PROJECT -1
EXPLORE WEATHER TRENDS

Submitted by:

ABHISHEK KATHURIA

DATA EXTRECTION

- This step was performed using the SQL commands to extract data from the temperatures database.

The Database Schema

There are three tables in the database:

- city_list - This contains a list of cities and countries in the database. Look through them in order to find the city nearest to you.
- city_data - This contains the average temperatures for each city by year (ºC).
- global_data - This contains the average global temperatures by year (ºC).

Steps To Extract Data

- To extract all the data from the table '**city_list**', I used the following SQL command:
  - Select * from city_list
- To extract all the data from the table '**city_data**', I used the following SQL command:
  - Select * from city_data where city = 'Delhi'

  This enabled me to select only those rows where the name of the city was 'Delhi'. Hence, I was able to extract the data specific to my city out of the whole table.

- To extract all the data from the table '**global_data**', I used the following SQL command:
  - Select * from global_data
- The final step was to download all the **'.csv'** file onto my local computer for data analysis.

Data Analysis using python (Jupyter notebook)

- The first step was to import all the required libraries:

Weather Trends

Import libraries

```
In [89]: import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
```

The libraries included **pandas** which is the main library used for data manipulation and analysis. The **numpy** library is used for manipulating multi dimensional arrays and other mathematical calculations. The **matplotlib** library is used for plotting charts and other visualizations.

- The next step is to import the dataset (which is the downloaded csv files)

Import dataset

```
In [34]: df_city=pd.read_csv('city_data_delhi.csv')
         df_global=pd.read_csv('global_data.csv')
```

```
In [90]: df_global.head(10)
```

Out[90]:

	year	avg_temp
0	1750	8.72
1	1751	7.98
2	1752	5.78
3	1753	8.39
4	1754	8.47
5	1755	8.36
6	1756	8.85
7	1757	9.02
8	1758	6.74
9	1759	7.99

```
In [91]: df_city.head()
```

Out[91]:

	year	city	country	avg_temp
0	1796	Delhi	India	25.03
1	1797	Delhi	India	26.71
2	1798	Delhi	India	24.29
3	1799	Delhi	India	25.28
4	1800	Delhi	India	25.21

- In the next step, I merged the two dataframes using inner join based on the column name 'year'. I also renamed the columns to indicate the average city temperature and average global temperature.

```
In [92]: # Merge the datasets using inner join based on the column name 'year'.
         df=df_city.merge(df_global, how='inner', on='year')

In [93]: # Rename the columns to indicate the average city temperature and average global temperature.
         df=df.rename(columns={'avg_temp_x':'avg_city_temp', 'avg_temp_y':'avg_global_temp'})

In [94]: df.head()
```

Out[94]:

	year	city	country	avg_city_temp	avg_global_temp
0	1796	Delhi	India	25.03	8.27
1	1797	Delhi	India	26.71	8.51
2	1798	Delhi	India	24.29	8.67
3	1799	Delhi	India	25.28	8.51
4	1800	Delhi	India	25.21	8.48

```
In [95]: df.shape
```

Out[95]: (218, 5)

- Next was to check whether any null values are present in the dataset.

```
In [96]: # Check for null values
         df.isnull().values.any()

Out[96]: True

In [98]: # Count the number of total null values present in the dataset
         df.isnull().sum()

Out[98]: year              0
         city              0
         country           0
         avg_city_temp    17
         avg_global_temp   0
         dtype: int64

In [99]: # Drop all columns with null values
         df=df.dropna()

In [101]: df.shape
          # This means that 17 rows have been omitted

Out[101]: (201, 5)
```

It can be clearly seen that there were **17 records** present which contained the null values, hence I eliminated all those records.

- This step involves calculating the **moving averages**. Here, I took the window size as **8** and used the predefined python function for calculating the moving average which is known as the **rolling function**.

```
In [103]: # Here, the window size is chosen as 8
          df['city_moving_avg'] = df.iloc[:,3].rolling(window=8).mean().dropna()
          df['global_moving_avg']= df.iloc[:,4].rolling(window=8).mean().dropna()

In [105]: df.head(10)

Out[105]:
```

	year	city	country	avg_city_temp	avg_global_temp	city_moving_avg	global_moving_avg
0	1796	Delhi	India	25.03	8.27	NaN	NaN
1	1797	Delhi	India	26.71	8.51	NaN	NaN
2	1798	Delhi	India	24.29	8.67	NaN	NaN
3	1799	Delhi	India	25.28	8.51	NaN	NaN
4	1800	Delhi	India	25.21	8.48	NaN	NaN
5	1801	Delhi	India	24.22	8.59	NaN	NaN
6	1802	Delhi	India	25.63	8.58	NaN	NaN
7	1803	Delhi	India	25.38	8.50	25.21875	8.51375
8	1804	Delhi	India	25.68	8.84	25.30000	8.58500
9	1805	Delhi	India	25.30	8.56	25.12375	8.59125

