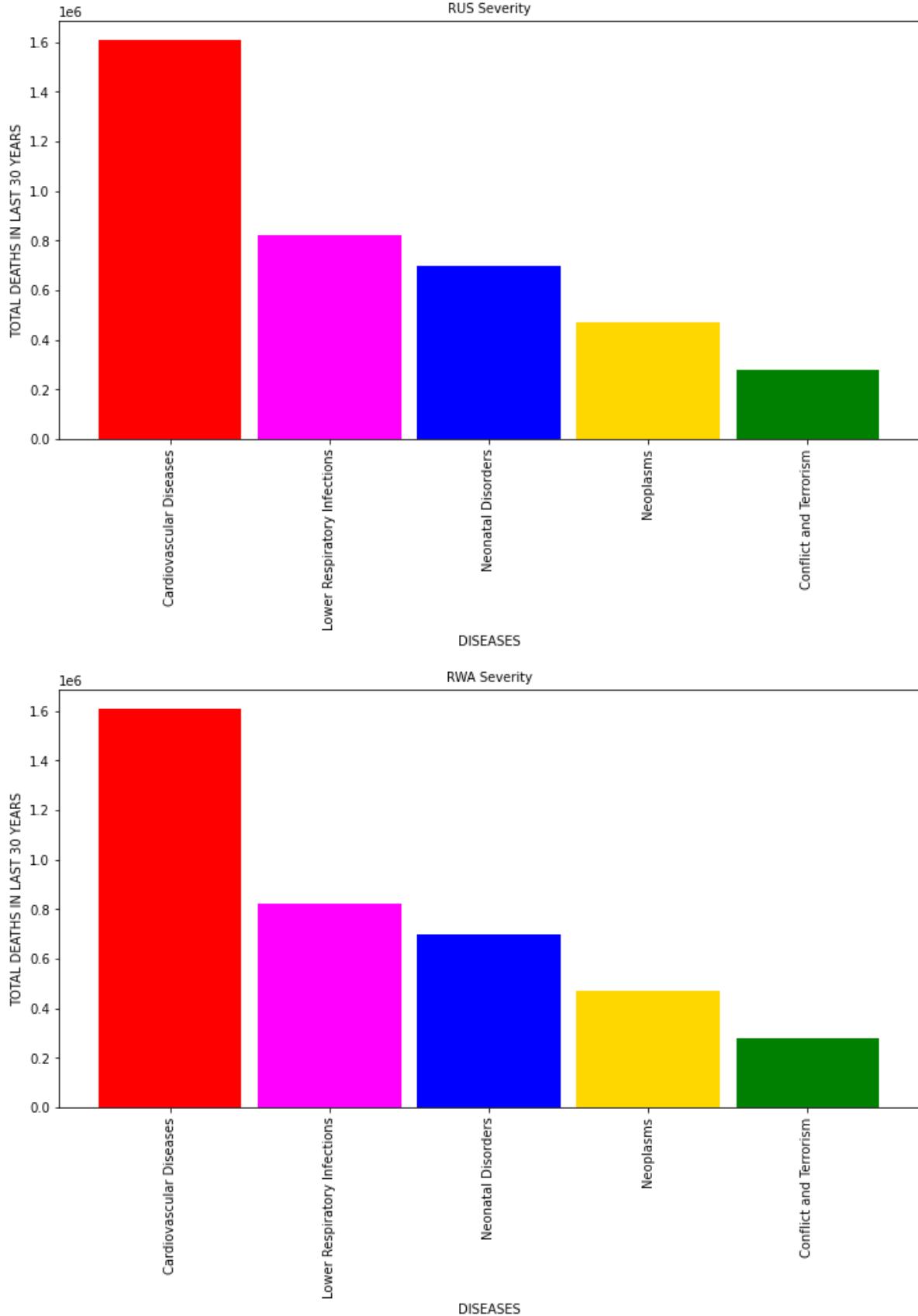
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



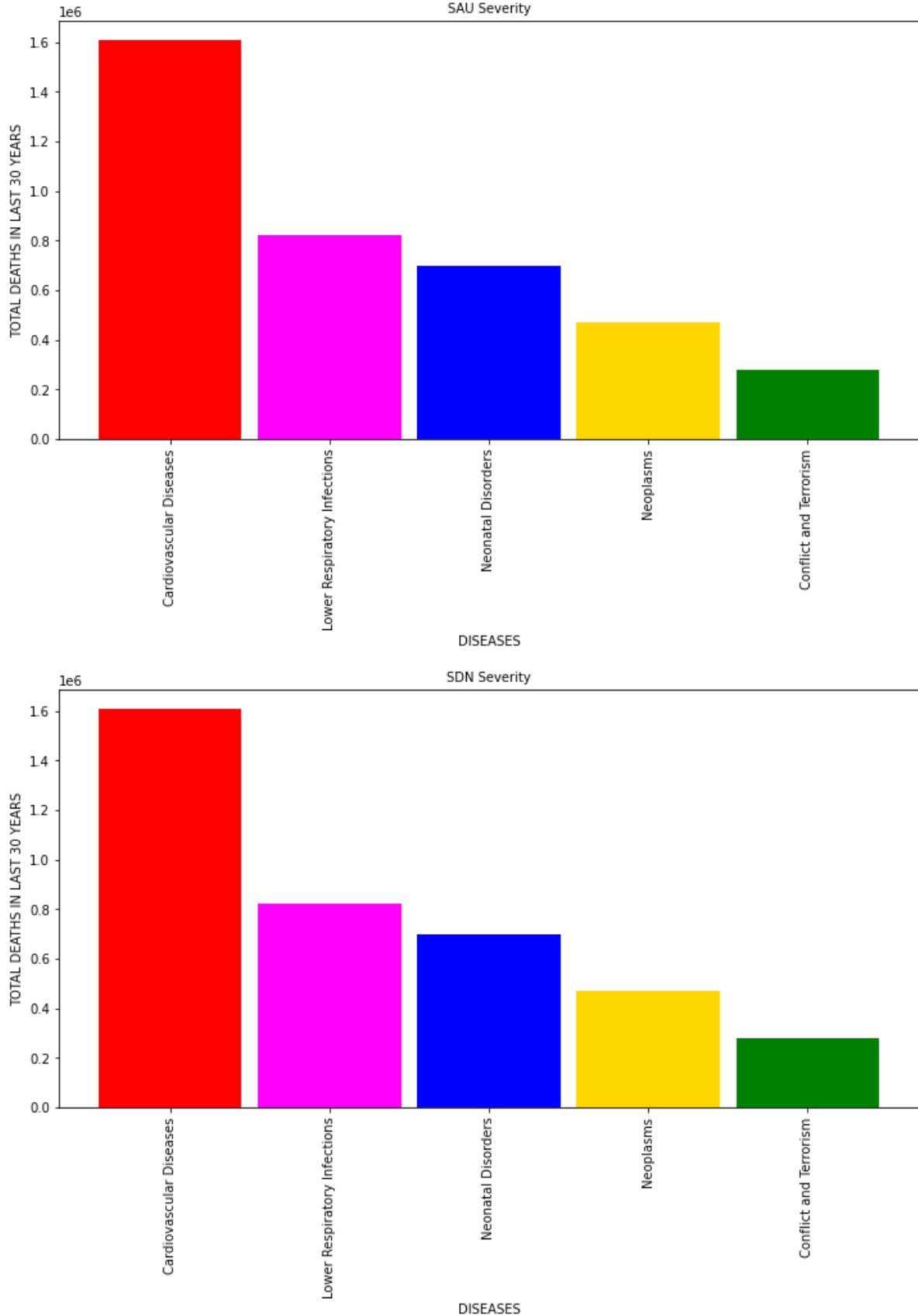
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



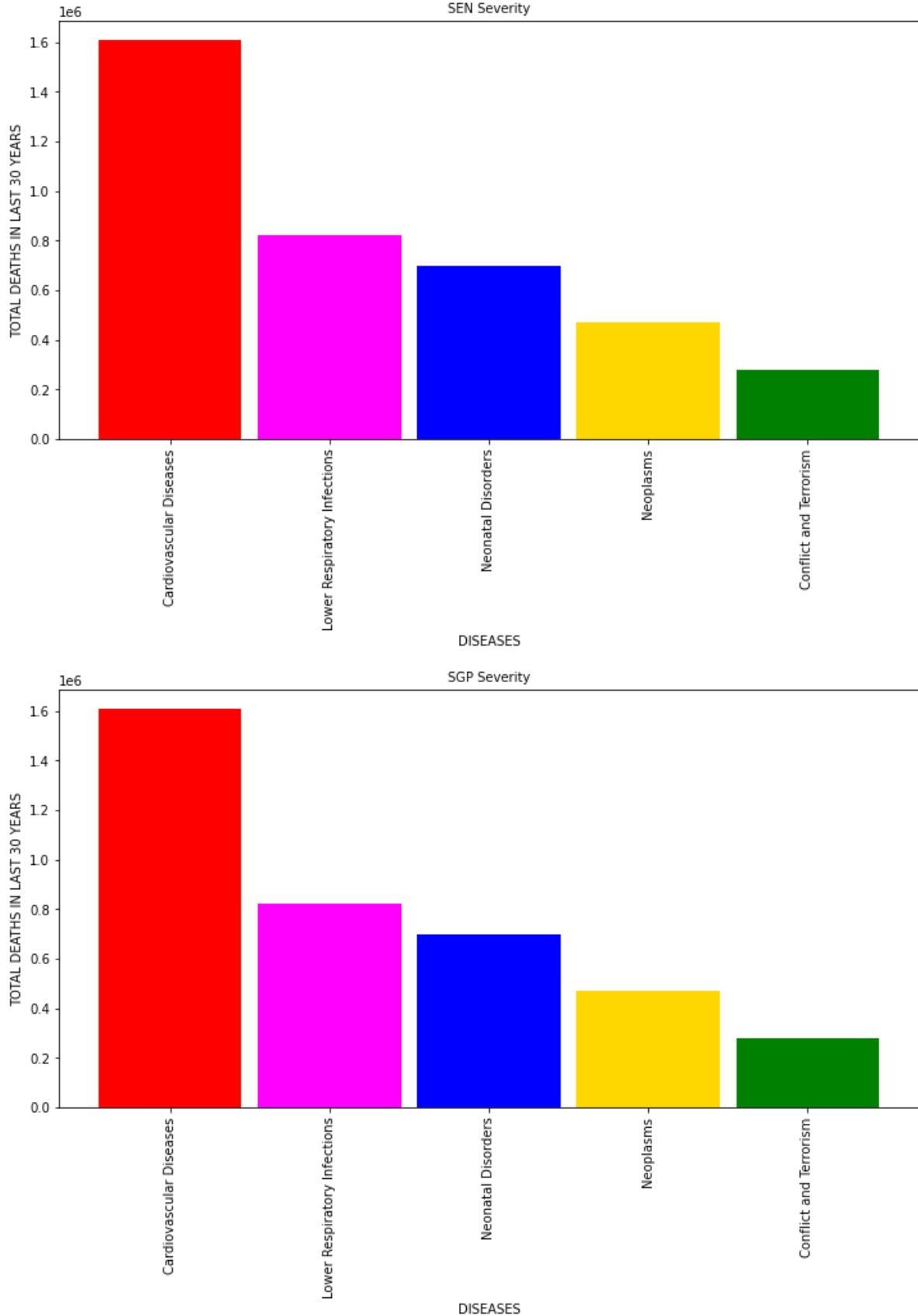
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



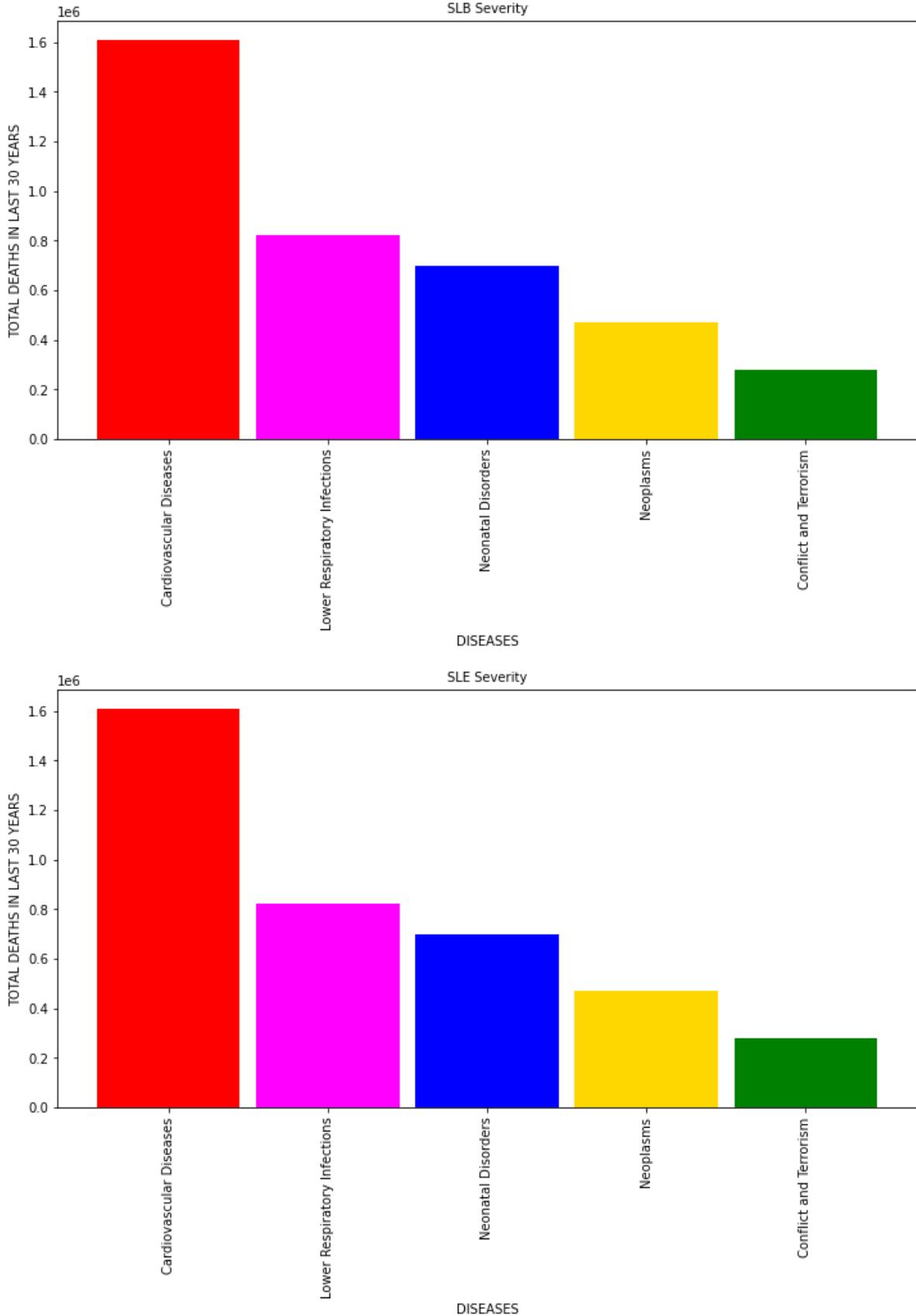
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



