DMDW ASSIGNMENT 2

BACHATE SHAILESH NAVNATH
BT18CSE103

PART 1 EXTERNAL DATASET

DATASET

In this project we will examine causes of death dataset, published by "Our World in Data" (https://ourworldindata.org/causes-of-death) The dataset consists of 34 different cause of death counts by country per year. It covers the years from 1990 to 2017.
I am using python packages like **NumPy, Pandas, Matplotlib, Seaborn, PyCountry, pygal_maps_world** to work on this dataset

STRUCTURE OF DATA

'Causes of death' dataset gives annual number of deaths by cause of death. The dataset contains 6686 entries. It consists of 37 columns, 34 of them give the dead counts (estimates?) for each death cause. Also, year and country information is provided. Each row represents combinations country/year pair

DATA CLEANING

There are some missing values in some of the columns. We can use the **df.info()** method to get the number of entries with non null values for each column.
A part of the data (218 rows) is filled with NaN except the **Terrorism** column. One can choose to fill those NaN cells with 0, or drop those rows which leads to loss of information for terrorism related death counts for that particular year/country pair. We will choose the first option.
The columns with null/NaN death counts will be filled with zero.

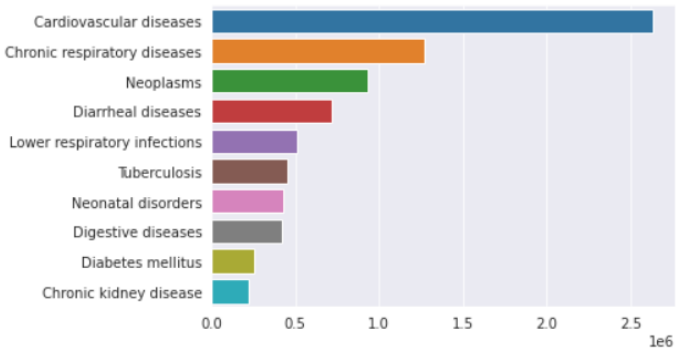
Values in the death count cells are floating numbers, We have to convert them to integers as these values represent counts of death (**using df.iloc[:,4:] = df.iloc[:,4:].astype(int)**)

EXPLORATORY DATA ANALYSIS

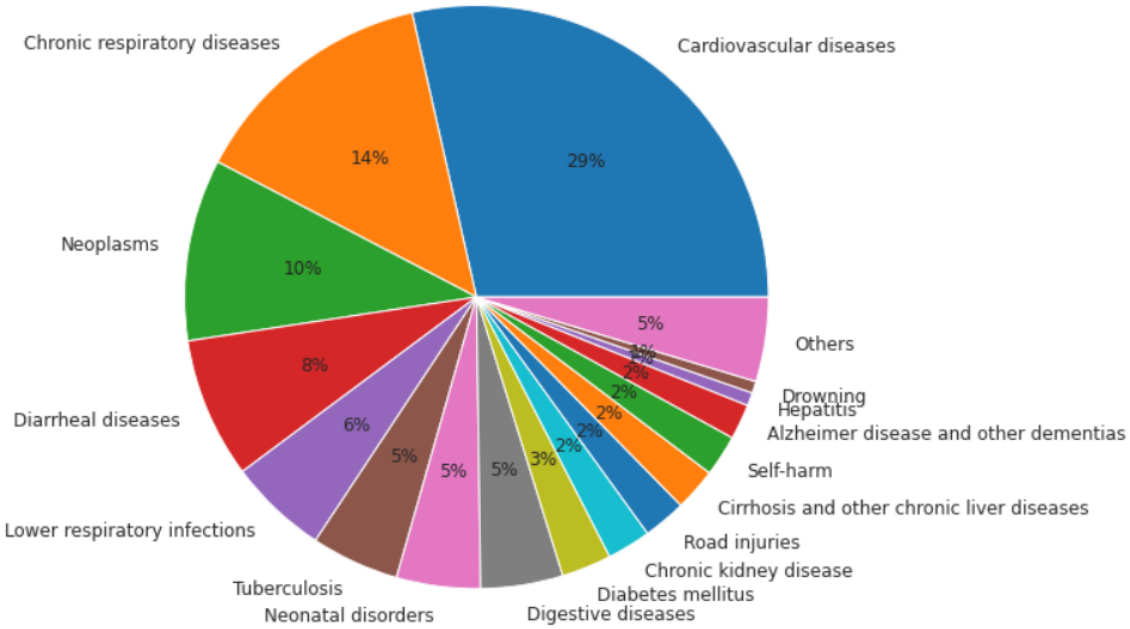
Top 10 causes of death for India in 2017 (latest available year)

```
# Top 10 causes of death for India in 2017 (latest available year)

india_2017 = df[df.Entity == "India"].groupby("Year").sum().loc[2017].sort_values(ascending=False)
sns.barplot(x=india_2017.values[:10],y=india_2017.index[:10],orient="h")
plt.show()
```



When we look at the pie chart for this data we can see, almost 3 out 10 deaths in India are caused by Cardiovascular diseases in 2017.



