CAUSE OF DEATH

Submitted by:

VINAYAK PATIL

FlipRobo SME:

KHUSHBOO GARG

ACKNOWLEDGMENT

I would like to express my special gratitude to Flip Robo Technologies team, who has given me this opportunity to deal with this dataset during my internship. It helped me to improve my analyzation skills. I want to express my gratitude to Ms. Khushboo Garg (SME, Flip Robo) as she has helped me to get out of all the difficulties I faced while doing the project. I also want to give huge thanks to entire Data Trained team.

Bibliography:

Reference used in this project:

1. Hands on Machine learning with scikit learn and tensor flow by Aurelien Geron.
2. Andrew Ng Notes on Machine Learning (GitHub).
3. Different projects on Github and Kaggle.
4. towardsdatascience, Analytics Vidya’s different papers on Data Science.
5. SCIKIT Learn Library Documentation.
6. Different conference papers on Recharchgate.
7. Demner-Fushman D, Chapman WW, McDonald CJ. What can natural language processing do for clinical decision support? J Biomed Inform.
8. Datta MW, Hernandez AM, Schlicht MJ, Kahler AJ, DeGueme AM, Dhir R, et al. Perlecan, a candidate gene for the CAPB locus, regulates prostate cancer cell growth via the Sonic Hedgehog pathway. Mol Cancer.
9. Extracting Cancer Mortality Statistics from Free-text Death Certificates | Proceedings of the 23rd Australasian Document Computing Symposium [Internet]. [cited 2020 Aug 14]. Available from: https://dl.acm.org/doi/abs/10.1145/3291992.3292003
10. Mujtaba G, Shuib L, Idris N, Hoo WL, Raj RG, Khowaja K, et al. Clinical text classification research trends: Systematic literature review and open issues. Expert Syst Appl. 2019 Feb 1;116:494–520.
11. Duarte F, Martins B, Pinto CS, Silva MJ. Deep neural models for ICD-10 coding of death certificates and autopsy reports in free-text. J Biomed Inform. 2018 Apr 1;80:64–77.
12. Mujtaba G, Shuib L, Raj RG, Rajandram R, Shaikh K. Automatic Text Classification of ICD-10 Related CoD from Complex and Free Text Forensic Autopsy Reports. In: 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA). 2016. p. 1055–8
13. Cabot C, Soualmia L, Dahamna B, Darmoni S. SIBM at CLEF eHealth Evaluation Lab 2016: Extracting Concepts in French Medical Texts with ECMT and CIMIND. In 2016.
14. Carvalho DV, Pereira EM, Cardoso JS. Machine Learning Interpretability: A Survey on Methods and Metrics. Electronics. 2019 Aug;8(8):832.
15. Berge GT, Granmo O, Tveit TO, Goodwin M, Jiao L, Matheussen BV. Using the Tsetlin Machine to Learn Human-Interpretable Rules for High-Accuracy Text Categorization With Medical Applications. IEEE Access. 2019;7:115134–46.

A Straightforward way to assess the health status of population is to focus on mortality or concepts like child mortality or life expectancy, which are based on mortality estimates. A focus on mortality, however, does not take in to account that the burden of diseases is not only that they kill people, but that they cause suffering to people who live with them. Assessing heath outcomes by both mortality and morbidity (the prevalent diseases) provides a more encompassing view on health outcomes. This is the topic of this entry. The sum of mortality and morbidity is referred to as the burden of disease and can be measured by a metric called disability adjusted life years. Daly are measuring lost health and are a standardized metric that allow for direct comparisons of disease burdens of different diseases across countries, between different population and over time. Conceptually one Daly is the equivalent of one year in good health because of either premature death of disease or disability. One Daly represents one lost year of healthy life. The first global burden of disease (GBD) was GBD 1990 and the Daly metric was prominently featured in the World Bank's 1993 World Development Report. Today it is published by both the researcher at the Institute of Health Metrics and Evaluation (IHME) and the disease burden unit at the World Health Organization (WHO), which was created in 1998. The IHME continues the work that was started in the early 1990s and publishes the Global Burden of Disease Study.

Content:

In this Dataset,we have Historical Dataof different cause of deaths for allagesaround theworld.The key featuresofthis Dataset are: Meningitis, Alzheimer's Disease and Other Dementias, Parkinson's Disease, Nutritional Deficiencies, Malaria, Drowning, Interpersonal Violence, Maternal Disorders, HIV/AIDS,DrugUseDisorders,Tuberculosis,Cardiovascular Diseases,LowerRespiratoryInfections, NeonatalDisorders, AlcoholUseDisorders,Self-harm,ExposuretoForcesof Nature,DiarrhealDiseases, EnvironmentalHeatandCold Exposure, Neoplasm’s, Conflict and Terrorism, Diabetes Mellitus, ChronicKidneyDisease,Poisonings,Protein-Energy Malnutrition, Road Injuries, Chronic Respiratory Diseases, Cirrhosis and Other Chronic Liver Diseases, Digestive Diseases,Fire,Heat,andHot Substances,AcuteHepatitis.