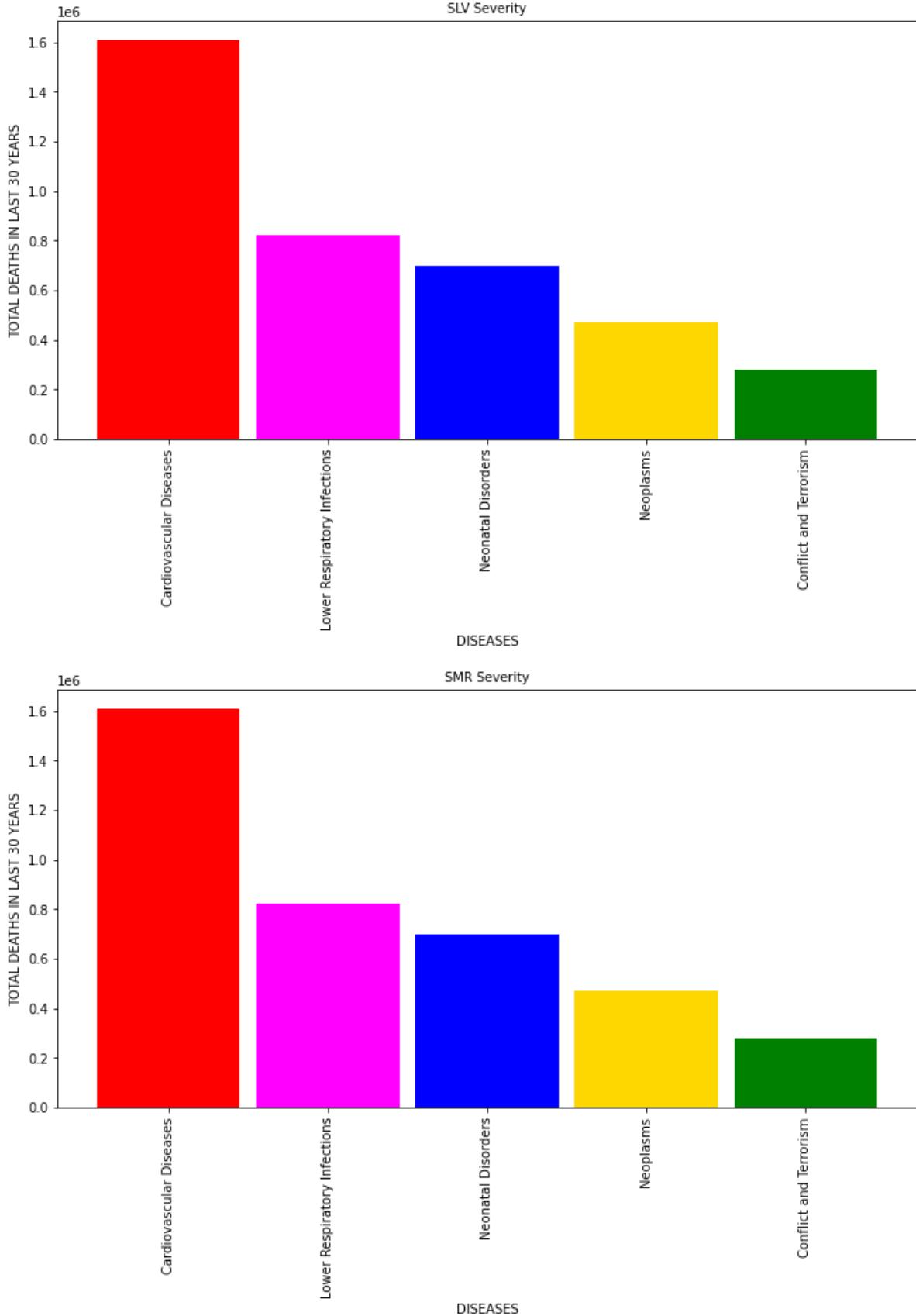
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



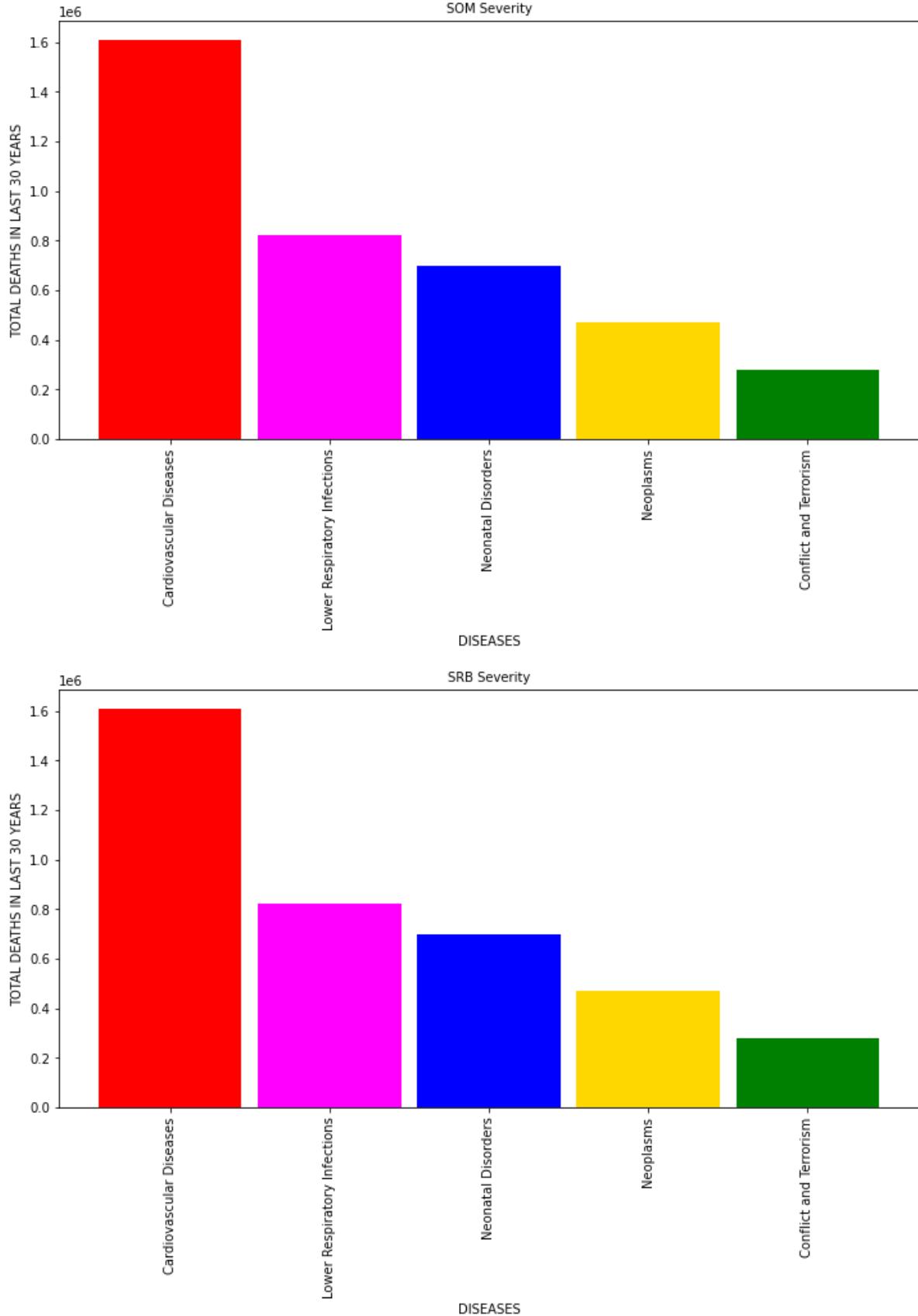
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



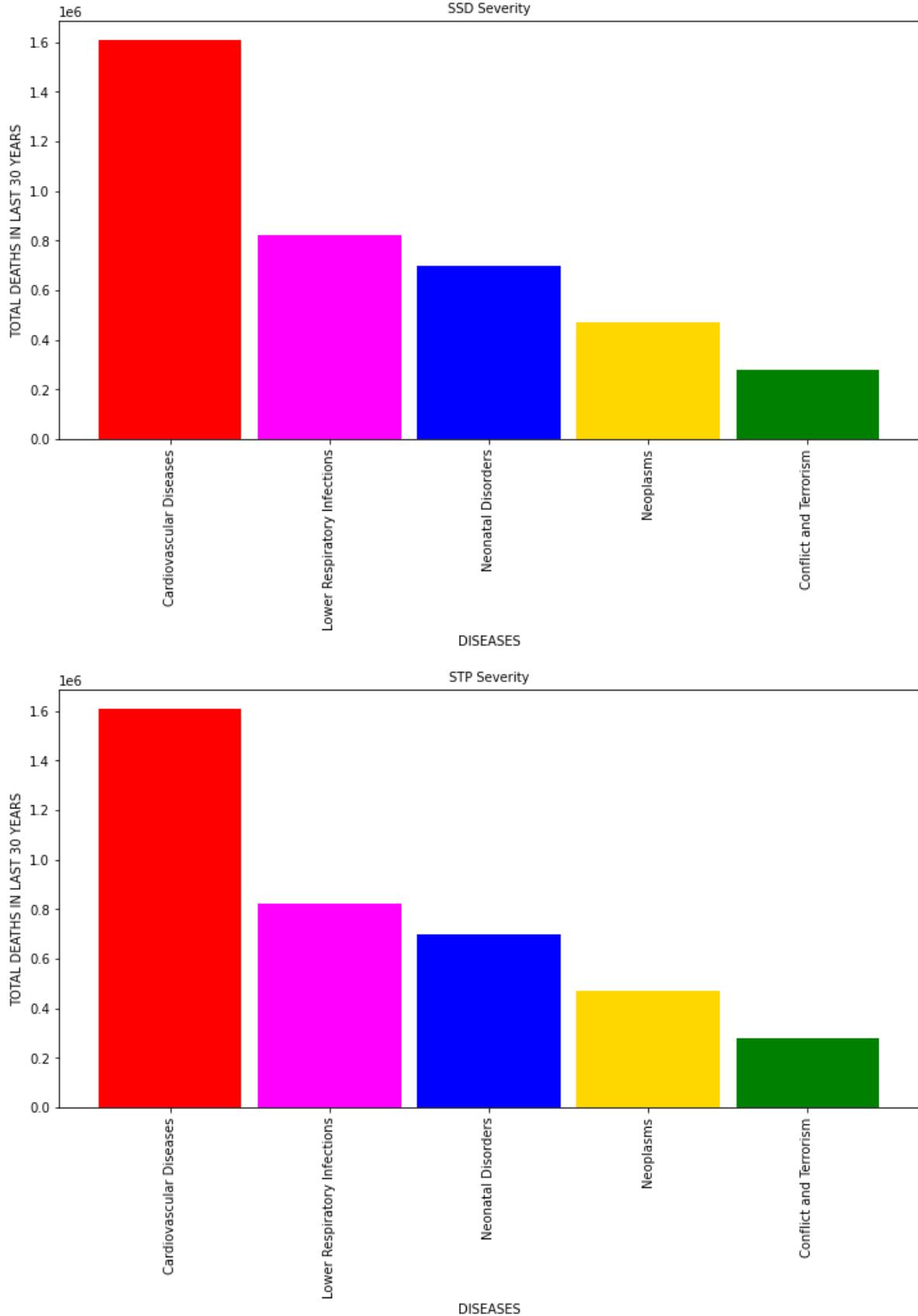
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