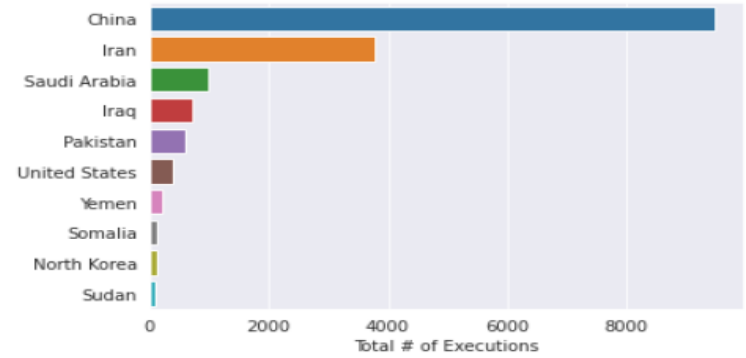
`df.groupby('Year').sum()` will give the total number of deaths per year by a particular cause of death.

The "Execution" column is actually not numeric as it has categories like ">1000", let's fix it and continue descriptive stats.

Top 10 countries with the highest execution numbers

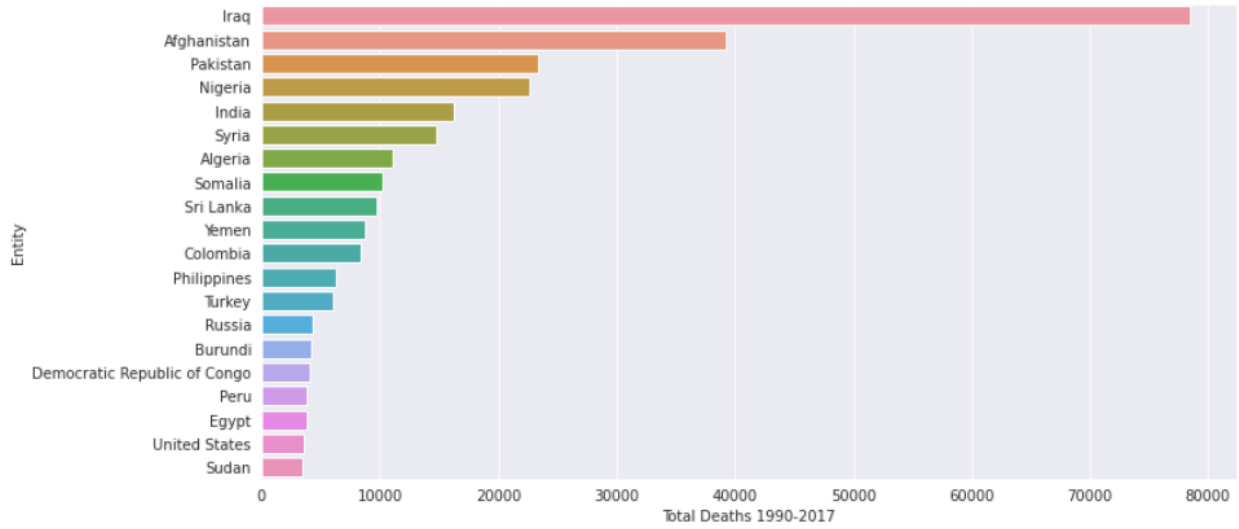
```
country_execution = df.groupby("Entity").sum()["Execution"].sort_values(ascending=False)
sns.barplot(x=country_execution.values[1:11],y=country_execution.index[1:11],orient="h")
plt.ylabel("")
plt.xlabel("Total # of Executions")
```

```
Text(0.5, 0, 'Total # of Executions')
```



Top 20 countries which suffered from terrorism the most.

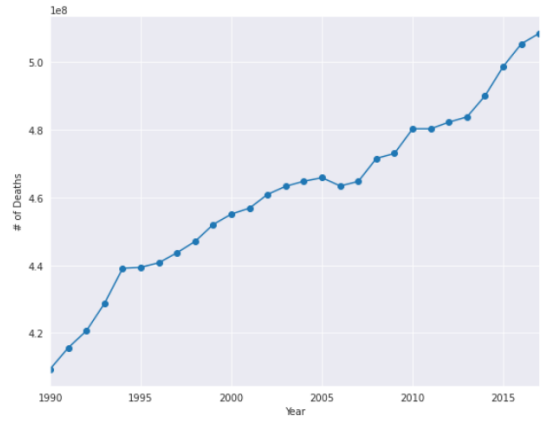
Our dataset Entity column contains a mix of country, continent, region, territory information too such as Sub-Saharan Africa, South America etc. For this graph we are only interested in the countries



```
Country_terrorism=df[df.Code.notnull()].groupby("Entity").sum()[
"Terrorism"].sort_values(ascending=False)
```

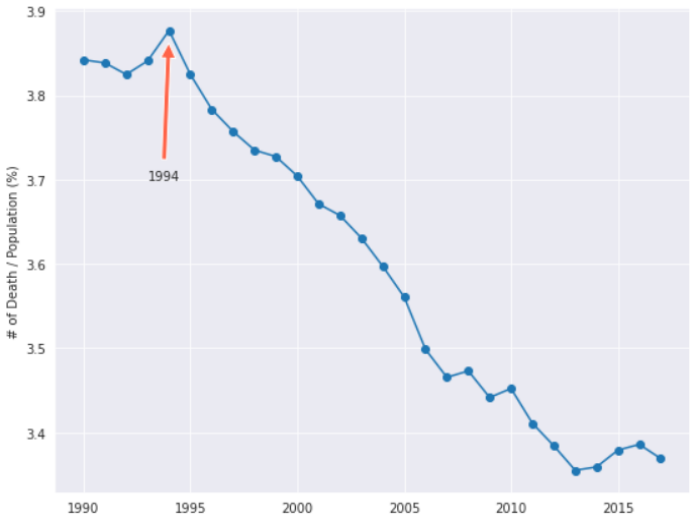
DEADLIEST YEAR

Deadliest year appears to be 2017 in terms of death counts, however this was due to the increase of the world population each year. We need to check the death rate per year.