- Now, the line chart was plotted using the **matplotlib** library. Here, **blue color** indicates the moving average of the temparature of the particular

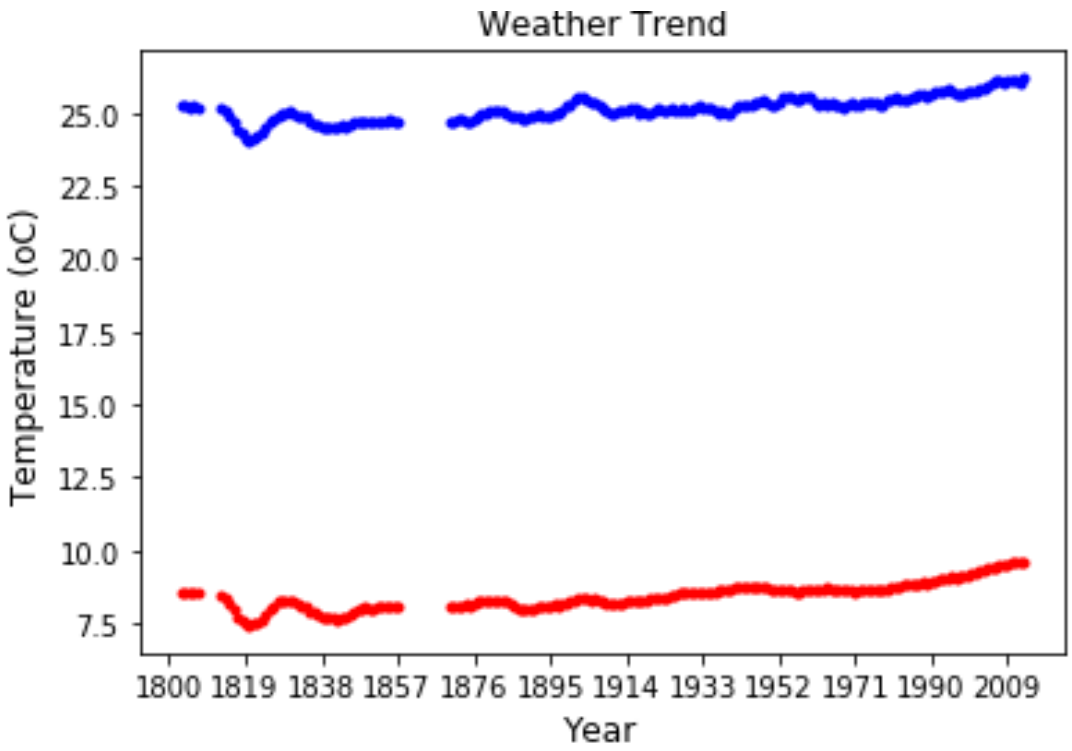
city which is chosen as '**Delhi**'. The **red color** indicates the moving average of the **global temperature**.

- **Line Chart**

```
In [108]: # The blue color indicated the moving average of the temparature of the particualr city which is chosen as 'Delhi'
          # The red color indicates the moving average of the global temperature
          plt.plot(df['year'],df['city_moving_avg'],'bo', df['year'],df['global_moving_avg'],'ro', linewidth=1, markersize=3)
          plt.xlabel('Year', fontsize=12)
          # xticks is used to increase the number of divisions for the x-axis
          plt.xticks(np.arange(1800,2015,18))
          plt.ylabel('Temperature (oC)', fontsize=12)
          plt.title('Weather Trend')

Out[108]: Text(0.5, 1.0, 'Weather Trend')
```

Line chart



Observations

- It can be clearly noted that the temperature of Delhi over the years was in the range **23.75** Degree Celsius **to 27.01** Degree Celsius.
- We can also see that the Global temperature over the years was in the range **8.2** Degree Celsius to **9.86** Degree Celsius.
- We can also see that there was a steep dip in the year **1819** for both the lines which clearly indicates that in the year 1819, the average temperature was **minimum** out of all the years. Hence, we can say that the year 1819 was the **coldest**.
- We can also see that from the year **1952** to **2013**, there is a steep increase in the curve of average temperature of Delhi as well as the global temperature, which indicates that the average temperature rose drastically over this time span. This means that **global warming** was an eminent issue over these years.
- One thing we can also notice is that the rise in the average temperature for Delhi is more than the global average temperature from the year **1971 to 2013** which means that Delhi was more affected by the temperature rise due to **pollution, burning of fossil fuels**, **rapid urbanization** and other factors which let to **global warming**