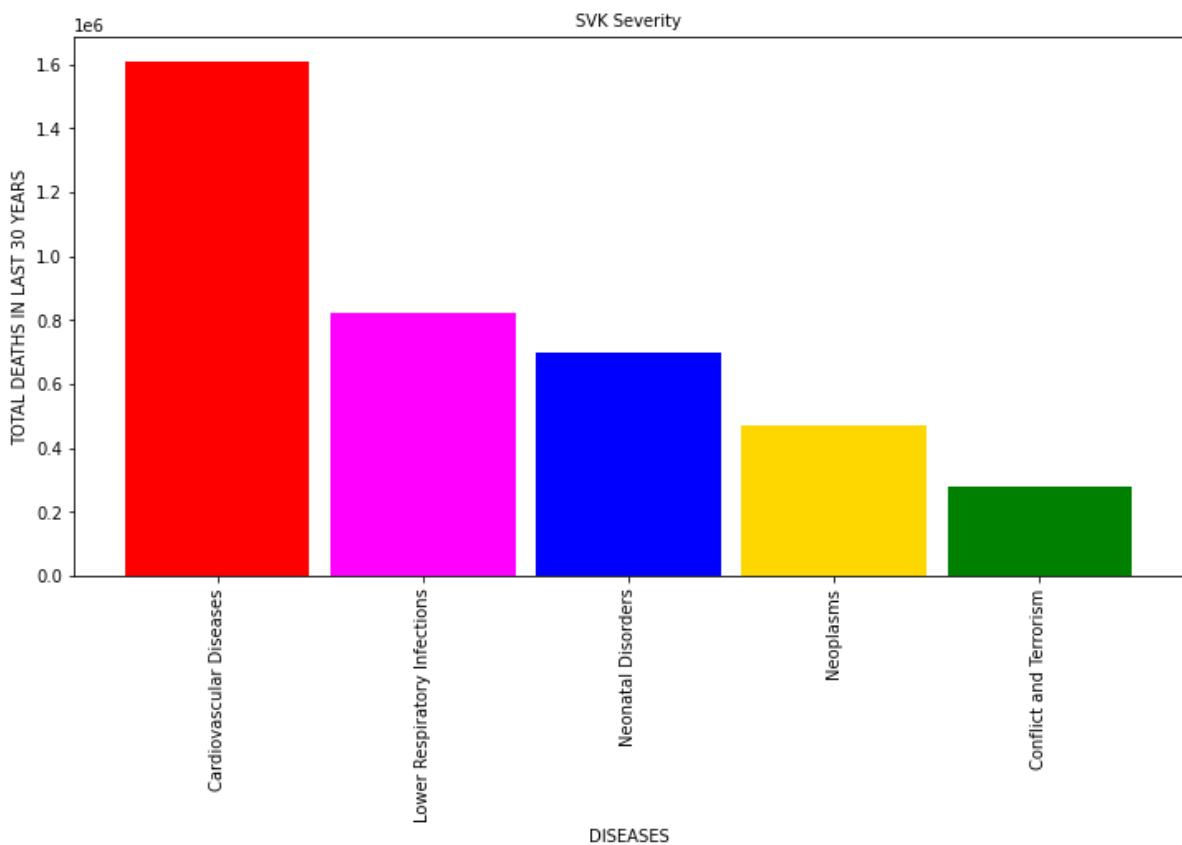
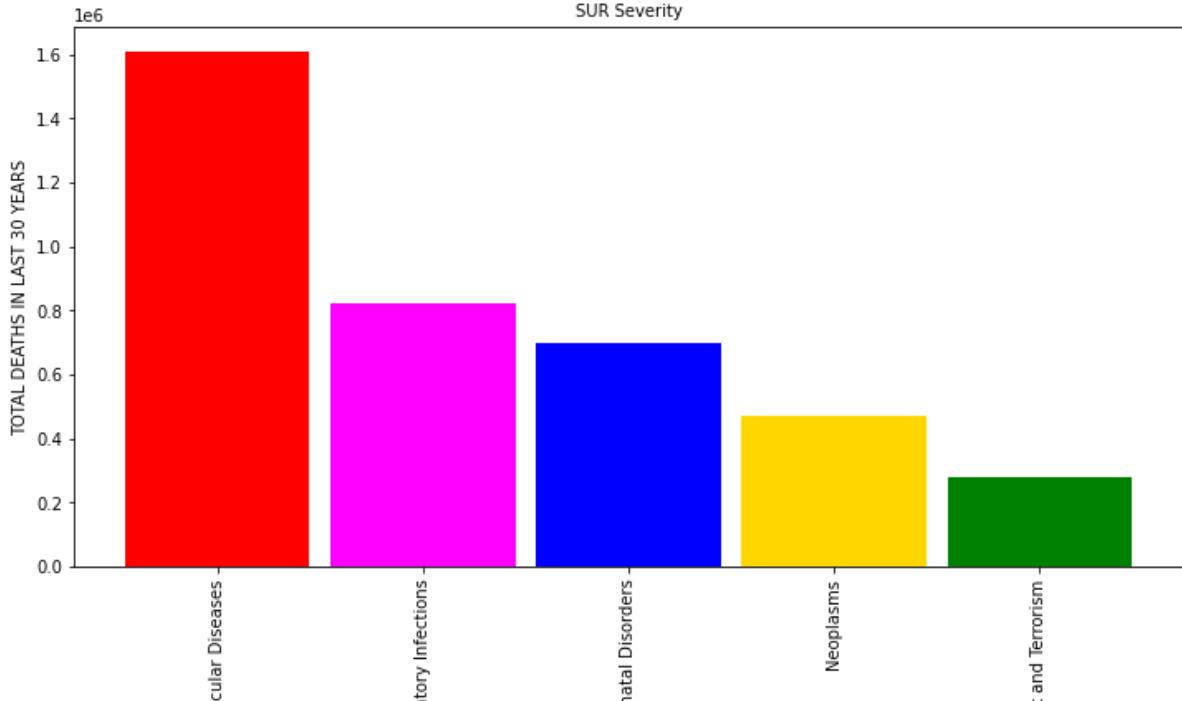
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death

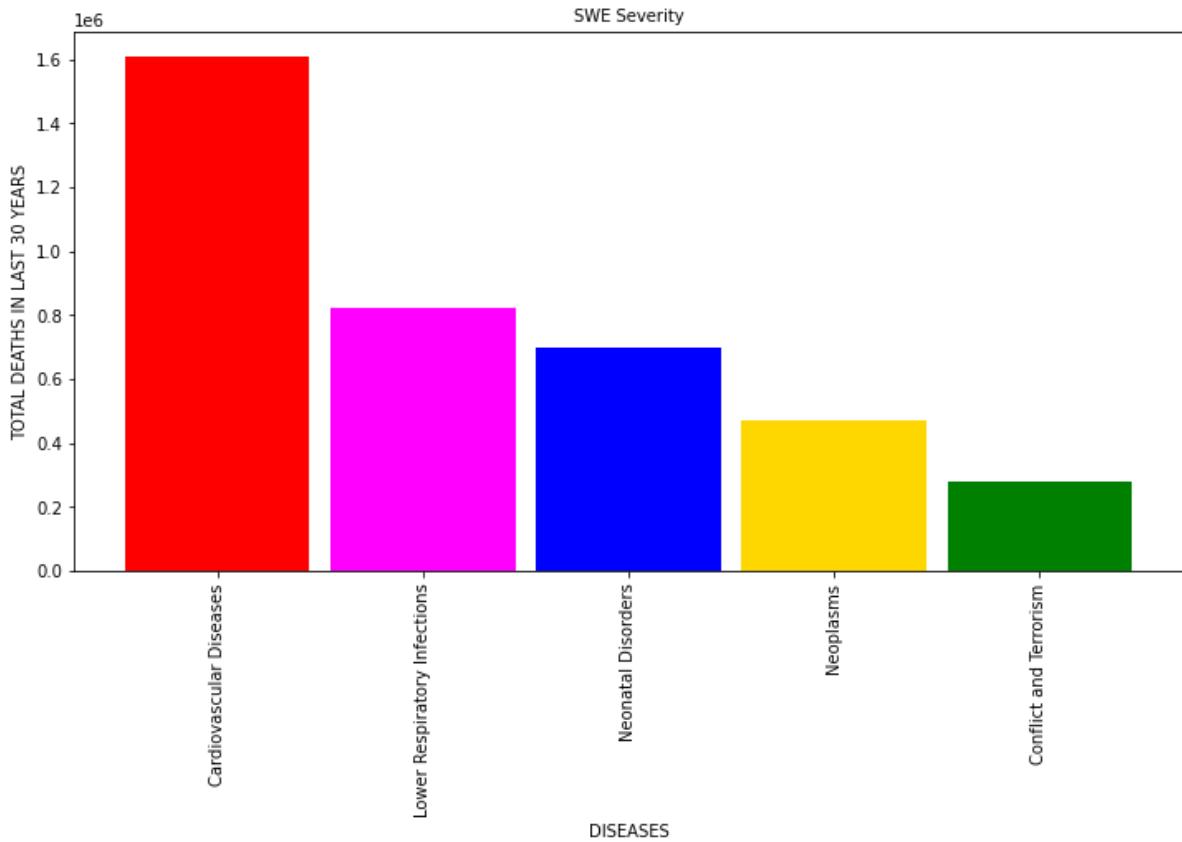
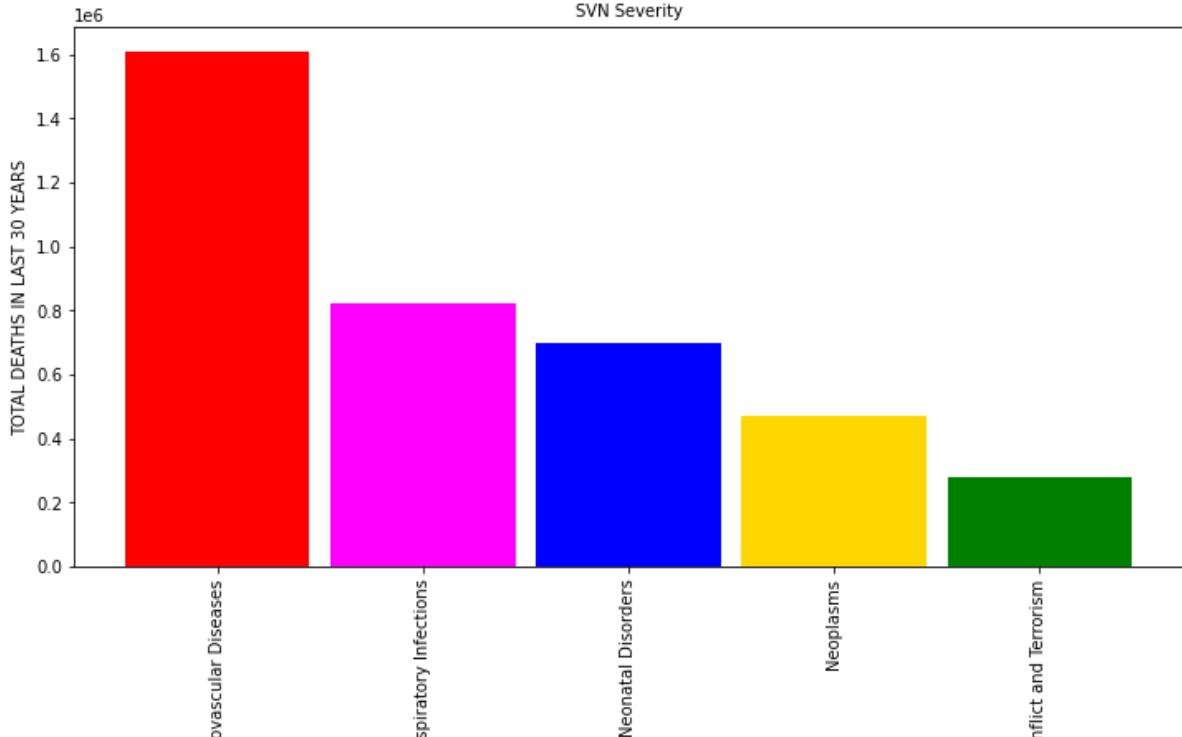


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death

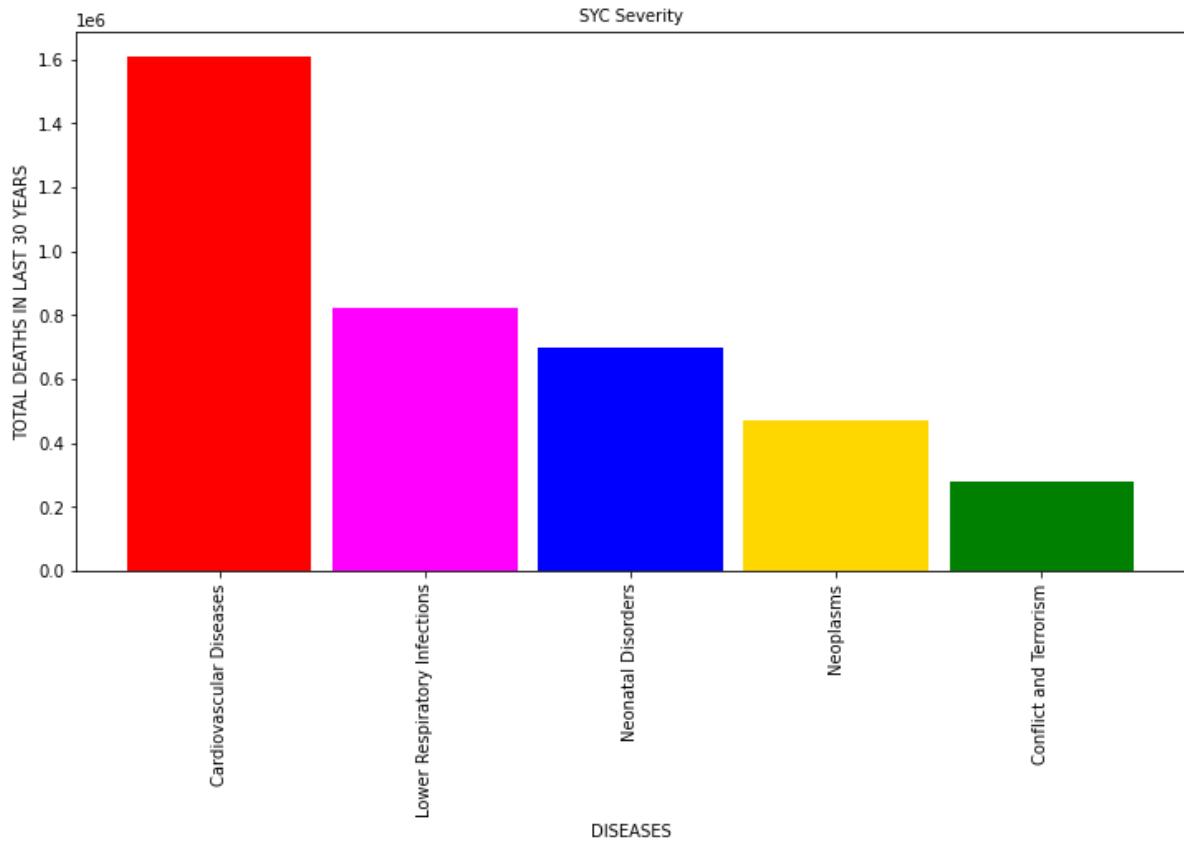
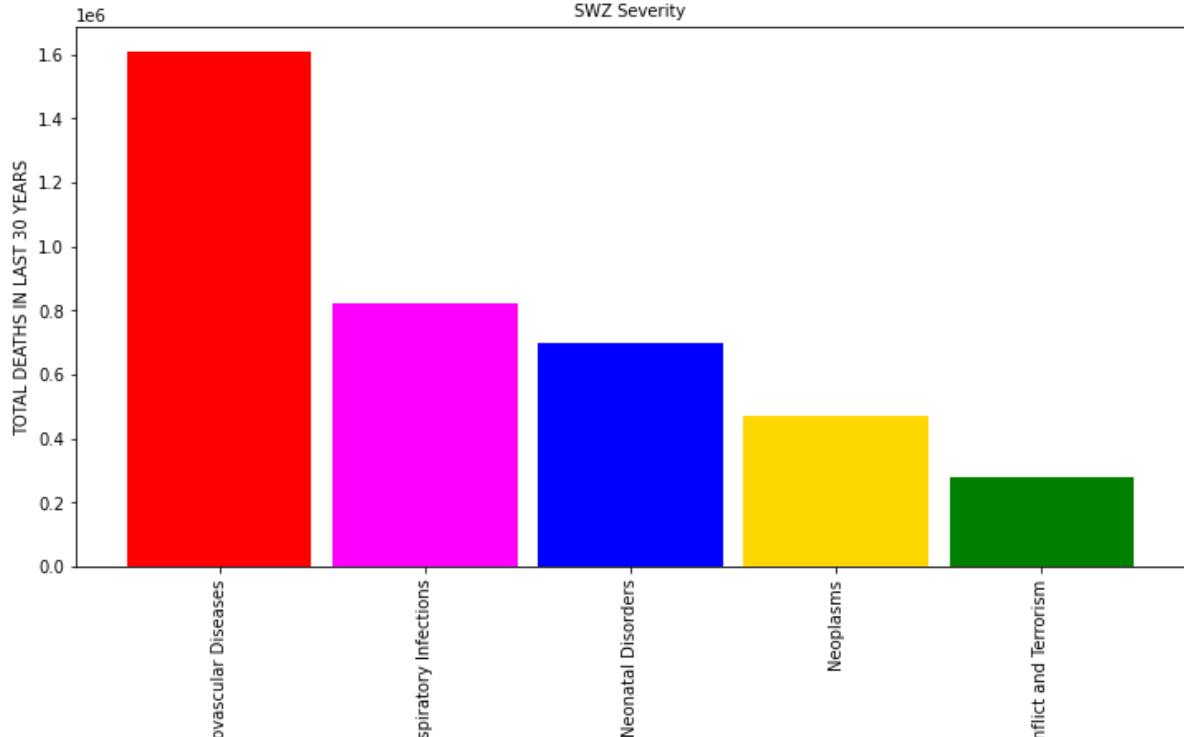