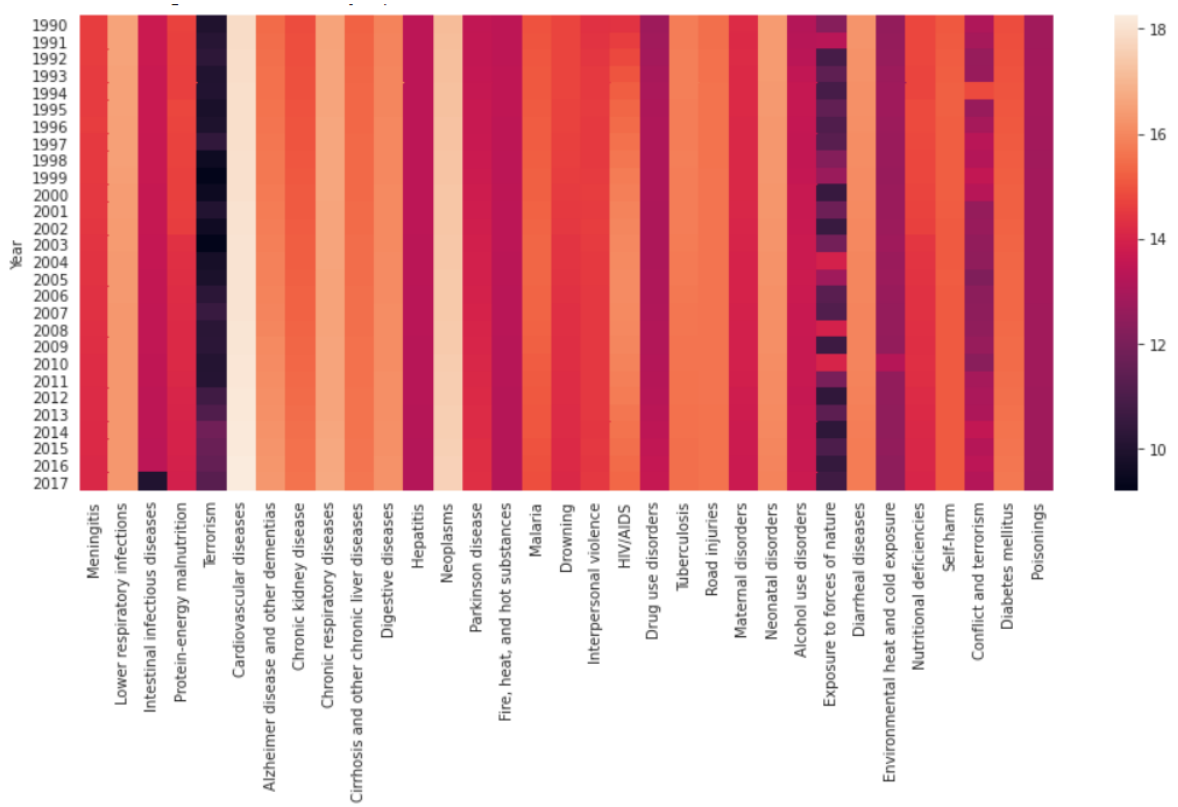


Let's use the world population per year which we got from the internet and plot year vs death rate (%):

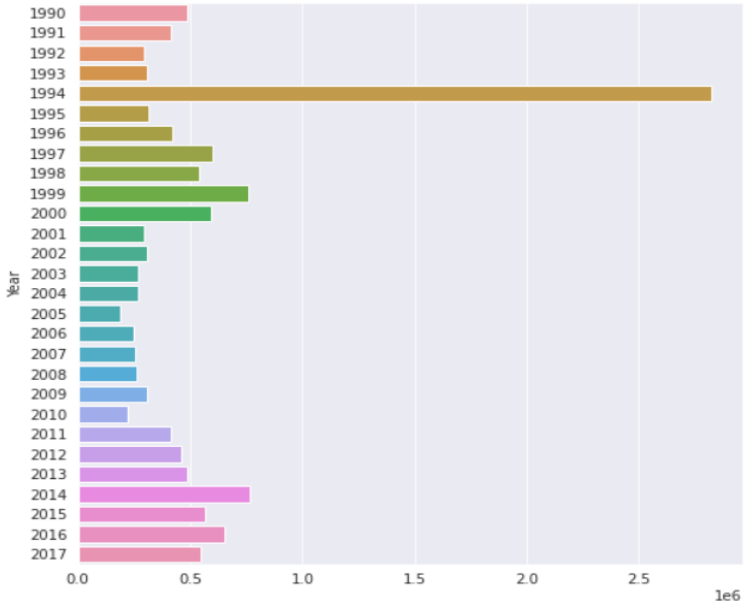
- Death rate has been going down every year and showing signs that 3.35% levels could be the plateau for this metric
- **1994** was the deadliest year with almost 4% of the population was died