Importing Some of the Libraries:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

```
import plotly.express as px
import plotly.offline as pyo
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

Importing the Dataset with Display Max Columns as there are 34 Columns in the Dataset:

```
df=pd.read_csv('cause_of_deaths.csv')
pd.set_option("display.max_columns",None)
df
```

	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	Tuber
0	Afghanistan	AFG	1990	2159	1116	371	2087	93	1370	1538	2655	34	93	
1	Afghanistan	AFG	1991	2218	1136	374	2153	189	1391	2001	2885	41	102	
2	Afghanistan	AFG	1992	2475	1162	378	2441	239	1514	2299	3315	48	118	
3	Afghanistan	AFG	1993	2812	1187	384	2837	108	1687	2589	3671	56	132	
4	Afghanistan	AFG	1994	3027	1211	391	3081	211	1809	2849	3863	63	142	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
6115	Zimbabwe	ZWE	2015	1439	754	215	3019	2518	770	1302	1355	29162	104	
6116	Zimbabwe	ZWE	2016	1457	767	219	3056	2050	801	1342	1338	27141	110	
6117	Zimbabwe	ZWE	2017	1460	781	223	2990	2116	818	1363	1312	24846	115	
6118	Zimbabwe	ZWE	2018	1450	795	227	2918	2088	825	1396	1294	22106	121	
6119	Zimbabwe	ZWE	2019	1450	812	232	2884	2068	827	1434	1294	20722	127	

6120 rows × 34 columns

Doing some shuffling of the dataset to see any Abnormal Values Present in the Dataset:-

```
df.tail()
```

	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	Tuber
6115	Zimbabwe	ZWE	2015	1439	754	215	3019	2518	770	1302	1355	29162	104	
6116	Zimbabwe	ZWE	2016	1457	767	219	3056	2050	801	1342	1338	27141	110	
6117	Zimbabwe	ZWE	2017	1460	781	223	2990	2116	818	1363	1312	24846	115	
6118	Zimbabwe	ZWE	2018	1450	795	227	2918	2088	825	1396	1294	22106	121	
6119	Zimbabwe	ZWE	2019	1450	812	232	2884	2068	827	1434	1294	20722	127	

```
df.sample(5)
```

	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	Tuber
1275	Costa Rica	CRI	2005	43	660	120	17	0	137	299	26	150	12	
2396	Hungary	HUN	2016	53	4538	891	34	0	137	143	11	32	51	
1573	Dominican Republic	DOM	2003	299	1148	231	441	16	252	1242	196	3686	15	
3784	New Zealand	NZL	1994	28	919	188	11	0	67	70	7	54	32	
4482	Russia	RUS	2002	2380	27337	7642	635	0	18248	56916	533	9450	9596	

Checking out the Data Types of the Columns in the Dataset:

```
# Now lets identify which types of data types do they all belongs

df.dtypes
```

```
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
```

Dataset contains both categorical columns and numerical columns.. There are only 2 numerical columns in whole dataset

Here we can see that there are 2 object columns and rest all the other columns are Numerical columns.

Let's check the info of the Dataset and here we get to know about the data type and counts of the column:

```
: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6120 entries, 0 to 6119
Data columns (total 34 columns):
 #   Column                                 Non-Null Count  Dtype
---  ------                                 --------------  -----
 0   Country/Territory                      6120 non-null   object
 1   Code                                   6120 non-null   object
 2   Year                                   6120 non-null   int64
 3   Meningitis                             6120 non-null   int64
 4   Alzheimer's Disease and Other Dementias 6120 non-null   int64
 5   Parkinson's Disease                    6120 non-null   int64
 6   Nutritional Deficiencies               6120 non-null   int64
 7   Malaria                                6120 non-null   int64
 8   Drowning                               6120 non-null   int64
 9   Interpersonal Violence                 6120 non-null   int64
 10  Maternal Disorders                     6120 non-null   int64
 11  HIV/AIDS                               6120 non-null   int64
 12  Drug Use Disorders                     6120 non-null   int64
 13  Tuberculosis                           6120 non-null   int64
 14  Cardiovascular Diseases                6120 non-null   int64
 15  Lower Respiratory Infections           6120 non-null   int64
 16  Neonatal Disorders                     6120 non-null   int64
 17  Alcohol Use Disorders                  6120 non-null   int64
 18  Self-harm                              6120 non-null   int64
 19  Exposure to Forces of Nature           6120 non-null   int64
 20  Diarrheal Diseases                     6120 non-null   int64
 21  Environmental Heat and Cold Exposure   6120 non-null   int64
 22  Neoplasms                              6120 non-null   int64
 23  Conflict and Terrorism                 6120 non-null   int64
 24  Diabetes Mellitus                      6120 non-null   int64
 25  Chronic Kidney Disease                 6120 non-null   int64
 26  Poisonings                             6120 non-null   int64
 27  Protein-Energy Malnutrition            6120 non-null   int64
 28  Road Injuries                          6120 non-null   int64
 29  Chronic Respiratory Diseases           6120 non-null   int64
 30  Cirrhosis and Other Chronic Liver Diseases 6120 non-null int64
 31  Digestive Diseases                     6120 non-null   int64
 32  Fire, Heat, and Hot Substances         6120 non-null   int64
 33  Acute Hepatitis                        6120 non-null   int64
dtypes: int64(32), object(2)
memory usage: 1.6+ MB
```

This tell us about columns name null value dtypes of columns and memory usage.. count of every column are equal which means there are no nan present in dataset..it tell dtype of every column and tere are two data type in dataset int64, object where 32 columns are int64 where as 2 column are object..

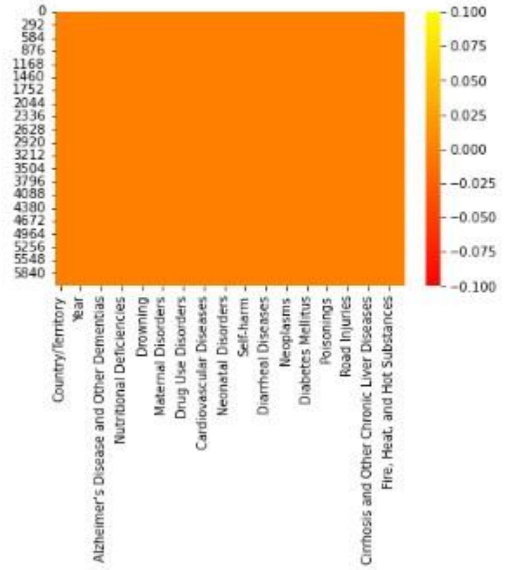
Let’s check null values in Dataset:

```
df.isnull().sum()
```

```
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
```

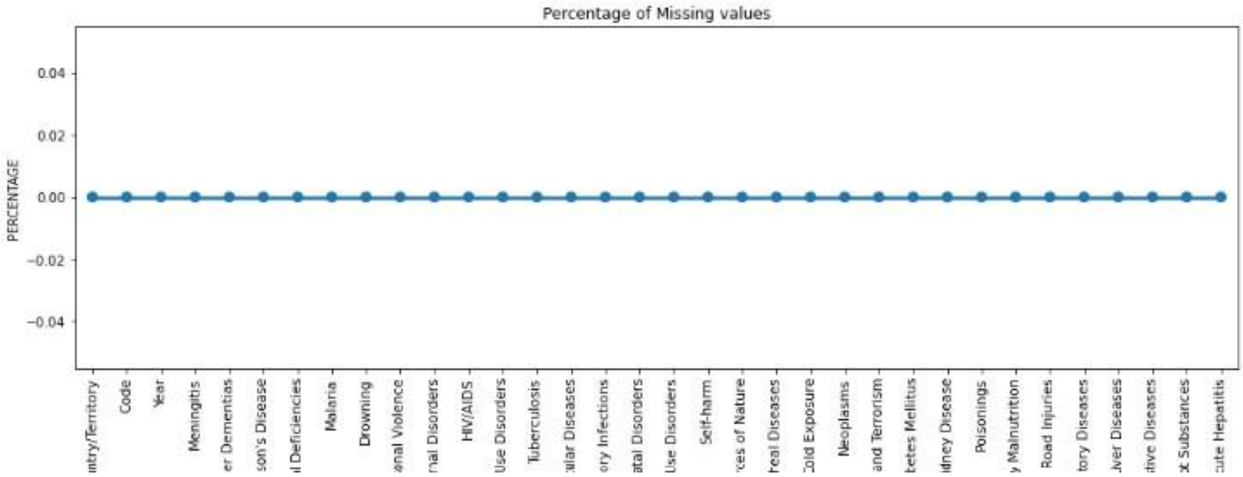
Count of nan is 0 in every column

```
# Let's visualize NaN values
sns.heatmap(df.isnull(),cmap="autumn")
plt.show()
```



dataset is free from nan value

```
missing = pd.DataFrame((df.isnull().sum())*100/df.shape[0]).reset_index()
plt.figure(figsize=(16,5))
ax = sns.pointplot('index',0,data=missing)
plt.xticks(rotation =90,fontsize =11)
plt.title("Percentage of Missing values")
plt.ylabel("PERCENTAGE")
plt.show()
```



Here we can see 0 NaN values present in dataset

Separating categorical and numerical columns from the dataset.

Separating numerical and categorcal columns

```
# Checking for categorical columns
categorical_col=[]
for i in df.dtypes.index:
    if df.dtypes[i]=='object':
        categorical_col.append(i)
print("Categorical columns are:\n",categorical_col)
```

```
Categorical columns are:
 ['Country/Territory', 'Code']
```

These two columns are only categorical in dataset

```
# Now checking for numerical columns
numerical_col=[]
for i in df.dtypes.index:
    if df.dtypes[i]!='object':
        numerical_col.append(i)
print("Numerical columns are:\n",numerical_col)
```

```
Numerical columns are:
 ['Year', 'Meningitis', "Alzheimer's Disease and Other Dementias", "Parkinson's Disease", 'Nutritional Deficiencies', 'Malari
a', 'Drowning', 'Interpersonal Violence', 'Maternal Disorders', 'HIV/AIDS', 'Drug Use Disorders', 'Tuberculosis', 'Cardiovascul
ar Diseases', 'Lower Respiratory Infections', 'Neonatal Disorders', 'Alcohol Use Disorders', 'Self-harm', 'Exposure to Forces o
f Nature', 'Diarrheal Diseases', 'Environmental Heat and Cold Exposure', 'Neoplasms', 'Conflict and Terrorism', 'Diabetes Melli
tus', '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']
```

These are numerical column of dataset

As told above we know that we have 2 categorical columns out if 34 columns and rest all 32 columns are numerical columns.

Let’s described the dataset:

```
df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Year	6120.0	2004.500000	8.656149	1990.0	1997.00	2004.5	2012.00	2019.0
Meningitis	6120.0	1719.701307	6672.006930	0.0	15.00	109.0	847.25	98358.0
Alzheimer's Disease and Other Dementias	6120.0	4864.189379	18220.659072	0.0	90.00	666.5	2456.25	320715.0
Parkinson's Disease	6120.0	1173.169118	4616.156238	0.0	27.00	164.0	609.25	76990.0
Nutritional Deficiencies	6120.0	2253.600000	10483.633601	0.0	9.00	119.0	1167.25	268223.0
Malaria	6120.0	4140.960131	18427.753137	0.0	0.00	0.0	393.00	280604.0
Drowning	6120.0	1683.333170	8877.018366	0.0	34.00	177.0	698.00	153773.0
Interpersonal Violence	6120.0	2083.797222	6917.006075	0.0	40.00	265.0	877.00	69640.0
Maternal Disorders	6120.0	1262.589216	6057.973183	0.0	5.00	54.0	734.00	107929.0
HIV/AIDS	6120.0	5941.898529	21011.962487	0.0	11.00	136.0	1879.00	305491.0
Drug Use Disorders	6120.0	434.006699	2898.761628	0.0	3.00	20.0	129.00	65717.0
Tuberculosis	6120.0	7491.928595	39549.977578	0.0	35.00	417.0	2924.25	657515.0
Cardiovascular Diseases	6120.0	73160.454575	291577.537794	4.0	2028.00	11742.0	42546.50	4584273.0
Lower Respiratory Infections	6120.0	13687.914706	48031.720009	0.0	345.00	2126.5	10161.25	690913.0
Neonatal Disorders	6120.0	12558.942647	56058.366412	0.0	131.00	916.0	7419.75	852761.0
Alcohol Use Disorders	6120.0	787.421242	3545.823616	0.0	9.00	80.0	316.00	55200.0
Self-harm	6120.0	3874.825327	18425.616418	0.0	94.00	533.0	1882.25	220357.0
Exposure to Forces of Nature	6120.0	243.485621	4717.104377	0.0	0.00	0.0	12.00	222641.0
Diarrheal Diseases	6120.0	10822.795425	65416.174485	0.0	20.00	296.5	3946.75	1119477.0
Environmental Heat and Cold Exposure	6120.0	292.295915	1704.466356	0.0	2.00	21.0	109.00	29048.0
Neoplasms	6120.0	37542.244771	161558.365445	1.0	809.75	5629.5	20147.75	2716551.0
Conflict and Terrorism	6120.0	538.243954	7033.308187	0.0	0.00	0.0	23.00	503532.0
Diabetes Mellitus	6120.0	5138.704575	16773.081040	1.0	236.00	1087.0	2954.00	273089.0
Chronic Kidney Disease	6120.0	4724.132680	16470.429969	0.0	145.75	822.0	2922.50	222922.0
Poisonings	6120.0	425.013399	2022.640521	0.0	6.00	52.5	254.00	30883.0
Protein-Energy Malnutrition	6120.0	1965.994281	8255.999063	0.0	5.00	92.0	1042.50	202241.0
Road Injuries	6120.0	5930.795588	24097.784291	0.0	174.75	966.5	3435.25	329237.0
Chronic Respiratory Diseases	6120.0	17092.374837	105157.179839	1.0	289.00	1689.0	5249.75	1366039.0
Cirrhosis and Other Chronic Liver Diseases	6120.0	6124.072059	20688.118580	0.0	154.00	1210.0	3547.25	270037.0
Digestive Diseases	6120.0	10725.267157	37228.051096	0.0	284.00	2185.0	6080.00	464914.0
Fire, Heat, and Hot Substances	6120.0	588.711438	2128.595120	0.0	17.00	126.0	450.00	25876.0
Acute Hepatitis	6120.0	618.429902	4186.023497	0.0	2.00	15.0	160.00	64305.0

Here we have described the whole dataset by DESCRIBE command.

1. We can see the count of all the columns that is 6120which means no Null value is present in the dataset.

2. We can see the mean and standard deviation of all theNumeric columns in the dataset.

3. We can see the Min and Max from all the columns.
4. We can see Quartiles over here too

VISUALISATION:

Now let’s divide all the factors of Death into 4 Categories:

These 4 Categories are:

1. Death by Diseases.
2. Death by Environment and Accident.
3. Death by Crime, Terror, Self-harm and Accident.
4. Death by Chronic Diseases.

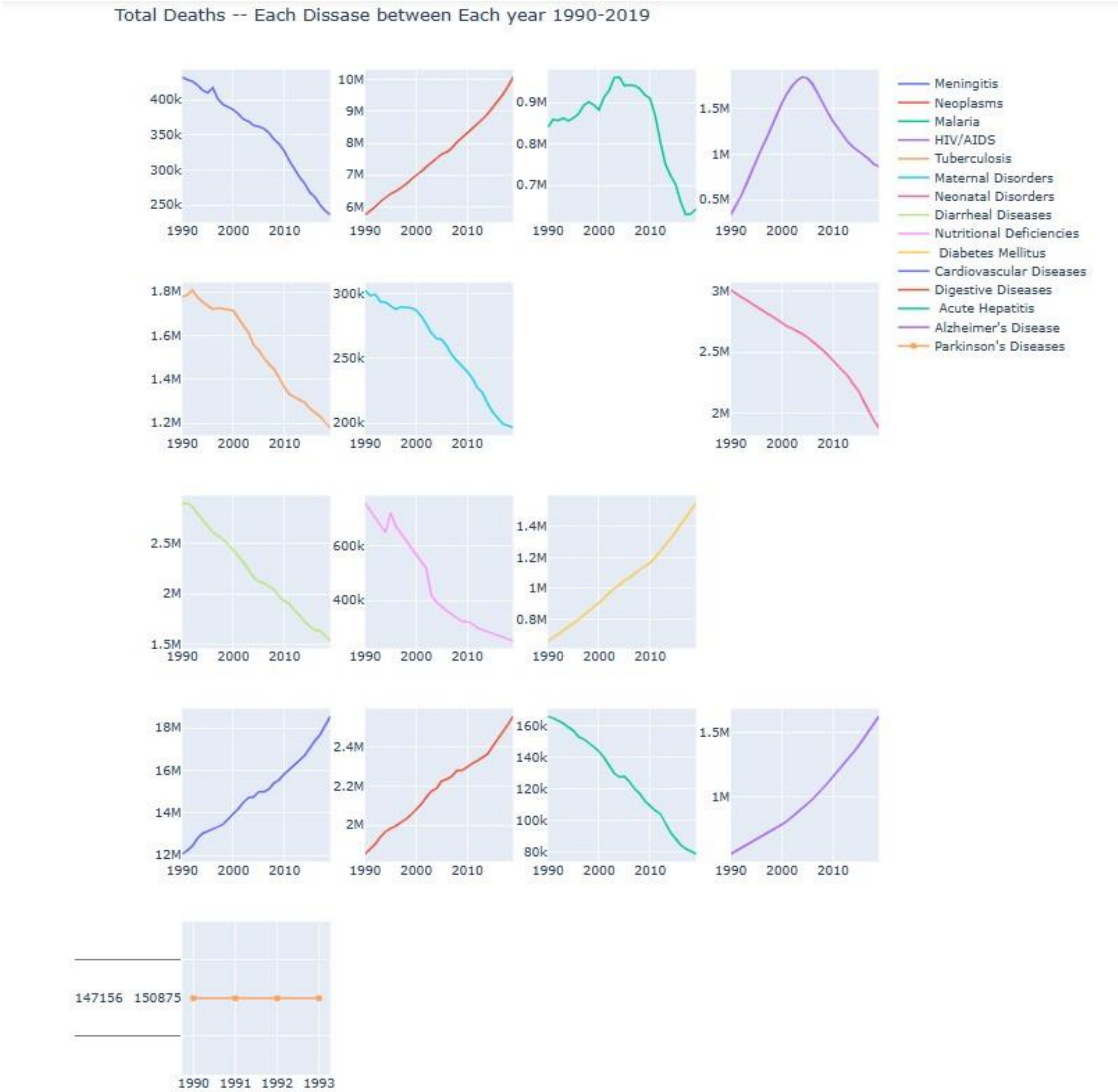
Now do the Analysis as per the death by Diseases:

Here I have done grouping of year and countries on the basis of Diseases.

```
groupingByYear = deathsBy_Disease.groupby(['Year'])[[
                         "Meningitis",
                         "Alzheimer's Disease and Other Dementias",
                         "Parkinson's Disease",
                         "Digestive Diseases",
                         "Malaria",
                         "Tuberculosis",
                         "Diabetes Mellitus",
                         "HIV/AIDS",
                         "Acute Hepatitis",
                        "Parkinson's Disease",
                       "Nutritional Deficiencies",
                      "Cardiovascular Diseases",
                      "Neoplasms",
                      "Neonatal Disorders",
                      "Maternal Disorders",
                      "Diarrheal Diseases",]].sum().reset_index()

groupingByCountries = deathsBy_Disease.groupby(['Country/Territory'])[[
                         "Meningitis",
                         "Alzheimer's Disease and Other Dementias",
                         "Parkinson's Disease",
                         "Digestive Diseases",
                         "Malaria",
                         "Tuberculosis",
                         "Diabetes Mellitus",
                         "HIV/AIDS",
                         "Acute Hepatitis",
                        "Parkinson's Disease",
                       "Nutritional Deficiencies",
                      "Cardiovascular Diseases",
                      "Neoplasms",
                      "Neonatal Disorders",
                      "Maternal Disorders",
                      "Diarrheal Diseases",]].sum().reset_index()
```

Now do plotting of death by Diseases:

```
fig = make_subplots(rows=5, cols=4)

fig.add_trace(go.Scatter(x=groupingByYear['Year'], y=groupingByYear['Meningitis'], name = 'Meningitis'),row=1, col=1)
fig.add_trace(go.Scatter(x=groupingByYear['Year'], y=groupingByYear['Neoplasms'], name = 'Neoplasms'),row=1, col=2)
fig.add_trace(go.Scatter(x=groupingByYear['Year'], y=groupingByYear['Malaria'],name='Malaria'),row=1, col=3)
fig.add_trace(go.Scatter(x=groupingByYear['Year'], y=groupingByYear['HIV/AIDS'],name='HIV/AIDS'),row=1, col=4)
fig.add_trace(go.Scatter(x=groupingByYear['Year'], y=groupingByYear['Tuberculosis'],name='Tuberculosis'),row=2, col=1)
fig.add_trace(go.Scatter(x=groupingByYear['Year'], y=groupingByYear['Maternal Disorders'],name='Maternal Disorders'),row=2, col=
fig.add_trace(go.Scatter(x=groupingByYear['Year'], y=groupingByYear['Neonatal Disorders'],name='Neonatal Disorders'),row=2, col=
fig.add_trace(go.Scatter(x=groupingByYear['Year'], y=groupingByYear['Diarrheal Diseases'],name='Diarrheal Diseases'),row=3, col=
fig.add_trace(go.Scatter(x=groupingByYear['Year'], y=groupingByYear['Nutritional Deficiencies'],name='Nutritional Deficiencies'),
fig.add_trace(go.Scatter(x=groupingByYear['Year'], y=groupingByYear['Diabetes Mellitus'],name=' Diabetes Mellitus'),row=3, col=3)
fig.add_trace(go.Scatter(x=groupingByYear['Year'], y=groupingByYear['Cardiovascular Diseases'],name='Cardiovascular Diseases'),ro
fig.add_trace(go.Scatter(x=groupingByYear['Year'], y=groupingByYear['Digestive Diseases'],name='Digestive Diseases'),row=4, col=
fig.add_trace(go.Scatter(x=groupingByYear['Year'], y=groupingByYear['Acute Hepatitis'],name=' Acute Hepatitis'),row=4, col=3)
fig.add_trace(go.Scatter(x=groupingByYear['Year'], y=groupingByYear["Alzheimer's Disease and Other Dementias"],name="Alzheimer's
fig.add_trace(go.Scatter(x=groupingByYear['Year'], y=groupingByYear["Parkinson's Disease"],name="Parkinson's Diseases"),row=5, co


fig.update_layout(height=1200, width=1000, title_text="Total Deaths -- Each Dissase between Each year 1990-2019")
fig.show()
```