Cause Of Death



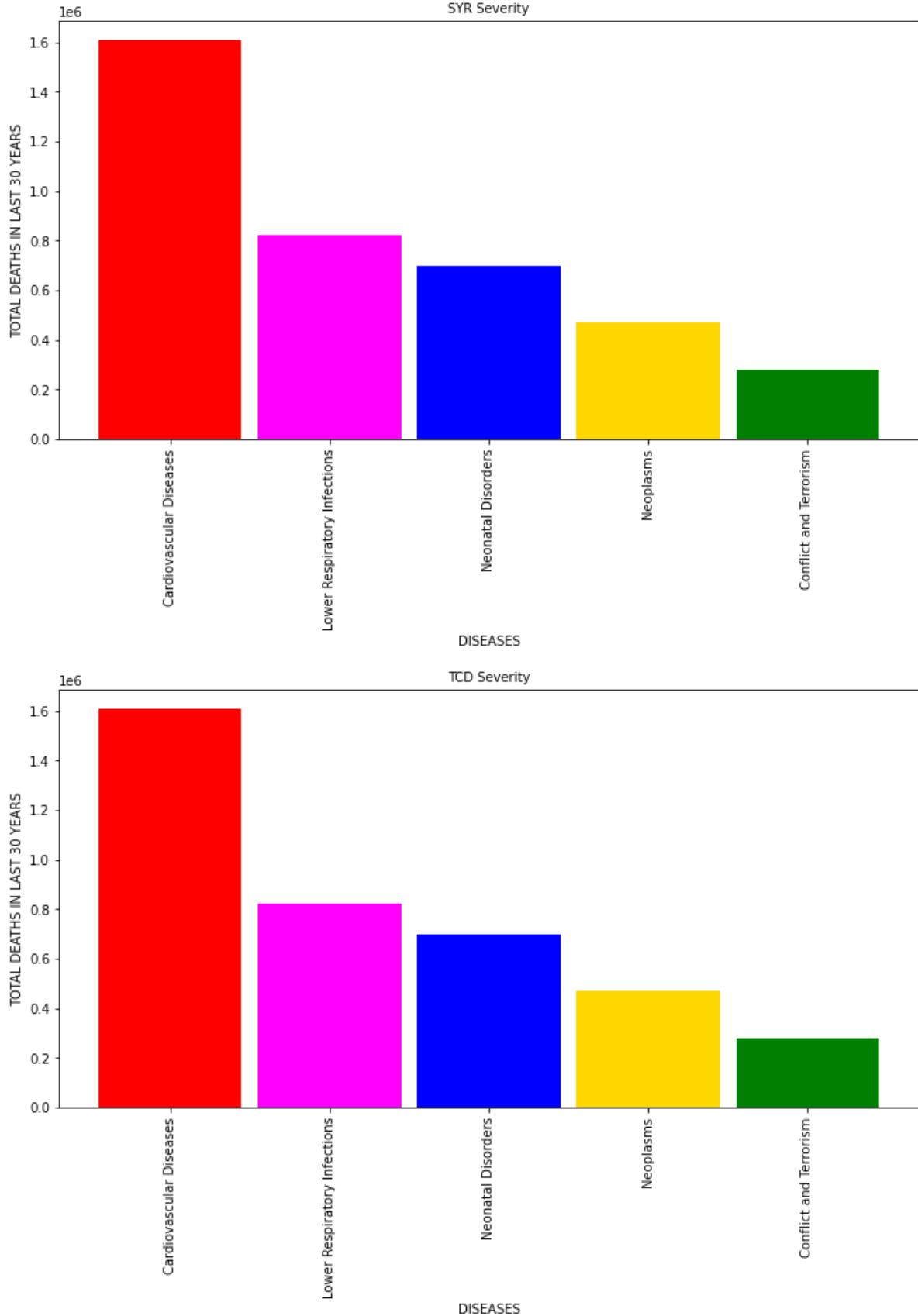
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



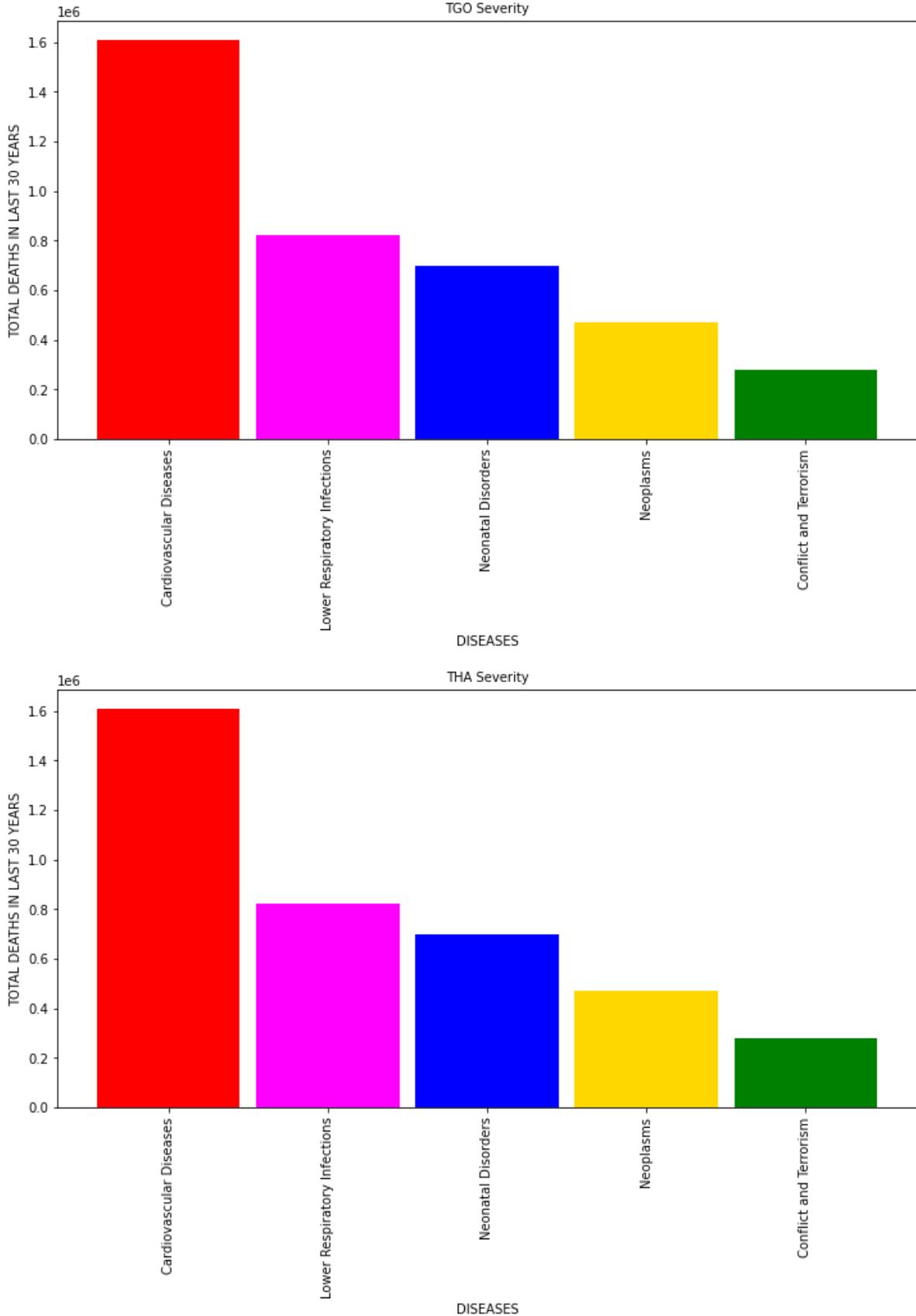
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



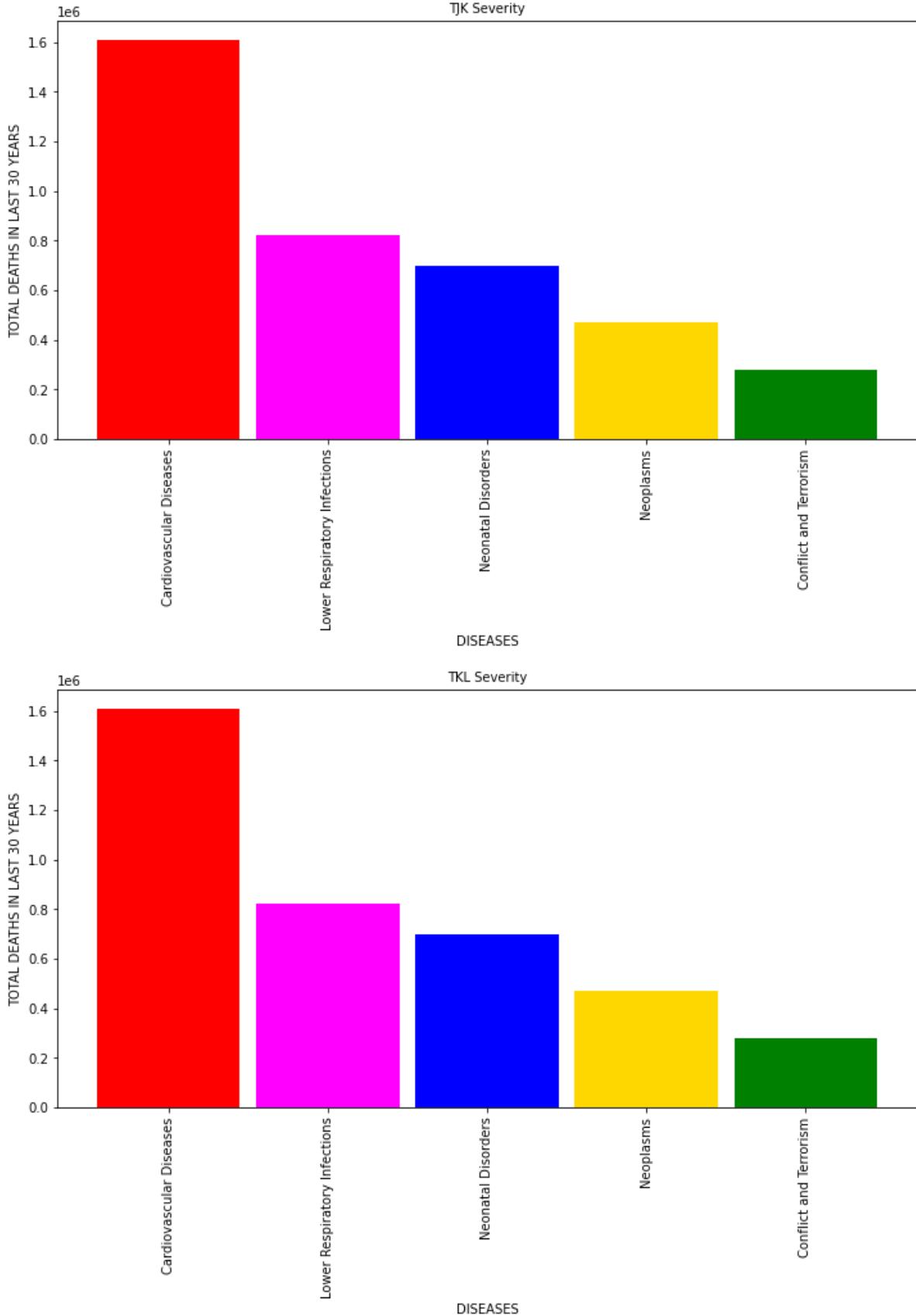
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