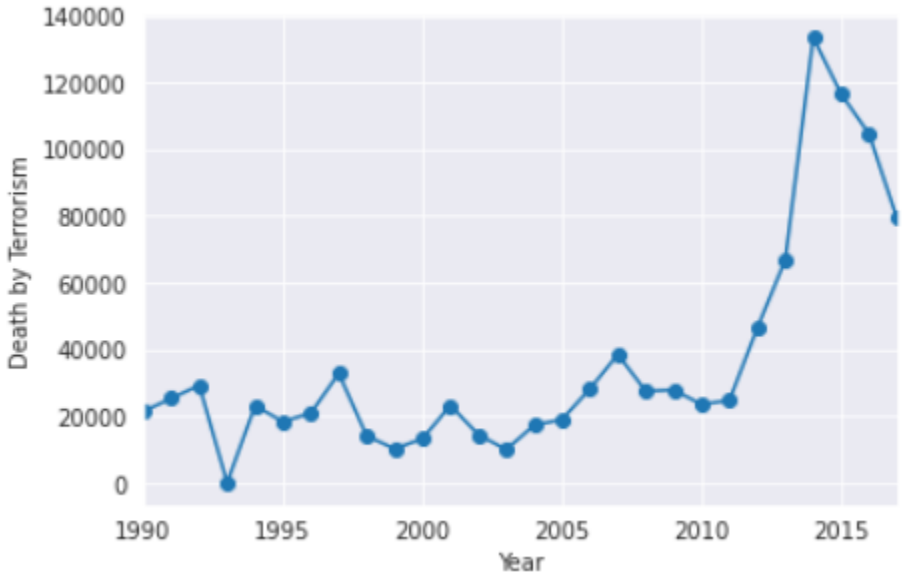
Let's examine what was the cause for this high death rate in 1994. Conflict column of the heatmap shows an interesting light colour on year 1994 (the lighter the colour the higher the number of deaths)



Conflict graph shows that **1994** was the year where the deadliest conflicts took place in the world.



Death by terrorism peaked in 2014, and has been declining since then, however no effect was observed on 1994's death toll



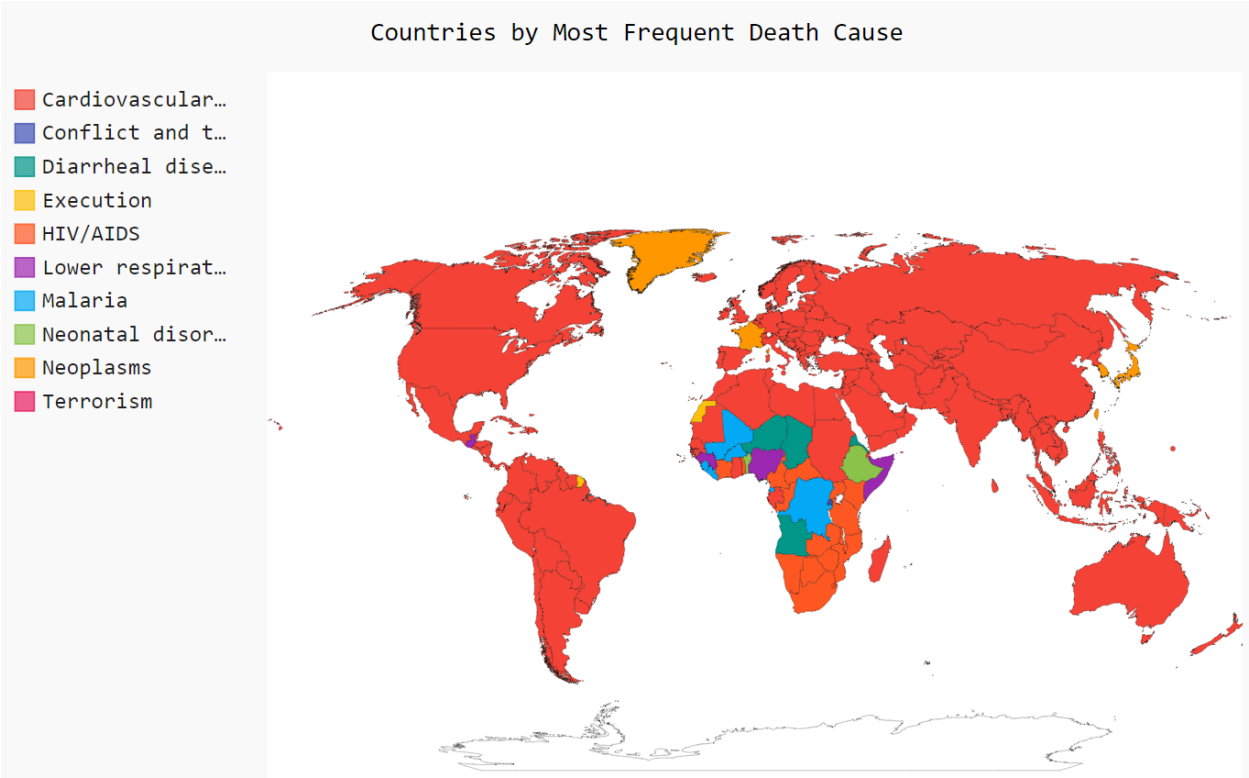
Which country suffers from what disease the most?

```
df.Code = df.Code.dropna().apply(str.lower)
cause_by_country =
df.groupby("Code").sum().drop(["Year","total_death"],axis=1).idxmax(axis=1)
cause_by_country.value_counts()
```