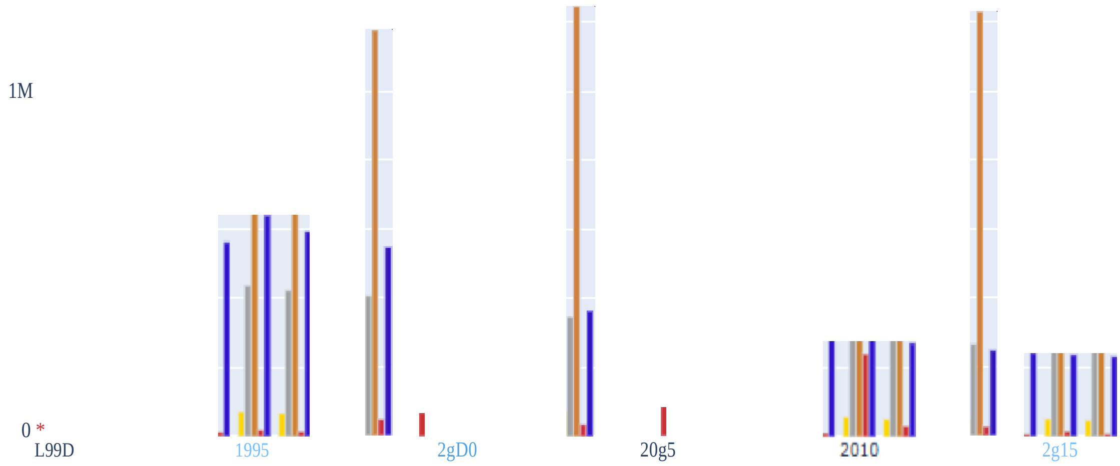
This is Plot shows how much Death has taken places by Diseases in all the year since 1990-2019.

Now Let’s See Deaths Taken Place By Environment and Accident.

I HAVE DONE GROUPBY OF ALL THE DEATH ACCORDING TO YEARWHICHFALLSUNDERTHIS CATEGORY.

```
tracel = go.Bar(
    x=deathsBy Environment Vd_Nature_group Year['Year'],
    y=deathsBy_Environment_Vd_Nature_group Year['Environmental Hea{ and cold Exposure'],
    name = 'Deaths - Enviornmental heat and cold exposure',
    marker=dict(color='éFFD70B')}

tracez = go.Bar{
    x=deathsBy_Env1ronnent_AfId_Nature_gnoup Year[ 'Year' ] ,
    y=deathsBy_Env1ronnent_AfId_Nature_gnoup Year[ 'Dro\'ining' ] ,
    name= 'reaths - Droi'ining' ,
    marker=d1ct color= ' ésEA0A1 ' ))

trace3 = go.Bar{
    x=deathsBy Env1raneent Afld_uature_group Year[ 'Year' ] , y=deathsBy
    Env1ronrent AfId_8ature_group Year['Road lnjur1es'], nafnz= 'Death s - Road
    inj ur ies ' ,
    frarker=d1ct{color= 'éCD7F32' ))

trace4 = go.Bar{
    x=death sBy_Env1ronnent_AfId_Nature coup Year[ ' Year' ] ,
    y=deathsBy_Env1ronnent_AfId_Nature_gnoup Year['Exposure to Forces of liature'], nafnz=
    'Exposure to forces of nature',
    marker=d1ct{color= ' d D2F 32 ' ))

traces = go.Bar{
    x=death sBy_Env1ronnent_AfId_Nature_gnoup Year[ ' Year' ] ,
    y=deathsBy Env1raneent Afld_uature_group Year[ ' Protein-Energ,' riaJnutrit‹on ' ], name=
    'rx°aths - PEf1 ' ,
    frarker=d1ct{color= 'ézf12cd ' ))

data = [trace1, trace2, trace 3, trace4-, traces] layout =
go. Layout(
    tzt1e=' 19se to ze19 Oeaths - En •ironnent Dr Nature',hezght = s00, x1dth=14-ee

fog = go . Fzgure{data=data, 1ayout=1ayout) fig.
show()
```

1990 to 2019 Deaths - Environment or Nature



Here following color is representing following columns :-

1. Yellow:- Environmental Heat and Cold Exposure

2. Grey :- Deaths - Drowning

3. Orange :- Road Injuries

4. Blue :- Protein-Energy Malnutrition(PEM)

This plot shows the total number of deaths caused by Environment_And_Accidental in year 1990 TO 2019. Here we can notice the least and the max death that took place in all the 4 categories in all the given year.

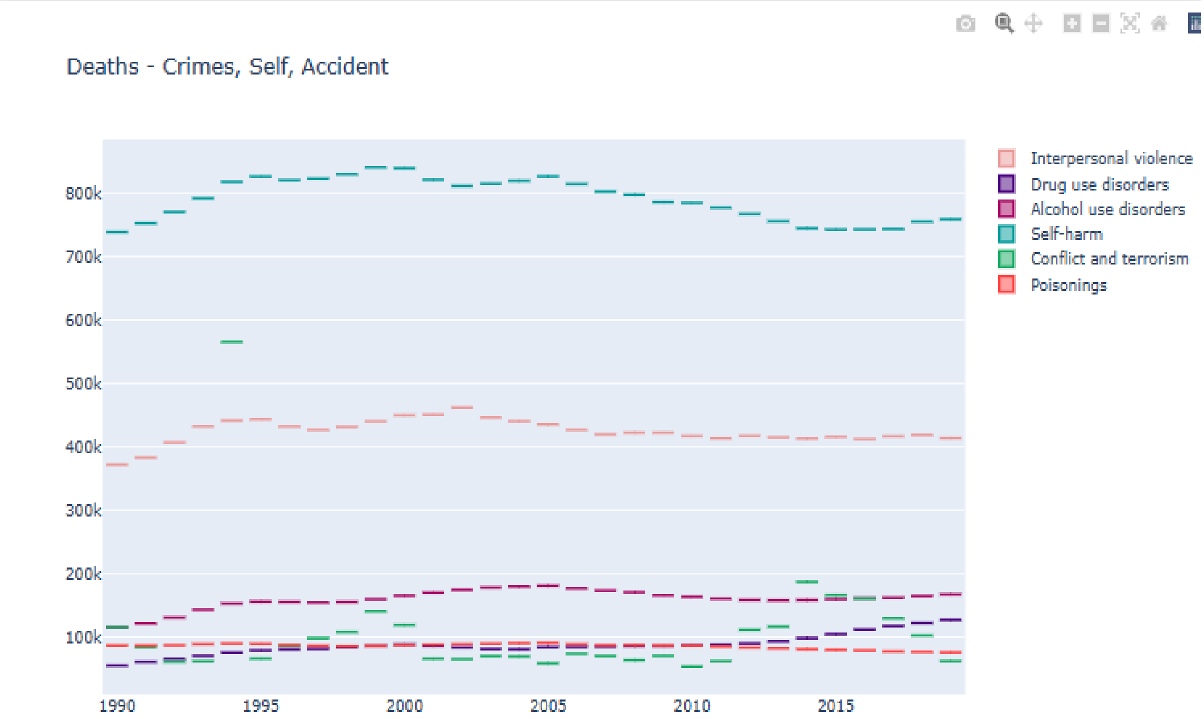
Death by Crime, Terror, Self-Harm and Accident.

```
groupingCrimesTerrorAccidentSelf = deathsBy_Crimes_Terror_Accident_SelfHarm.groupby('Year')[['Interpersonal Violence',
                                                      'Drug Use Disorders',
                                                      'Alcohol Use Disorders',
                                                      'Self-harm',
                                                      'Conflict and Terrorism',
                                                      'Poisonings'
                                                      ]].sum().reset_index()
```

```
groupingCrimesTerrorAccidentSelf.head()
```

	Year	Interpersonal Violence	Drug Use Disorders	Alcohol Use Disorders	Self-harm	Conflict and Terrorism	Poisonings
0	1990	372497	56133	116390	738804	116286	87951
1	1991	383689	61890	122478	752575	85017	87813
2	1992	407176	66826	131665	770286	62063	88435
3	1993	432858	71603	143901	791904	62733	90036
4	1994	441971	76717	153859	817682	566082	90897

```
fig = go.Figure()

fig.add_trace(go.Violin(x= groupingCrimesTerrorAccidentSelf['Year'] ,
                        y= groupingCrimesTerrorAccidentSelf['Interpersonal Violence'],
                        name='Interpersonal violence',
                        line_color='#ea9999'))

fig.add_trace(go.Violin(x= groupingCrimesTerrorAccidentSelf['Year'] ,
                        y= groupingCrimesTerrorAccidentSelf['Drug Use Disorders'],
                        name='Drug use disorders',
                        line_color='#48007c'))

fig.add_trace(go.Violin(x= groupingCrimesTerrorAccidentSelf['Year'] ,
                        y= groupingCrimesTerrorAccidentSelf['Alcohol Use Disorders'],
                        name='Alcohol use disorders',
                        line_color='#a60661'))

fig.add_trace(go.Violin(x= groupingCrimesTerrorAccidentSelf['Year'] ,
                        y= groupingCrimesTerrorAccidentSelf['Self-harm'],
                        name='Self-harm ',
                        line_color='#009999'))

fig.add_trace(go.Violin(x= groupingCrimesTerrorAccidentSelf['Year'] ,
                        y= groupingCrimesTerrorAccidentSelf['Conflict and Terrorism'],
                        name='Conflict and terrorism',
                        line_color='#15a962'))

fig.add_trace(go.Violin(x= groupingCrimesTerrorAccidentSelf['Year'] ,
                        y= groupingCrimesTerrorAccidentSelf['Poisonings'],
                        name='Poisonings',
                        line_color='#ff4040'))



fig.update_traces(meanline_visible=True)
fig.update_layout(title_text='Deaths - Crimes, Self, Accident',violingap=0, violinmode='overlay',height=600,width=1000)
fig.show()
```



We can clearly see in this plot which shows...

CRIMES_TERROR_ACCIDENT_SELF-HARM and here
Come to know that in all the years the maximum death have been taken place by Conflicts and Terrorism and the max death was in between 1990 and 2000.

Poisoning seems to be constant in all the years.

The second highest death has taken place by Interpersonal violence

And rest all the case of seems to be under 200k in all the given years.

Death by Chronic Diseases

Now do grouping ofchronic diseases as per year and relevantdiseases.

DEATH BY CRONIC DISEASES

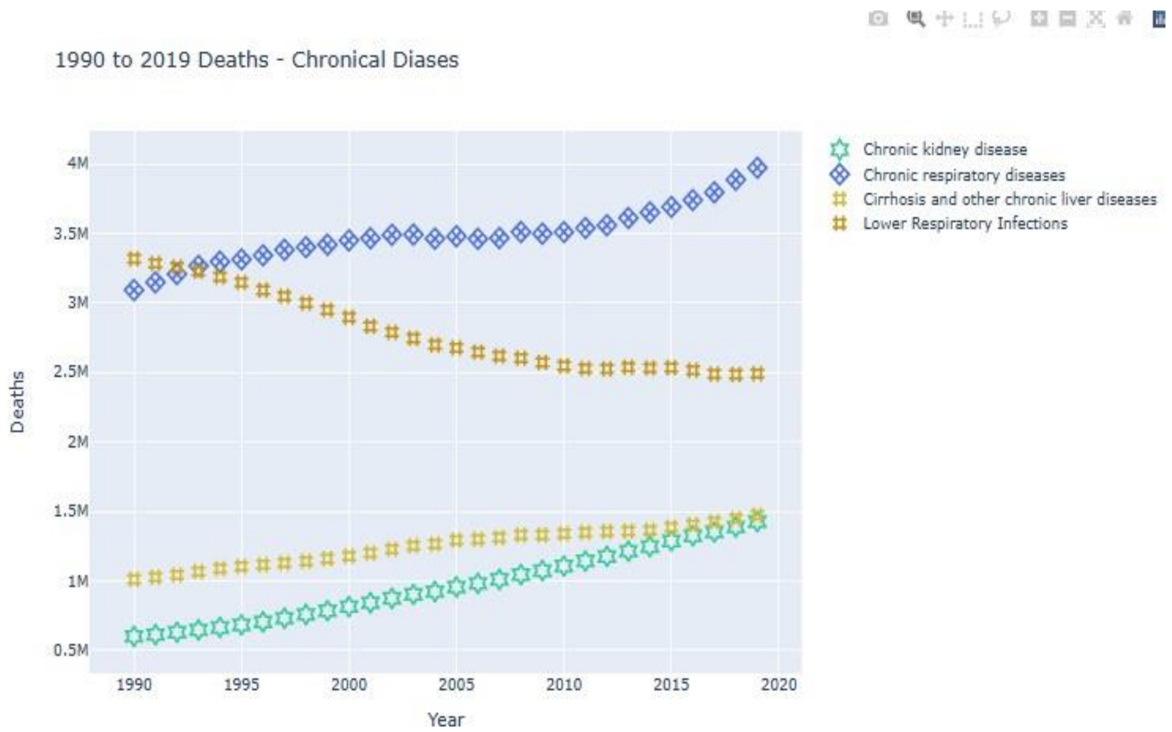
In [40]:
```
chronic_Deaths_GroupingByYear = deathsBy_Chronic_Disases.groupby('Year')[['Chronic Kidney Disease',
                                                      'Chronic Respiratory Diseases',
                                                      'Cirrhosis and Other Chronic Liver Diseases','Lower Respiratory Infections']].sum().re
```

In [41]:
```
chronic_Deaths_GroupingByYear.head()
```

Out[41]:

	Year	Chronic Kidney Disease	Chronic Respiratory Diseases	Cirrhosis and Other Chronic Liver Diseases	Lower Respiratory Infections
0	1990	600925	3092759	1012423	3318264
1	1991	613589	3148288	1026870	3282941
2	1992	630160	3207816	1042953	3258792
3	1993	647255	3266812	1067730	3226972
4	1994	665365	3297292	1089331	3187285

```
traceB = gp.Scatter(
    x = chronic_Deaths_0roupingByvear 'Tear'],
    y = chrori1c_Deaths_GroupiugByYear[ 'Chroulc c1dfx°y DiseB se ' ],
    naee = ' Chroñ1c kIdt+ey di sease, '
    be    = ' markers',
    marter = dict(
        size = 12,
        color = 'rgb(51,284,k53)',
        syEiol = 'hexagram-open ,
        line =dict(width =2)))

trace1 = go. Scatter(
    x = chronic_Deaths_&roupingBy¥ear['Tear' l,
    y = chronic_Deaths_&roupingBy¥ear['Chronic respiratory Diseases'],
    na*e = 'Chronic respiratory diseases',
    mode = 'markers',
    acer = dict(
        six = 12,
        c0l0F - 'rgb(77,113,222)',
        sydol = 'diamond-x-qpen',
        line = difi(wiât =2)))

tracez = go.scatter(
    x = chronic Deaths sroupingBy¥ear['Tear'],
    y=  chmnicDeRhs_sxp     ear[cir#o1saMotñer cñroni Lier oixsses]
    nae = 'cirrhosis and other chronic liver diseases',
    mode = 'markers',
    macter =dict(
        size = 12,
        color = 'rgb(21L,I88,53)',
        syEiol = 'hash-open',
        line = dici(wzdli = 2)))

trace3 = go. Scatter(
    x = chronic_Deaths_OroupingByYear 'Year'],
    y = chronic_Deaths_OroupingByYear 'Lower Respiratory Infections'],
    name = 'Lower Respiratory Infections',
    mode = 'markers',
    earter = d1ct(
        six = 12,
        c0Uv - 'rgb(28B,I58,2B)',
        sydol = 'hazy-seen',
        line =difi(wiât =2)))

data =[tracee,trace,t tracez,trace3l
layout = go.Layovt(
    title = '1m0 to 2e*s Deaths - Chronical Dzases',
    xaxls = d1ct(t1t1e = 'Year' ),
    yxG = d1ct(t1Ne = ' Death)
    NwermWe ='closerC,


fig = go.Figure(data=data, 1ayout=1aycxJt)
fig -slow( )
```



WE CAN SEE THAT THE MAXIMUM DEATH IS CAUSED BY CHRONIC RESPIRATORY DISEASES AND LEAST DEATH IS CAUSED BY CHRONIC KIDNEY DISEASES IN ALL THE GIVEN YEARS WHICH IS 1990 TO 2019.

CONCLUSION

Total rows 6120 and 34 columns in the dataset.

I found out that there are many diseases which continuously increasing such as Neoplasm's, HIV/AIDS, Diabetes, Cardiovascular Diseases, Digestive disorder and Alzheimer.

I found out that there are many diseases which are continuously decreasing too such as Acute Hepatitis, Diarrheal Diseases, Nutritional Diseases and Meningitis

Parkinson Diseases seems to be constants till 1990 to 1993 after that no data is present for the same.

We can see that in all the given years i.e. 1990 to 2019, Road accident have taken Maximum life's and the least can death can be seen in Exposure to force of Nature

In case of Death by crime, self-harm and Accident -> Maximum deaths have been taken place by Conflict and Terrorism and the second highest death have been recorded by - Interpersonal Violence.

Rest all other factors of death are under 200k which can be even further minimized

ALL THE GOVERNMENT AND CONCERNED BODIES SHOULD TAKE RESONABLE STEP TO ENSURE THAT ALL THE AREAS WITH MAXIMUM DEATHS CAN BE MINIMIZED AND PROPER ACTION SHOULD BE TAKEN

IN CASE OF CONFLICT &TERRIOSM AND INTERPERSONAL VIOLENCE SO THAT IT SHOULD BE REDUCED TO MINIMAL.