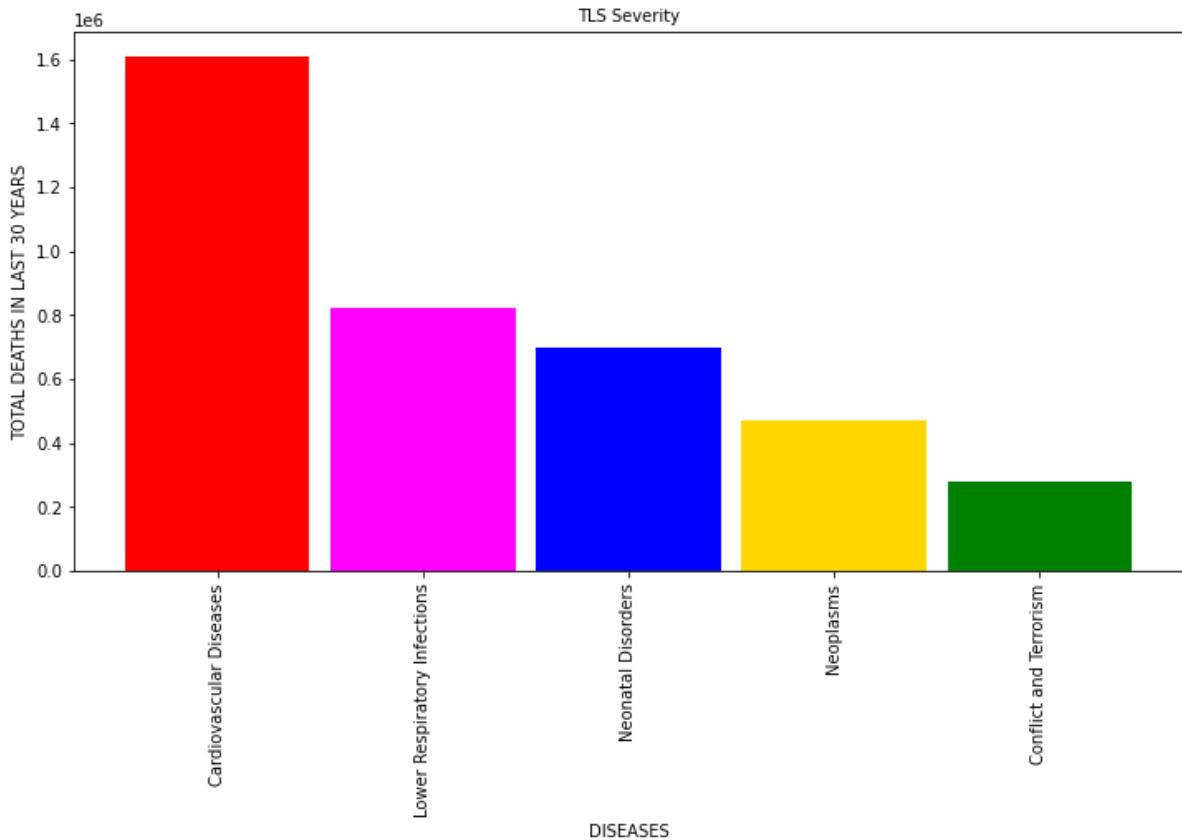
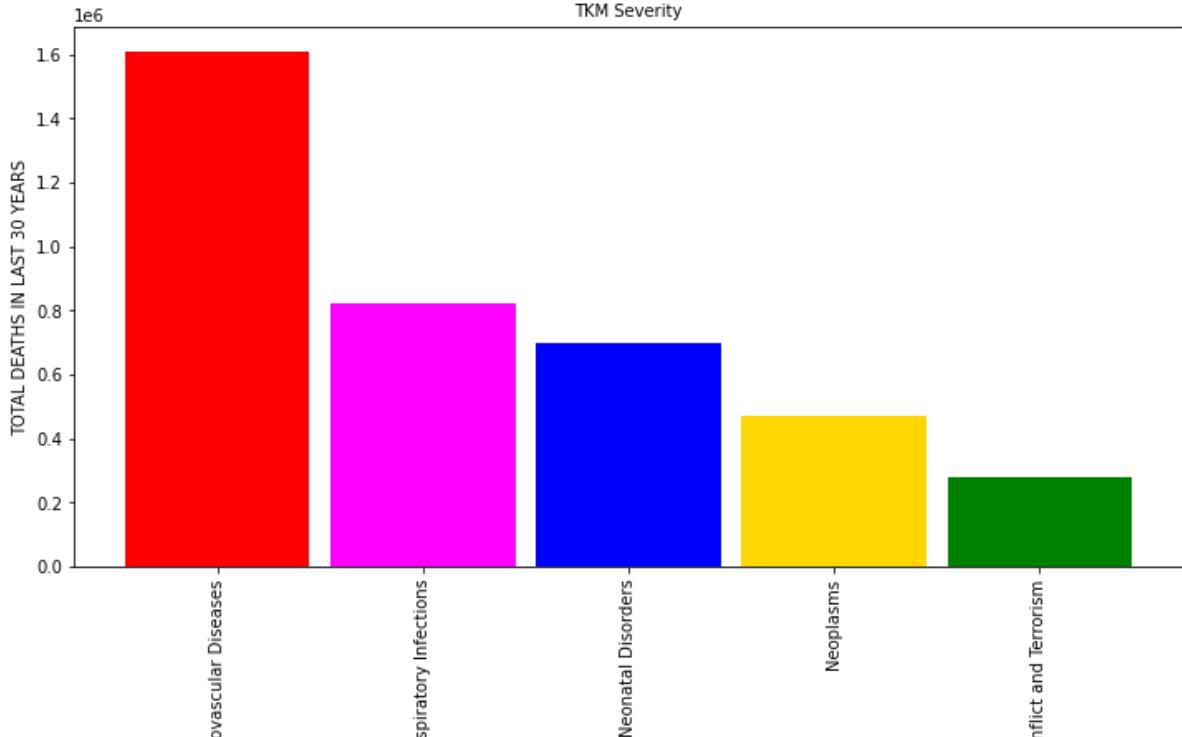
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death

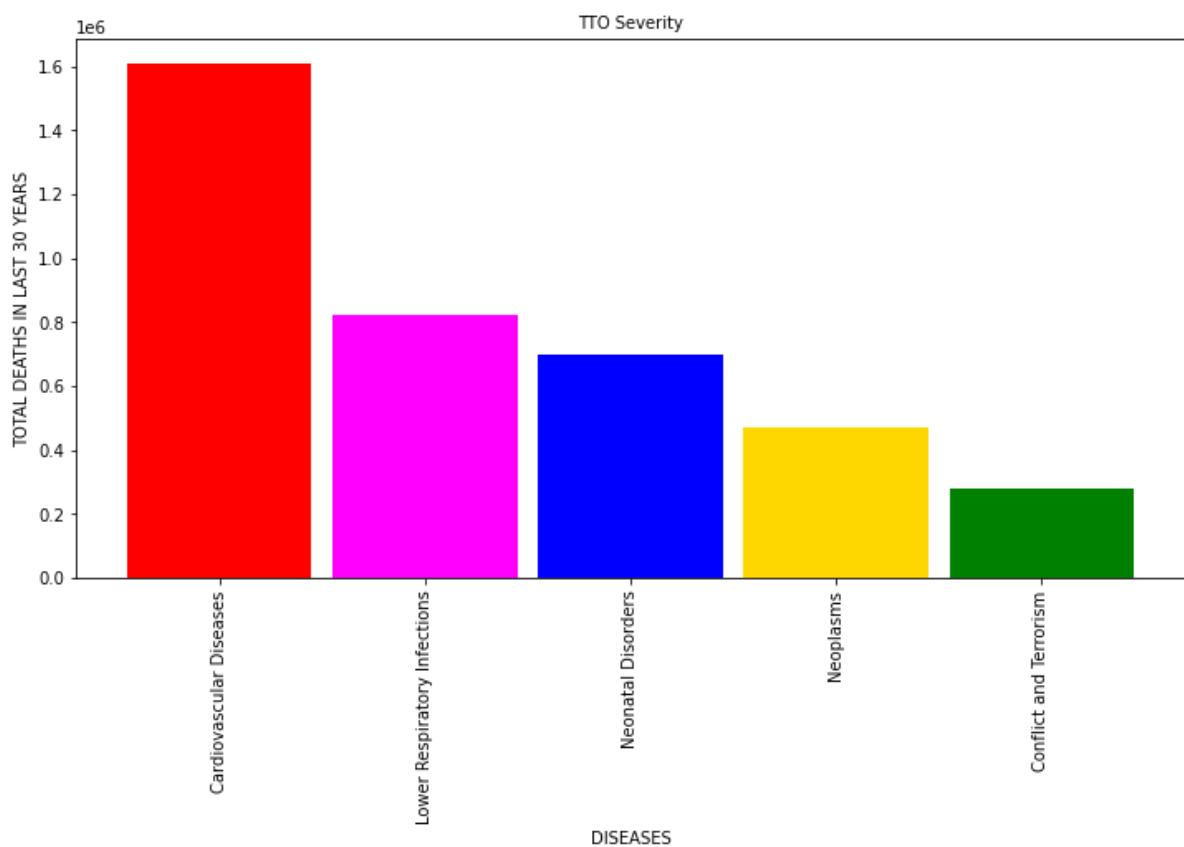
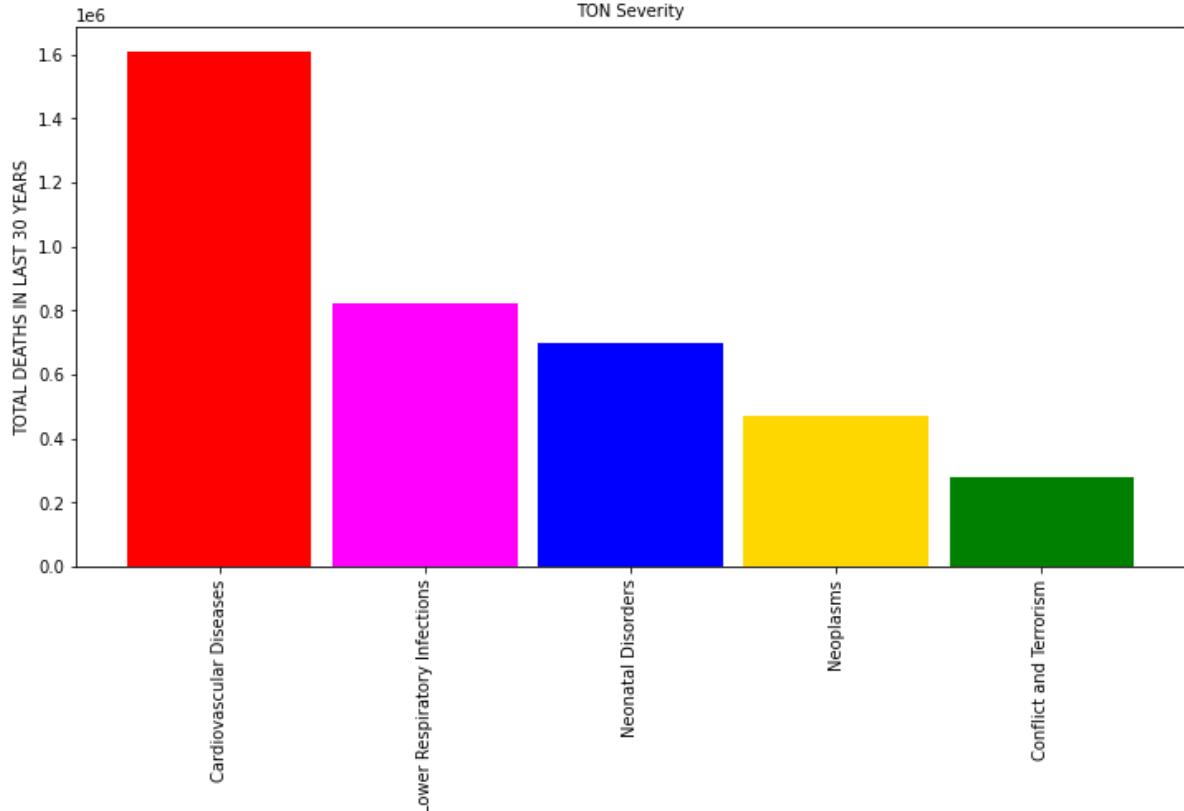


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

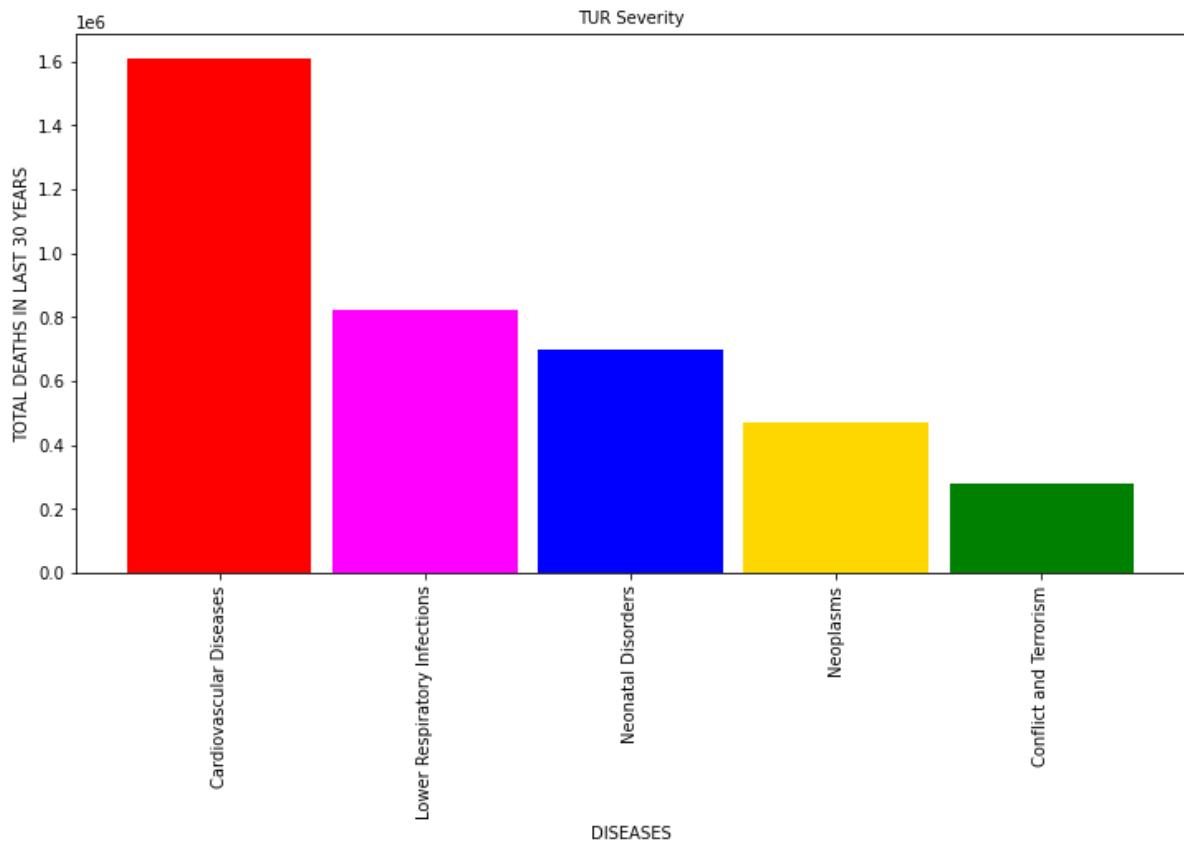
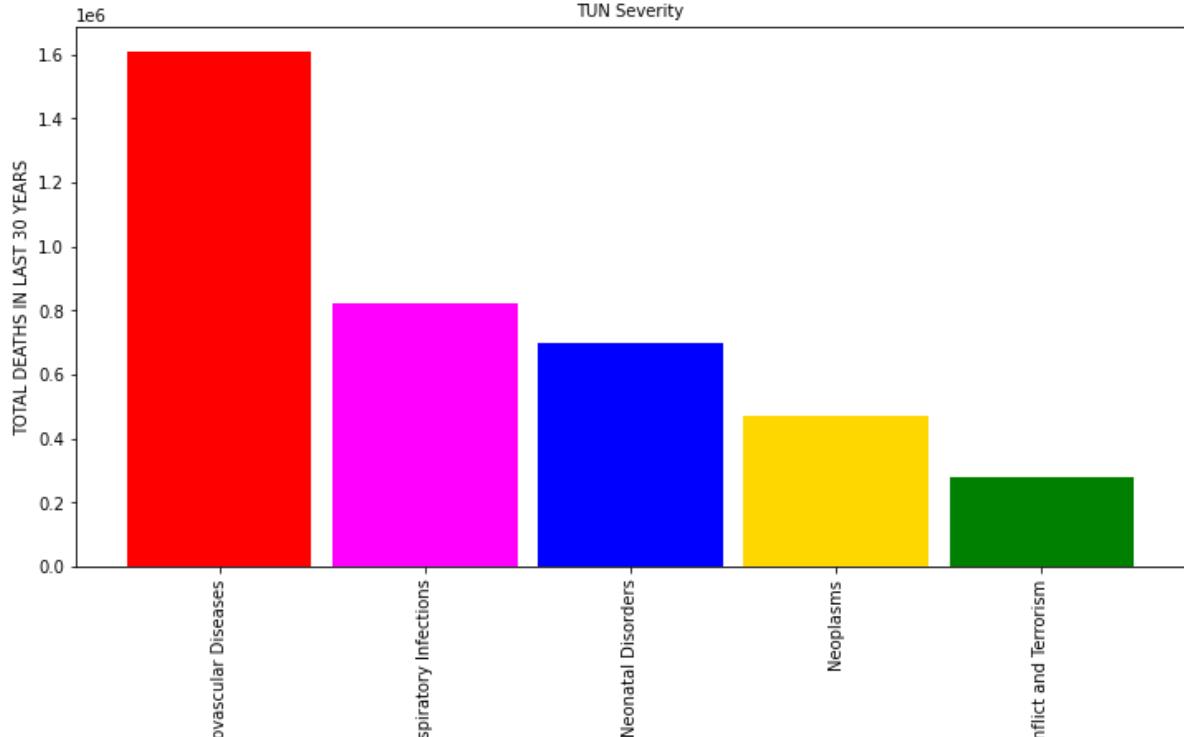
Cause Of Death