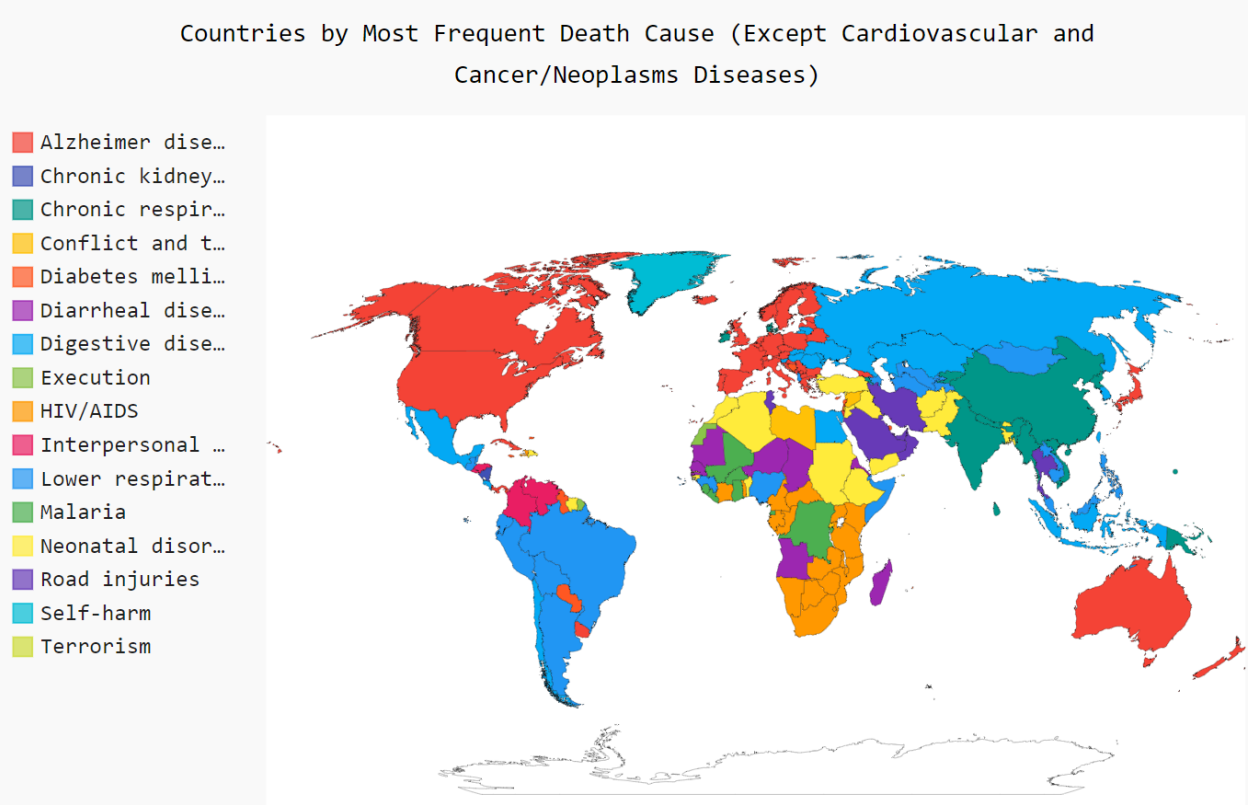
Cardiovascular diseases	155
HIV/AIDS	17
Execution	7
Malaria	7
Terrorism	6
Neoplasms	5
Lower respiratory infections	5
Diarrheal diseases	4
Neonatal disorders	2
Conflict and terrorism	1

Group countries by the most frequent disease types

In order to colour countries by disease, let's use the pygal package. We will need the pycountry package to convert 3 digit country codes to 2 digit country codes which is needed in pygal world mapping. We will use pycountry to convert country names into country codes which will be used in the map.



It is obvious that Cardiovascular Disease dominates the death toll all over the world, how would the World map look if we took them out?