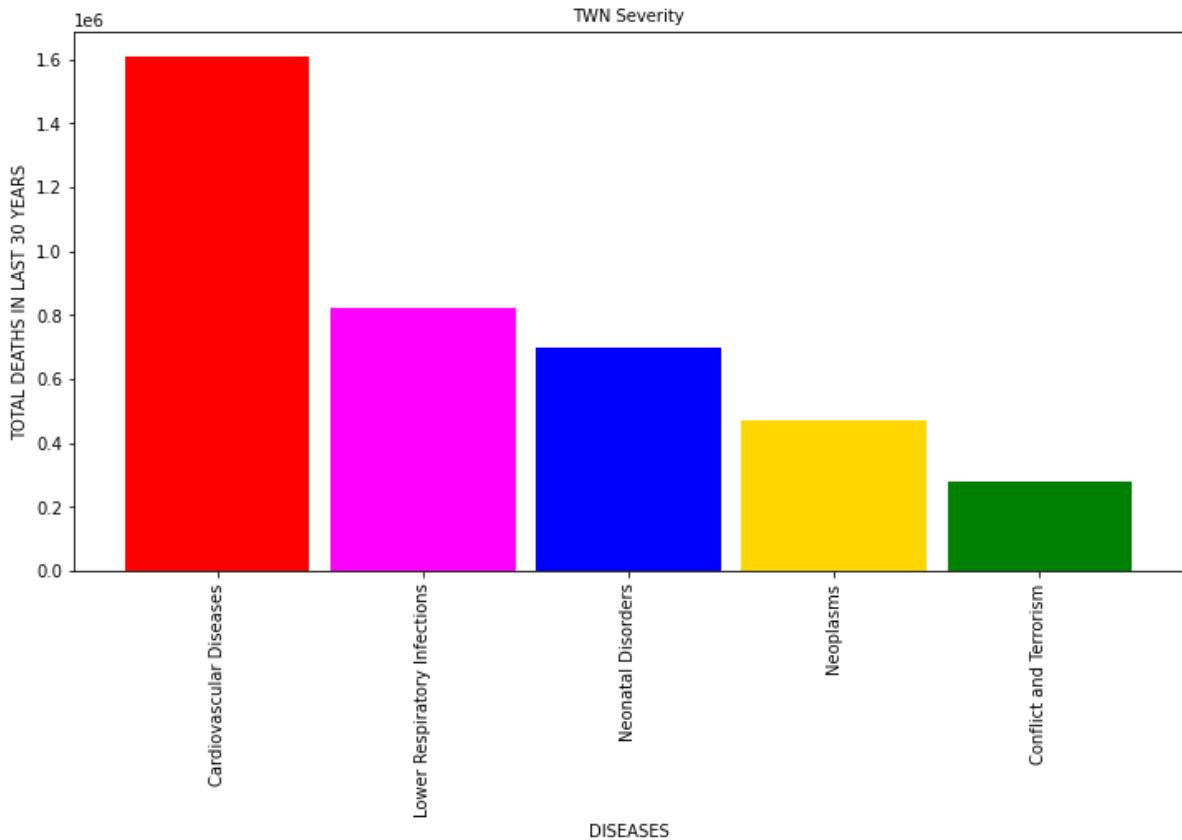
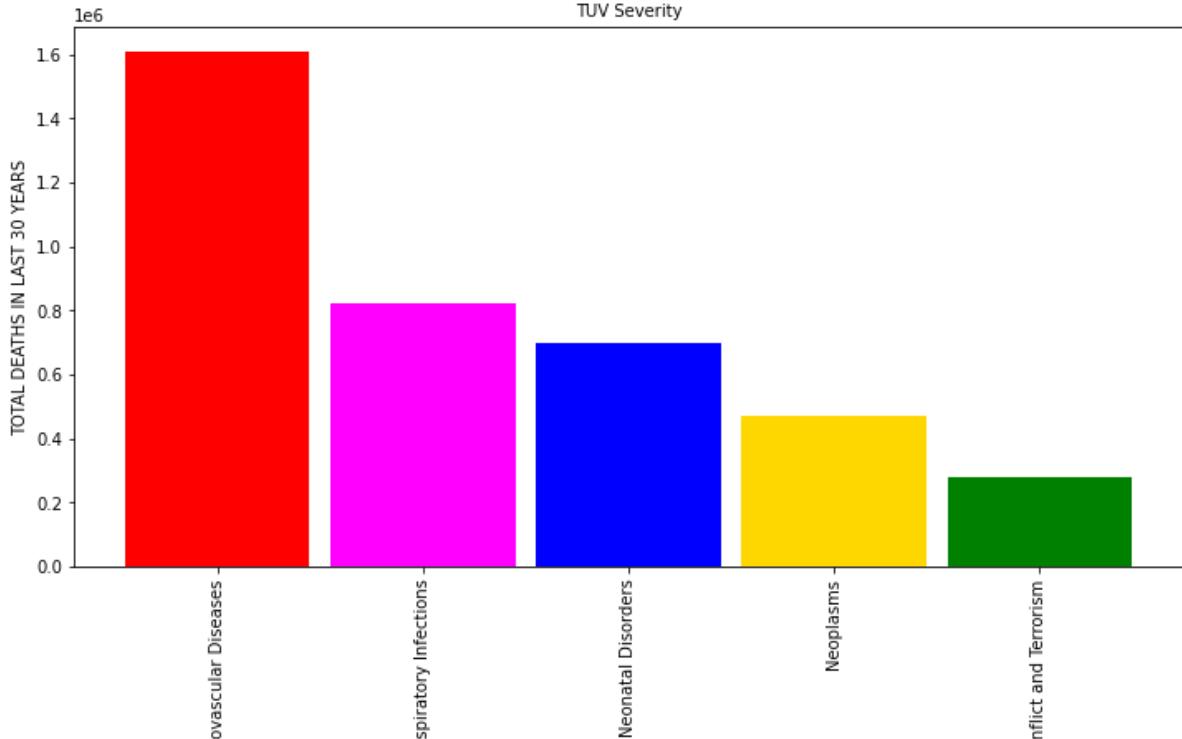


Cause Of Death

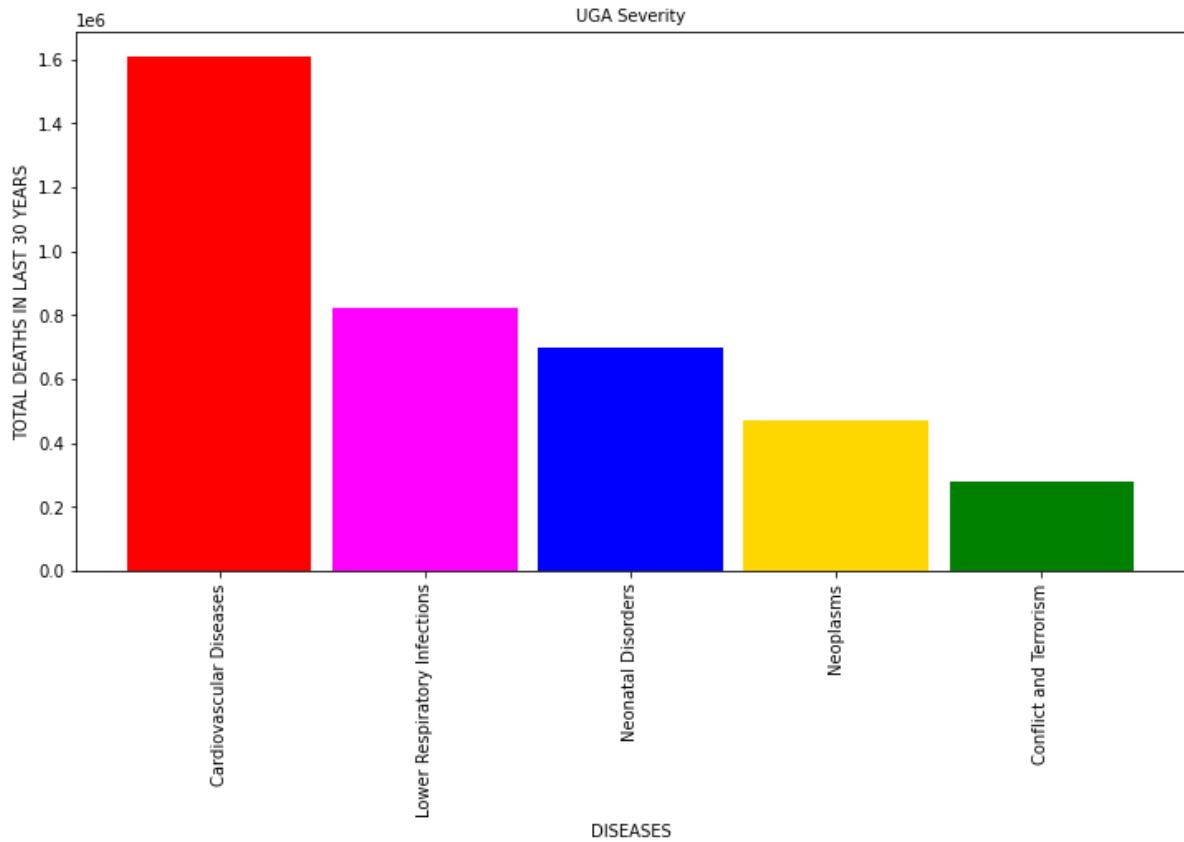
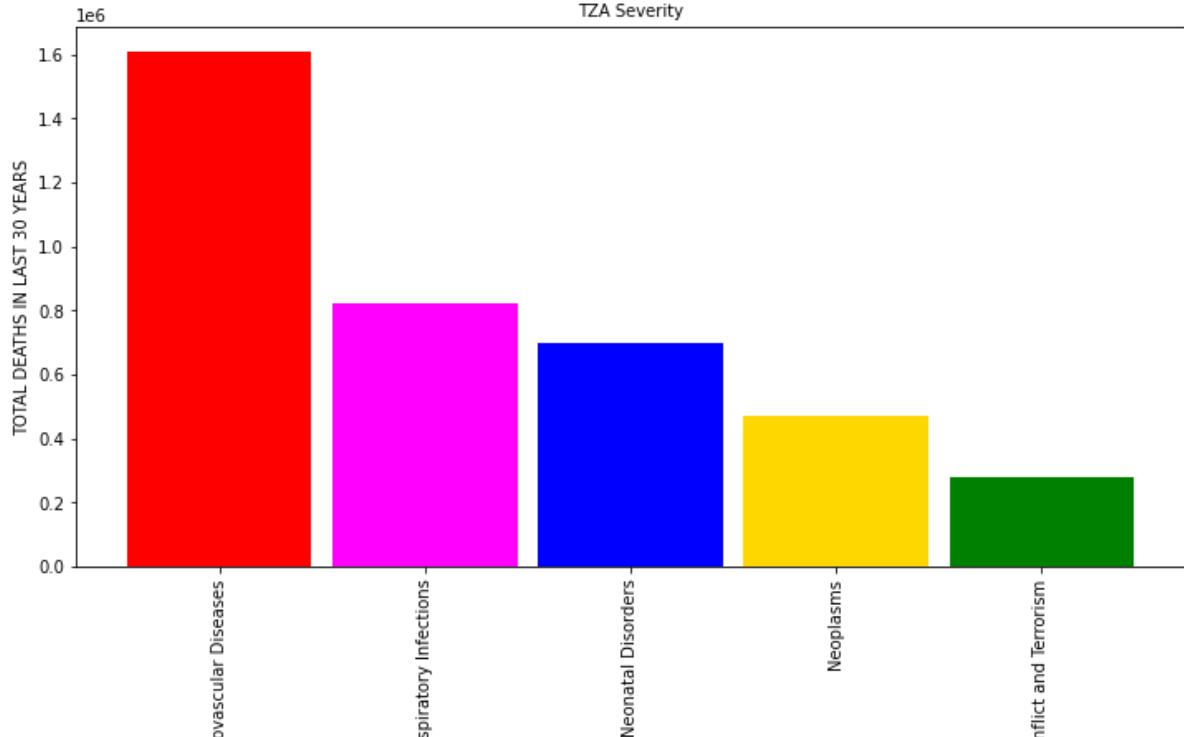


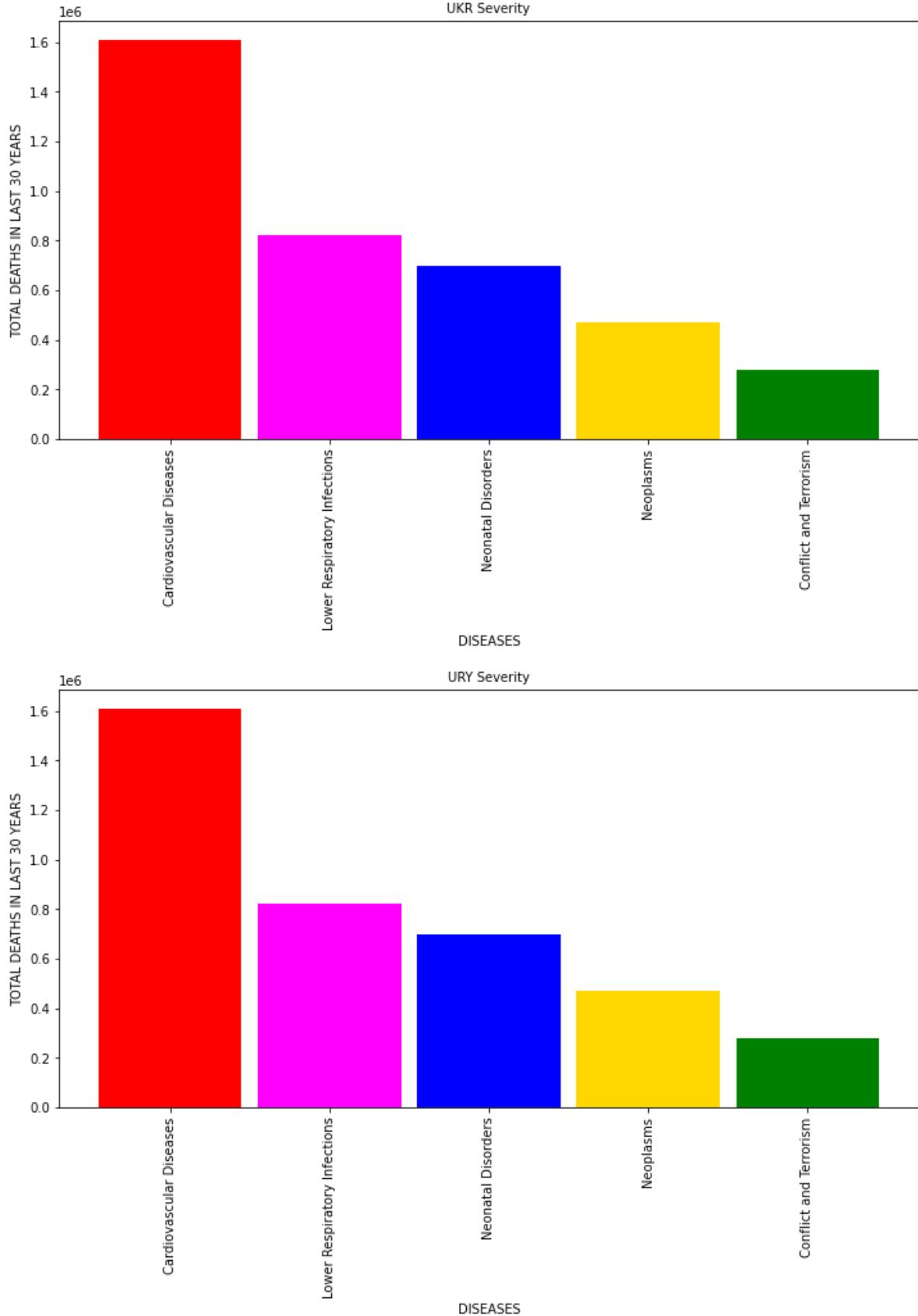
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