- India, China and surrounding countries suffer from Respiratory diseases like **Asthma** and **Lung Cancer** etc.
- It appears that the second most leading cause of death in rich countries like the USA, Canada, Europe, Japan and Australia is **Dementia** (mainly caused by Alzheimer's disease). This could be due to the high frequency of elder people in their population.
- **Conflict** is leading cause for war territories like Syria and Palestine
- For Russia and their neighbours like old USSR countries, Eastern European countries' leading cause is **Digestive** diseases like Ulcer, Cirrhosis, Hepatitis. It could be related to excessive consumption of alcohol.
- **Diarrheal** diseases causes deaths mostly in the mid African region
- **HIV/AIDS** deaths are the most frequent in the South African region

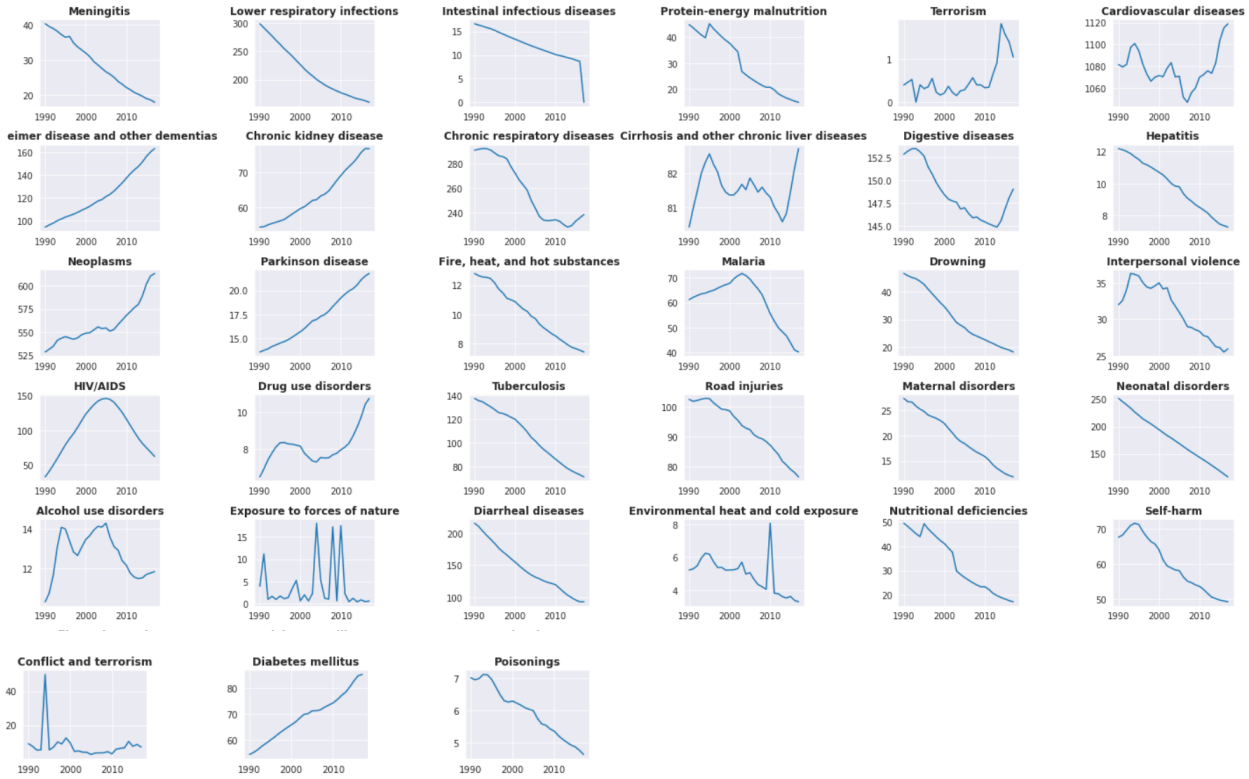
- Some of the South American countries have **Homicide** as the leading cause for death.
- Lower respiratory infections like **Tuberculosis**, **Pneumonia** is the leading cause for mainly South American countries like Brazil and Argentina.
- The countries where **road accidents** are one of the leading causes of death are in the Gulf region like Iran, Saudi Arabia, UAE. It could be related to their habit of car stunt driving.
- Greenland is the only country where **suicide** is leading cause of death.

Historical Trends of Diseases

Let's now group the diseases by their historical increasing or decreasing trends. Calculations will be based on deaths per 100000 for a particular disease.

- Even though Digestive and Respiratory diseases were dropping constantly for a period of time, they started to enter an increasing trend since 2013. Still way better as compared to 90s
- Suicide rates are dropping since mid-90s
- Peak in the Heat graph could be a mistake in the dataset or it is a very distinguishing event happened in 2010 which caused this sudden jump
- Alzheimer disease and other dementias, Chronic kidney diseases, Neoplasms, Parkinson, and Diabetes diseases have been increasing since the beginning of our data span (1990)
- While deaths related to Meningitis, Lower respiratory infections, Intestinal infectious diseases, Hepatitis, Drowning, Fire, heat, and hot substances, Tuberculosis, Road injuries, Maternal disorders, Neonatal disorders, Diarrheal diseases, Poisonings have been decreasing since the beginning of our data span(1990)
- Drug and alcohol use disorders are correlated with Liver disease trend
- HIV/AIDS related deaths rate started to decline since 2005
- Malaria related deaths rate started to decline since 2004
- While drug related deaths are consistently increasing since 2005

- Natural disasters show no trend as expected.



PART 2 : INBUILT DATASET

DATASET

I am using the IRIS dataset for this project.

The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other. I am using python libraries like **NumPy, Pandas, Matplotlib, Seaborn** to work on this dataset

Attribute Information:

1. sepal length
2. sepal width

3. petal length
4. petal width
5. species: -- Setosa -- Versicolor -- Virginica

Here, species attribute is categorical and other four are numerical attributes

STATISTICAL INSIGHTS:

1. CENTRAL TENDENCIES LIKE MEAN MEDIAN MODE

Mean, median, mode of each attribute is calculated, we can also apply these methods on data groups. For example we can divide the dataset into groups according to their species and then calculate central tendency values for each group.

2. STANDARD DEVIATION and VARIANCE

The sample variance is a measure of dispersion, roughly the “average” squared distance of a data point from the mean.

The standard deviation is the square root of the variance and interpreted as the “average” distance a data point is from the mean.

3. QUANTILE

The pth percentile is the number in the dataset such that roughly p% of the data is less than this number. This number is also referred to as a quantile.

4. NUMBER OF UNIQUE ENTRIES IN EACH COLUMN

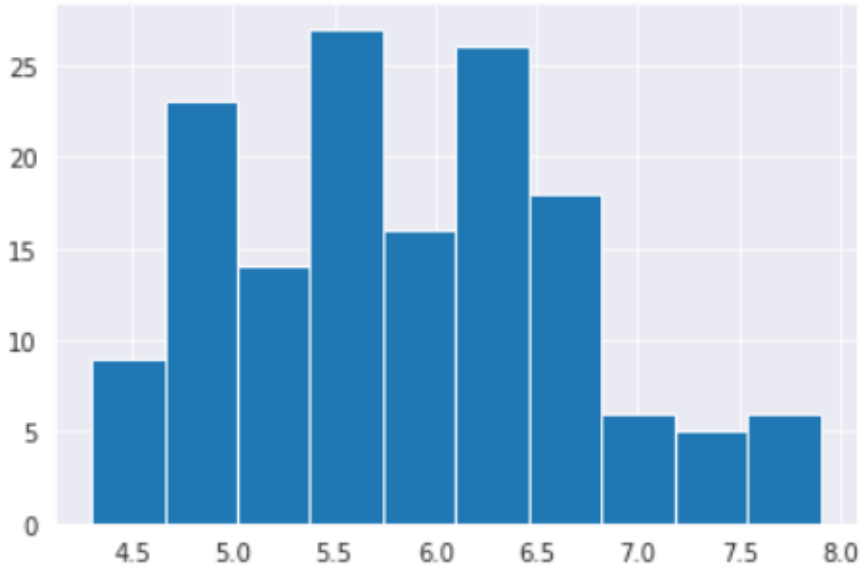
There’s only one duplicate entry in the dataset.

5. NUMBER OF NULL ROWS

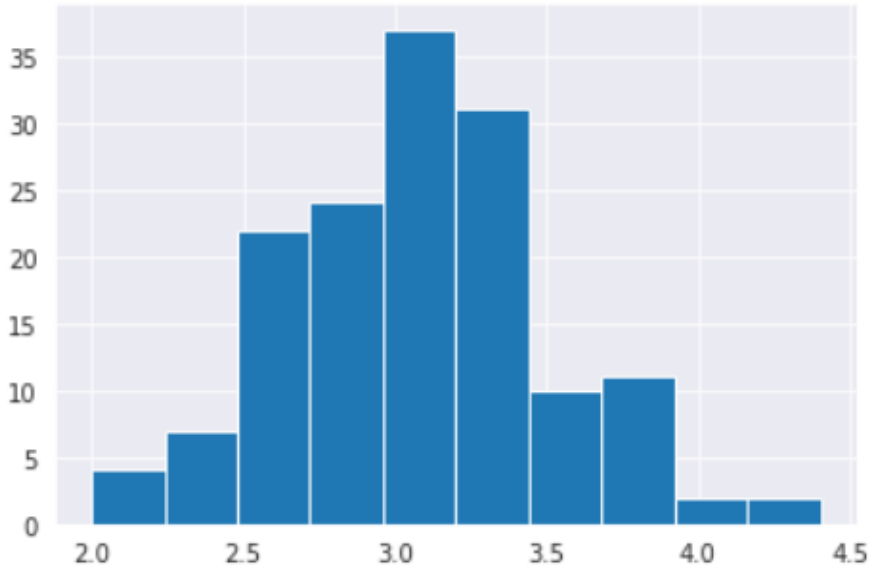
There are no null rows in this dataset.

DATA VISUALIZATION: HISTOGRAMS

```
1 # histograms
2 df['sepal_length'].hist()
3 plt.show()
```



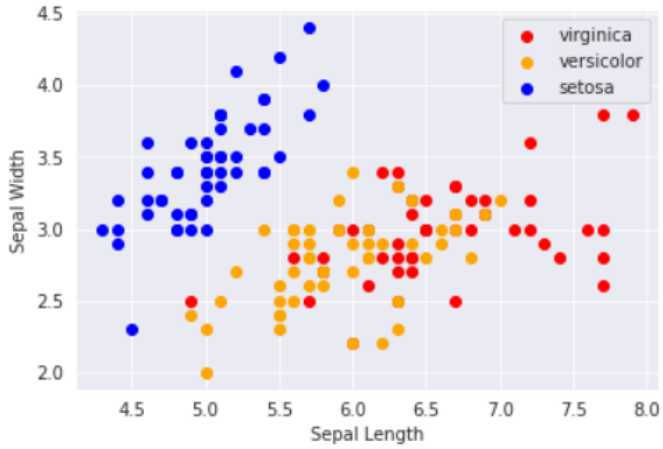

```
1 df['sepal_width'].hist()
2 plt.show()
```

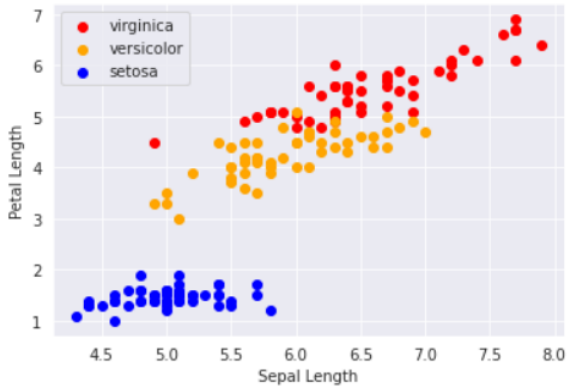


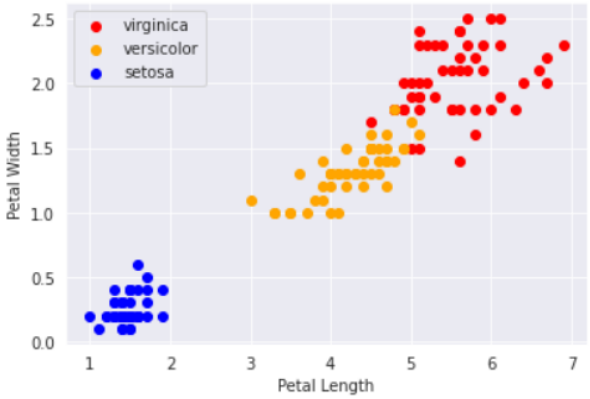
DATA VISUALIZATION: SCATTER PLOTS

```
# scatterplot
colors = ['red', 'orange', 'blue']
species = ['virginica','versicolor','setosa']
```

```
for i in range(3):
    x = df[df['species'] == species[i]]
    plt.scatter(x['sepal_length'], x['sepal_width'], c = colors[i], label=species[i])
plt.xlabel("Sepal Length")
plt.ylabel("Sepal Width")
plt.legend()
plt.show()
```







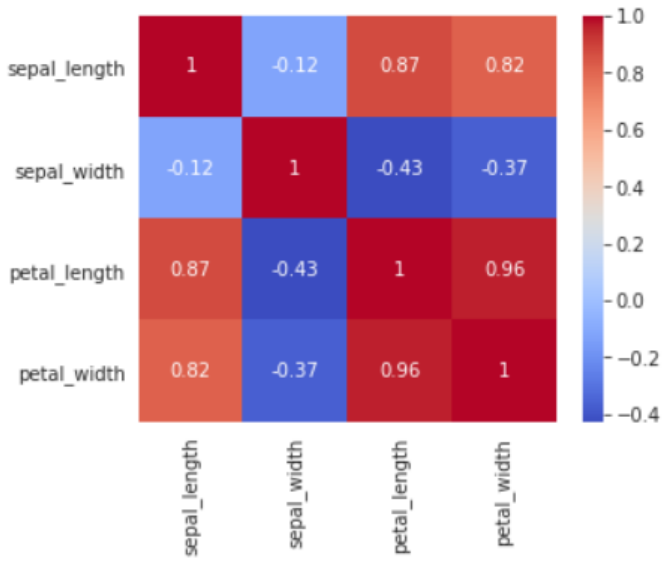
CORRELATION MATRIX

A correlation matrix is a table showing correlation coefficients between variables. Each cell in the table shows the correlation between two variables. The value is in the range of -1 to 1. If two variables have high correlation, we can neglect one variable from those two.

```
1 df.corr()
```

	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.117570	0.871754	0.817941
sepal_width	-0.117570	1.000000	-0.428440	-0.366126
petal_length	0.871754	-0.428440	1.000000	0.962865
petal_width	0.817941	-0.366126	0.962865	1.000000

```
1 corr = df.corr()
2 fig, ax = plt.subplots(figsize=(5,4))
3 sns.heatmap(corr, annot=True, ax=ax, cmap = 'coolwarm')
4 plt.show()
```



DATA INSIGHTS

Setosa has smaller sepal length when compared to virginica and versicolor, but sepal width values are pretty high compared to the other two.

Setosa has smaller petal length and width values compared to other two and is much more concentrated.

Petal length and width values for versicolor lie in between Setosa and Virginica.

But sepal length and width values are spread all over the graph, i.e. values are distributed all over the graph, but we can conclude that generally Versicolor lies in between the other two in sepal length, but sepal width of versicolor is shorter than setosa and has almost similar values as that of virginica.

Virginica has highest petal length and width values and these are concentrated in a high value range.

The Sepal length value of virginica is also highest and the values are spread over a long range. But Setosa has larger sepal width on average.

High correlation between petal length and width columns can be found.

Generally the iris plant having larger petal value also has larger petal values. Same can not be said for the sepal length and sepal width values though.