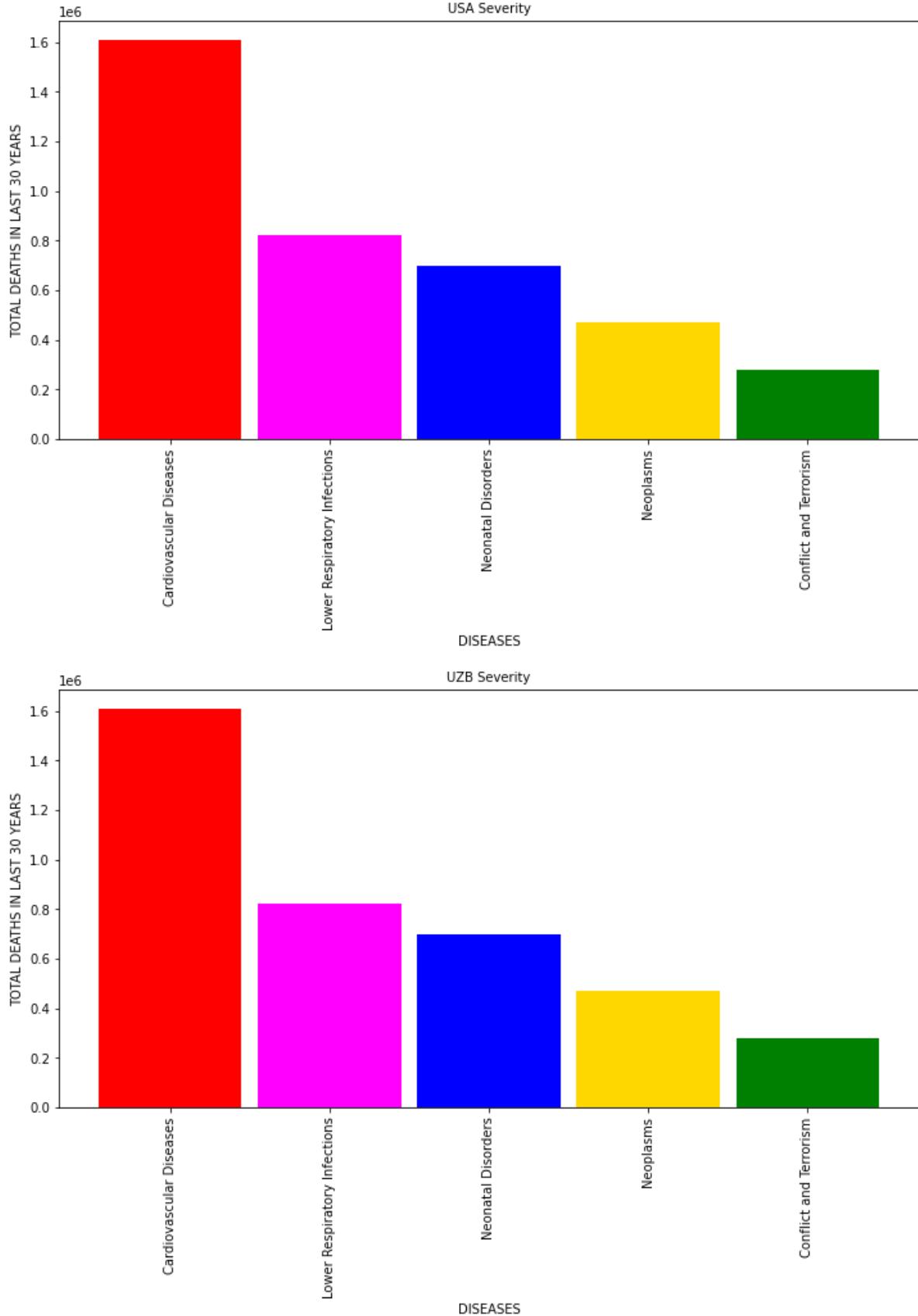


Cause Of Death



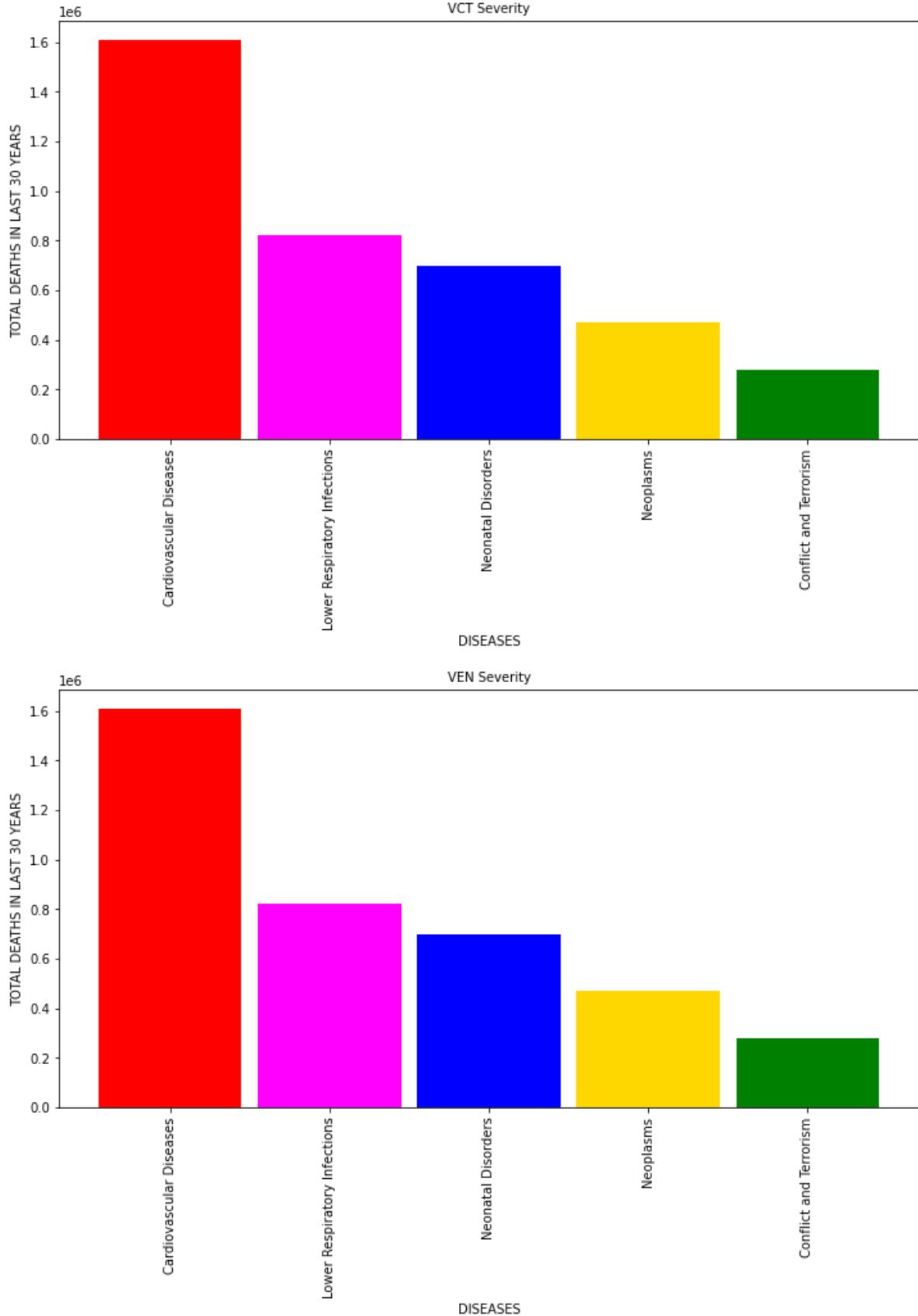


Cause Of Death



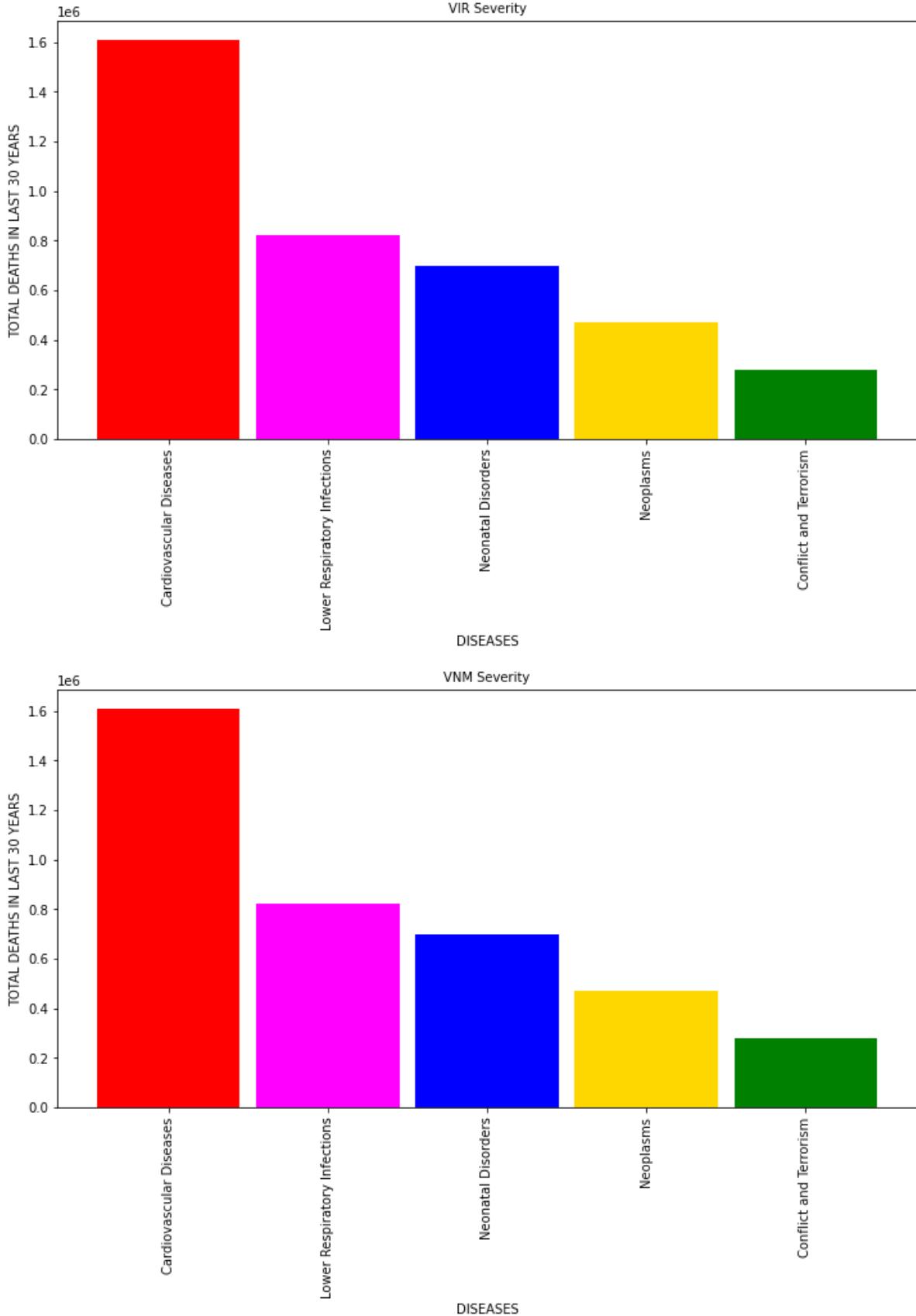
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



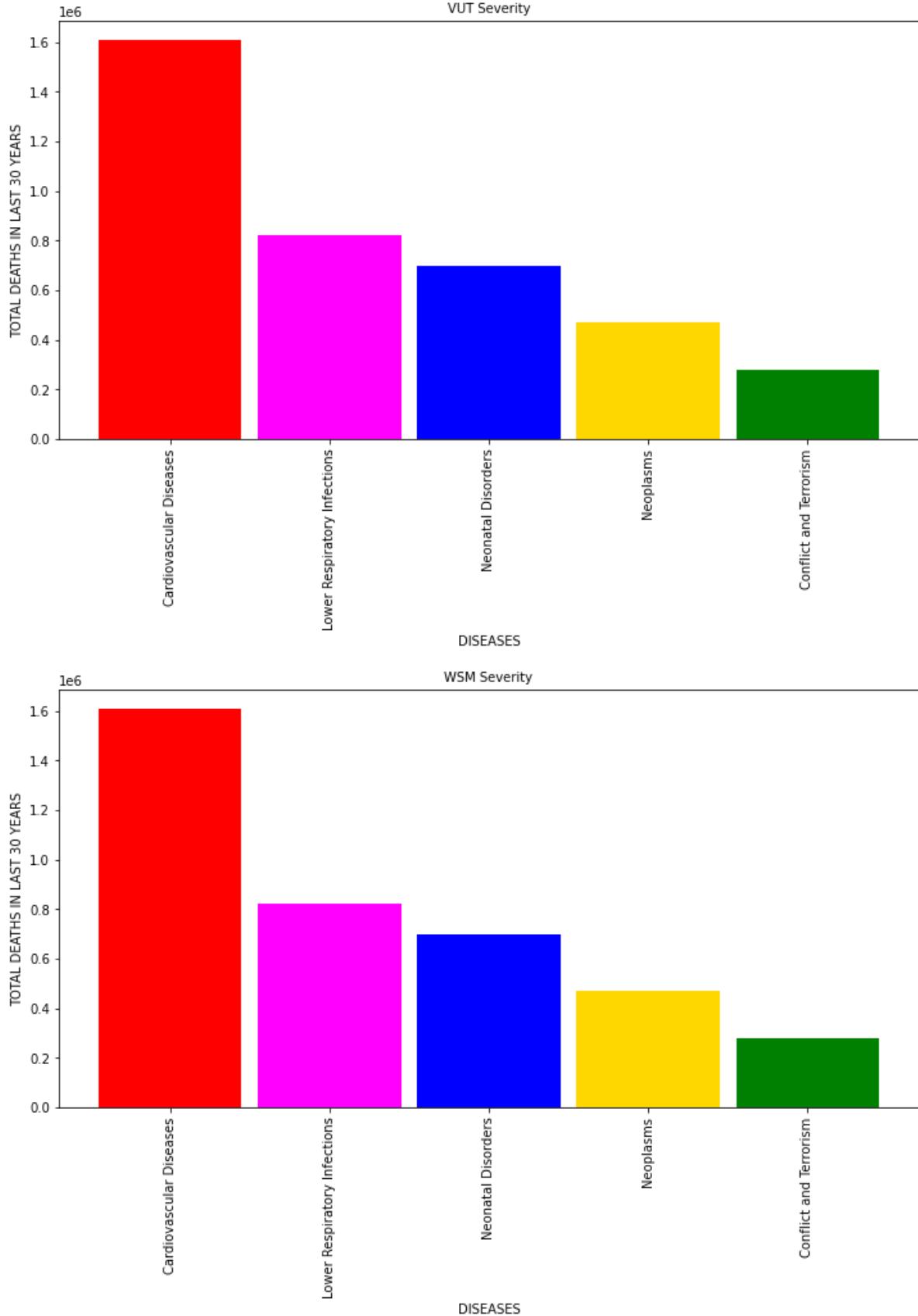
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



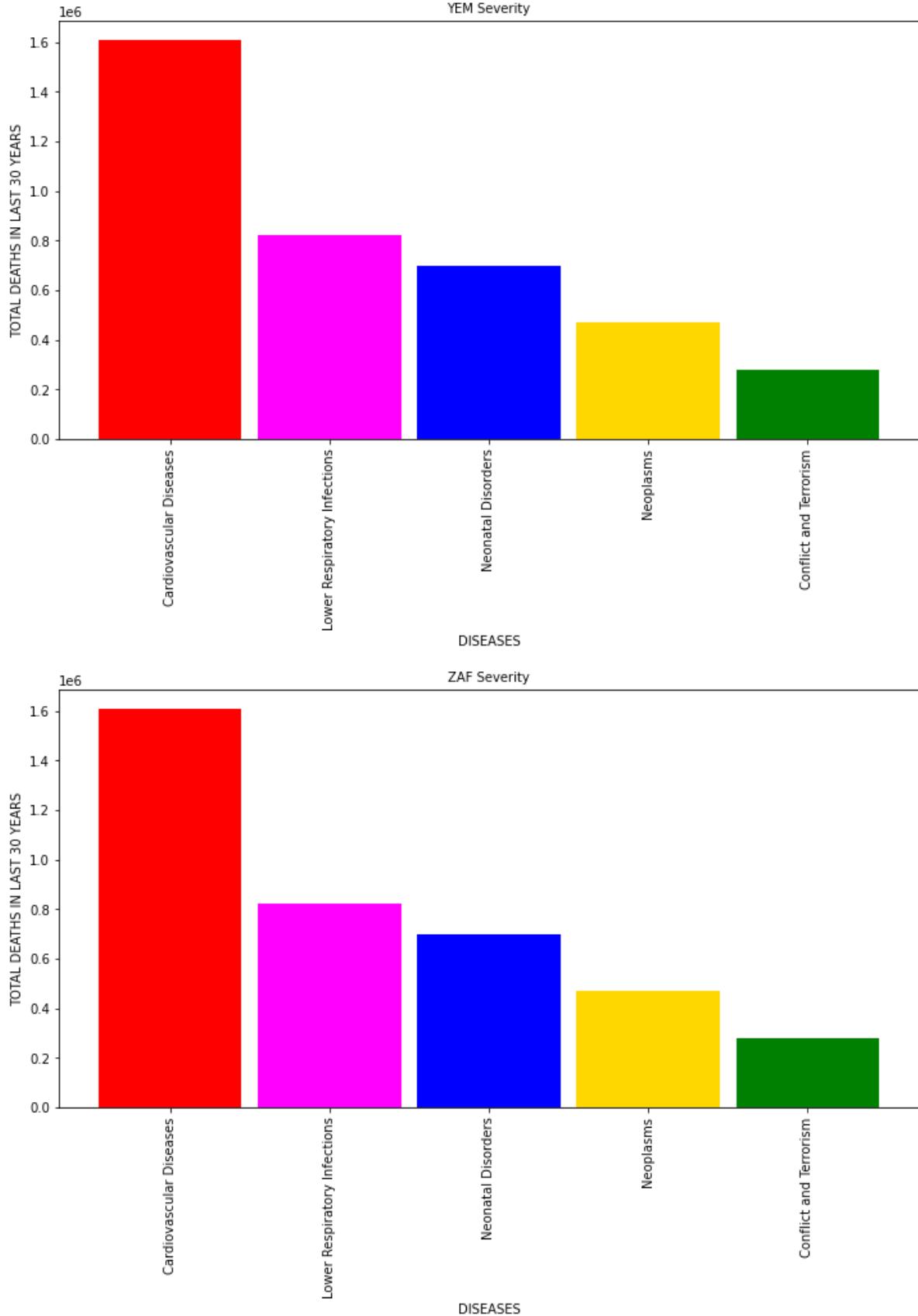
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death

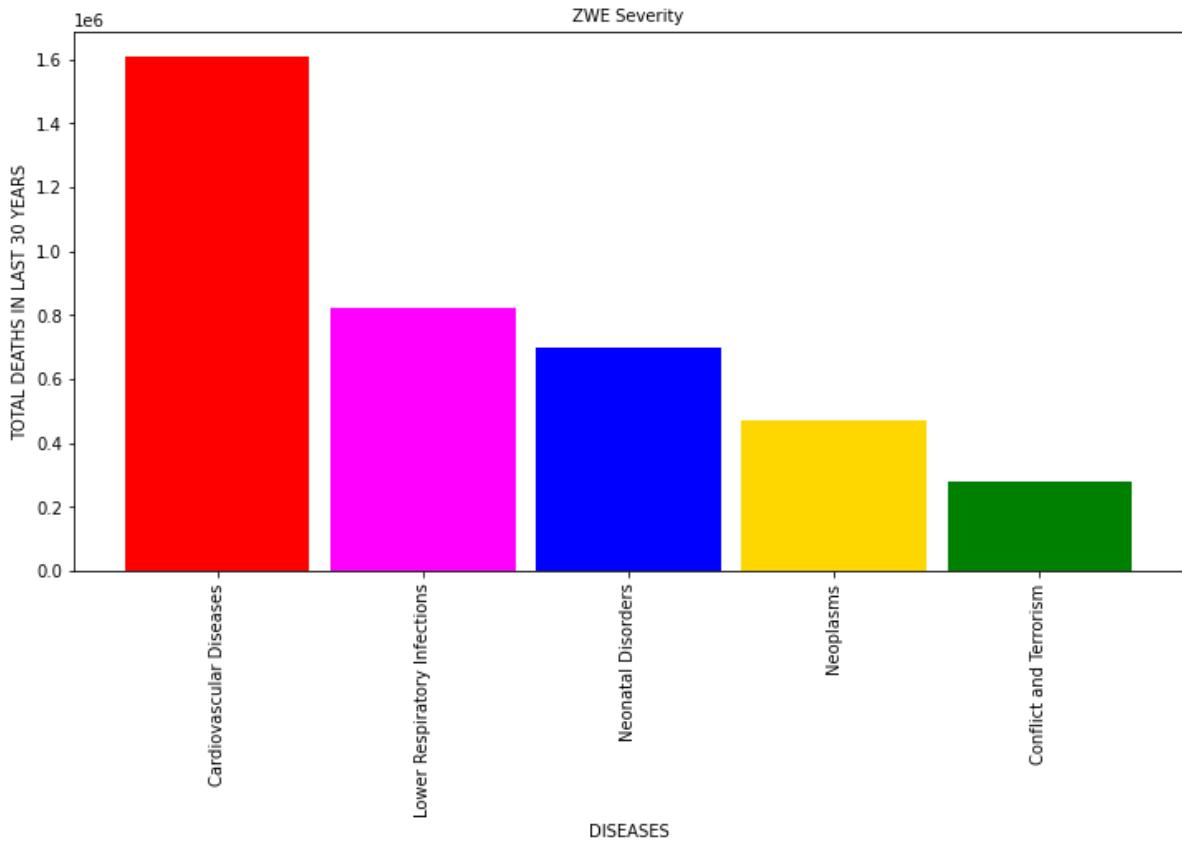
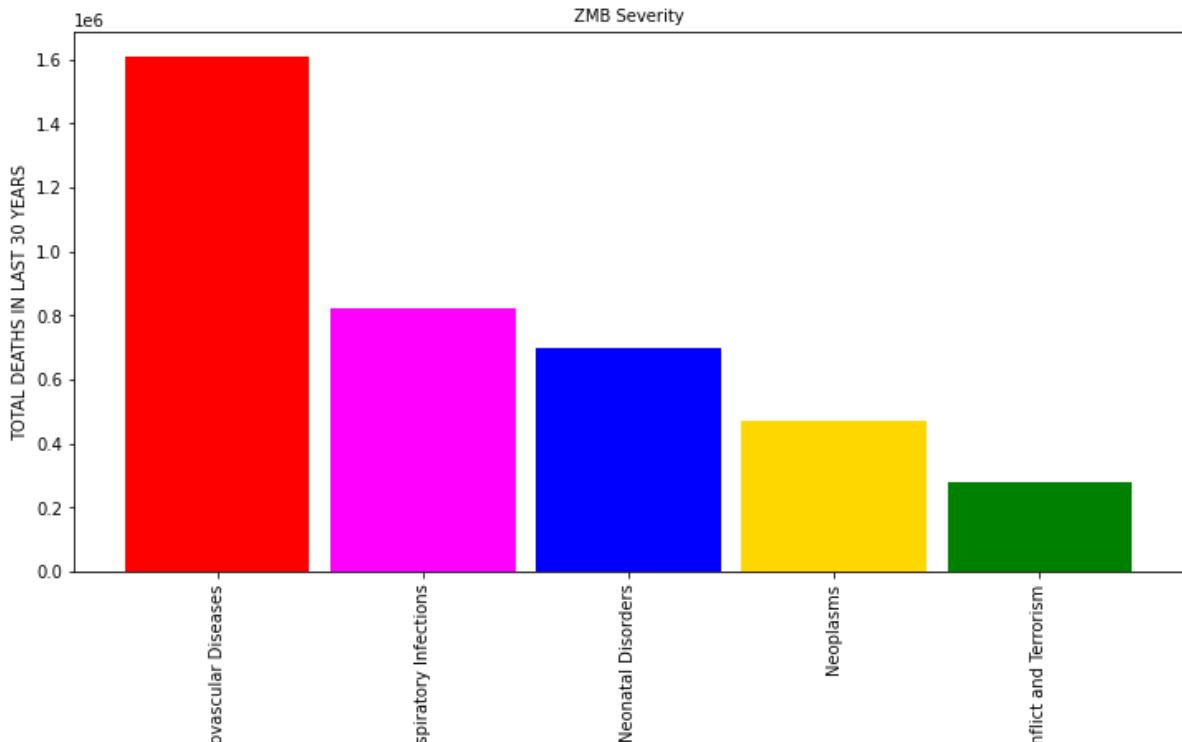


Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Cause Of Death



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



" Top 10 Countries with the highest death rates worldwide"

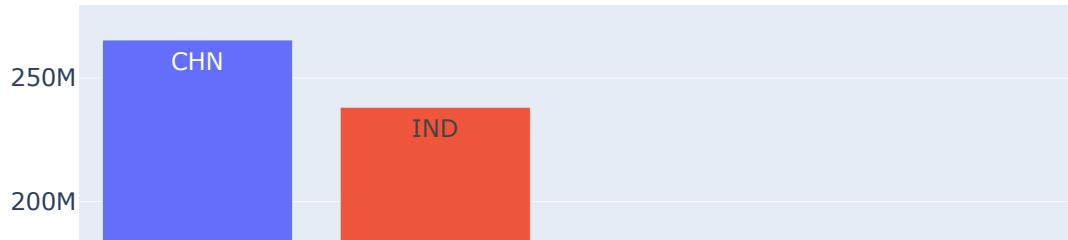
```
In [145]: df1['Total_Deaths'] = df1.sum(axis=1)
```

```
To [145]: sumall = df1[['Code', 'Total_Deaths']].sort_values('Total_Deaths', ascending = False)
```

In [147...]

```
px.bar(sumall,x = 'Code' , y = "Total_Deaths",text="Code",color = "Code",title="Co
```

Countries with the highest death rates worldwide



In [148...]

```
disease = df1.sum()[1:-1].to_frame().reset_index()
```

In [149...]

```
disease.rename(mapper={'index':'Disease',0:'Total_Deaths'},axis=1,inplace=True)
```

In [150...]

```
disease = disease.sort_values(by='Total_Deaths',ascending=False).reset_index(drop=True)
```

In [151...]

```
# Aggregating Countries with values Lesser than 23713931 into Others
disease.loc[disease.Total_Deaths < 23713931 ].sum()
disease.loc[len(disease.index)] = ['Others',98347130]
disease.drop(disease.index[16:-1],axis=0,inplace =True)
disease.reset_index(drop=True,inplace=True)
```

In [152...]

```
fig = px.pie(disease, names = 'Disease' , values = 'Total_Deaths', color_discrete_sequence=['blue','orange','red','green','purple','brown','pink','yellow','grey','lightblue','lightgreen','lightorange','lightred','lightpurple','lightbrown','lightpink','lightyellow'])
fig.update_traces(textposition='inside', textinfo='percent+label')
fig.update_layout(margin=dict(t=0, b=0, l=0, r=0))
fig.update(layout_showlegend=False)
```



Conclusions:-

CHINA , INDIA & USA has highest death rates due to

Cardiovascular,, Neoplasms ,and Respiratory are the top 3 killer diseases in the world.

In []: