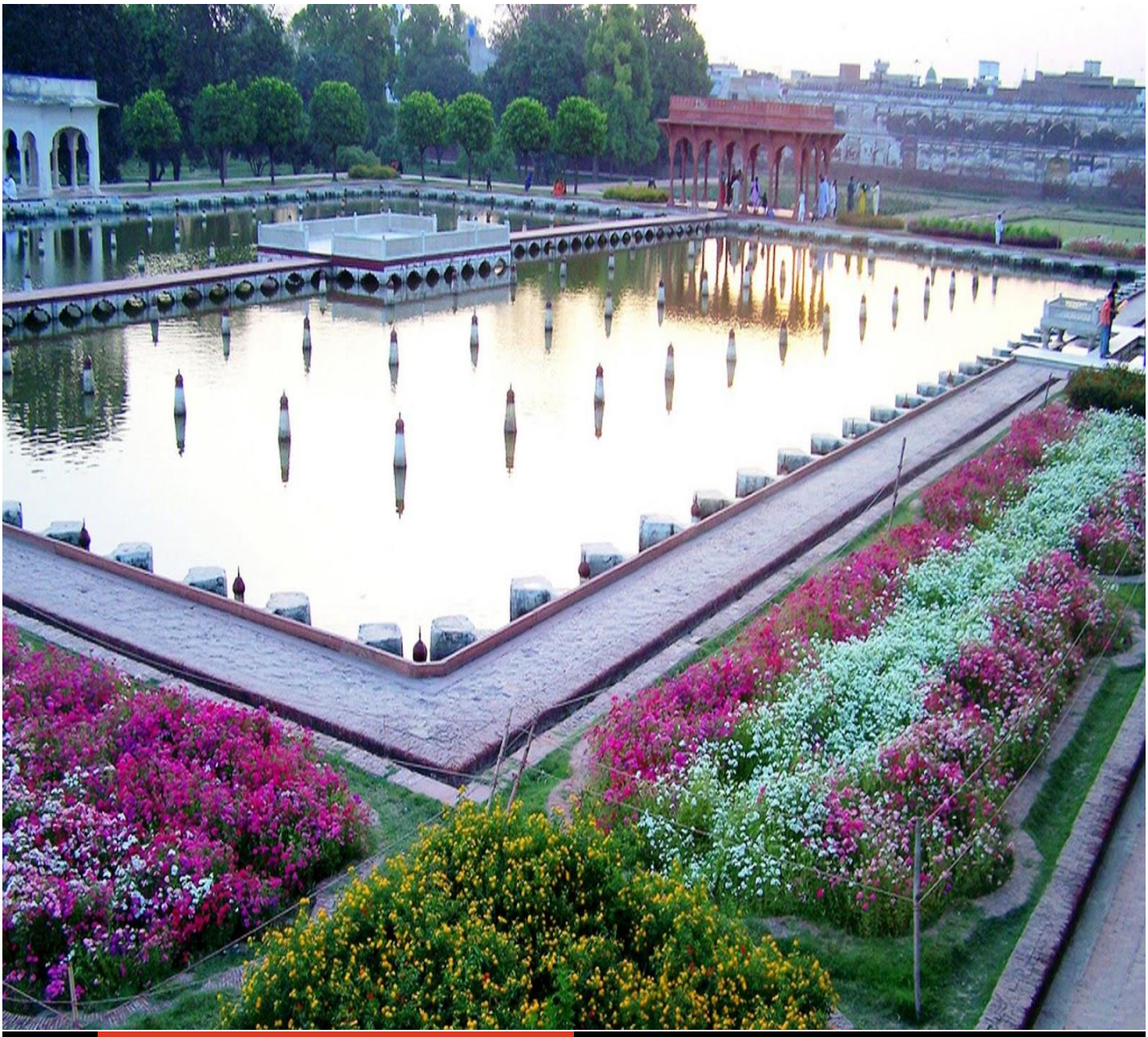


EXPLORE WEATHER TRENDS

JANUARY 17, 2021

USAMA TARIQ

Udacity – Data Analyst
Nanodegree – Project – 1

OVERVIEW

Local temperature (°C) data of Lahore (LHR), the city of gardens -located in the north-eastern portion of Pakistan (PK), is analyzed, and compared with the global temperature (°C) data. I accessed the Client's, Udacity-Advance Your Career, database to extract the required data, manipulate, visualize, and interpret valuable insights from data for better and concrete decision making.

Goals

- Extract the data from the database, transform it into comma-separated values (CSV) file.
- Load the CSV file for analysis.
- Visualize and compare the local and global temperature data.
- Identify the trends of the local and global temperature data.

Tools

- SQL: for extraction of the data from the database
- Jupyter Notebook: as IDE for Python & Pandas
- Pandas: for loading the CSV file for analysis.
- Python: to visualize and compare the local and global temperature data.
- Microsoft Office 365: interpreting the insights of data and share valuable results in the form of a report.

EXTRACT THE DATA

Finding the Closest Big City

One the basis of my SQL expertise, I wrote the following query to find the closest city listed in the cities of the Pakistan. And I found my own city in the database.

```
1. select *
2. from city_list
3. where country = 'Pakistan';
```

Above query's results were

city	country
Faisalabad	Pakistan
Gujranwala	Pakistan
Hyderabad	Pakistan
Islamabad	Pakistan
Karachi	Pakistan
Lahore	Pakistan
Multan	Pakistan
Peshawar	Pakistan
Rawalpindi	Pakistan

Extracting City Level data

For city-level data, the city I select is Lahore, Pakistan. The query used to extract the whole data for 'Lahore' is listed below.

```
1. select *
2. from city_data
3. where city='Lahore';
```

And the results for this query were like,

Output 198 results Download CSV

year	city	country	avg_temp
1816	Lahore	Pakistan	24.69
1817	Lahore	Pakistan	22.74
1818	Lahore	Pakistan	23.43
1819	Lahore	Pakistan	22.83

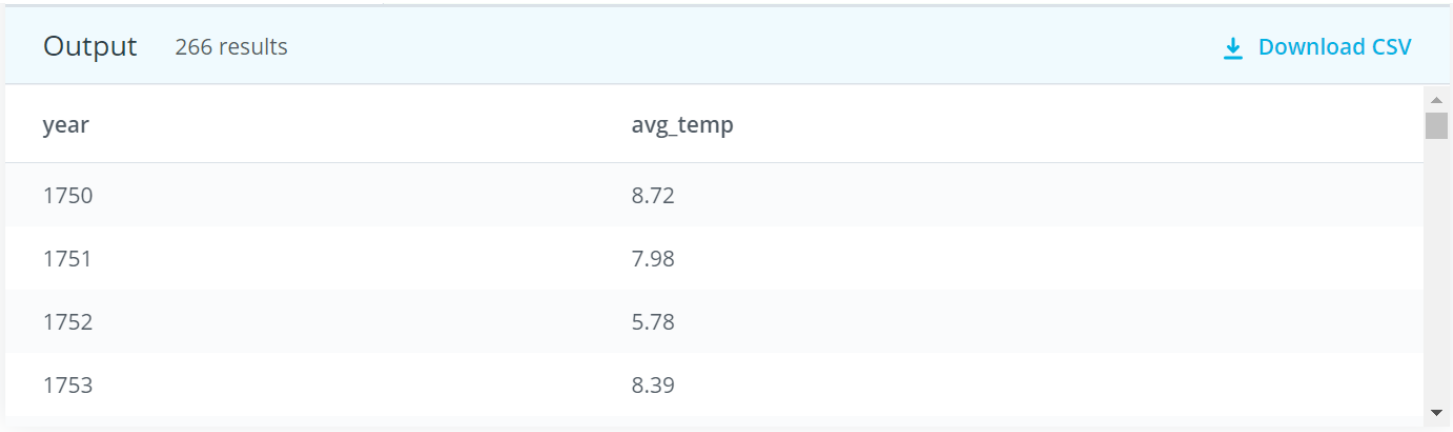
Although the number of records is limited to 198, but its burdensome to show the full data.

Extracting Global Level data

Query for retrieving the global level data,

```
1. select *
2. from global_data;
```

This query yielded 266 number of records as a global-level data.



Finally, the required data for the analysis has been extracted and transformed successfully. In the upcoming step, the data will be load for analysis.

LOADING DATA (CSV)

For the analytical purposes of the data, csv is supported by many different tools. But I prefer the Jupyter notebooks over Spreadsheets.

In Jupyter notebooks, you can analyze data critically with the help Python and several supported libraries like Pandas, NumPy etc.

So, the loading of data from csv file to a Data Frame in Python is done by,

```
1. df_CityData = pd.read_csv('City-Level-Data (CSV).csv')
2. df_GlobalData = pd.read_csv('Global-Data (CSV).csv')
```

The first line of code reads data from "City-Level-Data (CSV).csv" file and writes it to a "df_CityData" data frame. And the second line of code reads data from "Global-Data (CSV).csv" file and writes it to a "df_GlobalData" data frame.

To ensure whether the data has been successfully read and stored in the desired data frames, the lines of code used to examine, and the results yielded are shown side by side.

```
1. df_CityData.head()
```

	year	city	country	avg_temp
0	1816	Lahore	Pakistan	24.69
1	1817	Lahore	Pakistan	22.74
2	1818	Lahore	Pakistan	23.43
3	1819	Lahore	Pakistan	22.83
4	1820	Lahore	Pakistan	24.22

```
1. df_GlobalData.head()
```

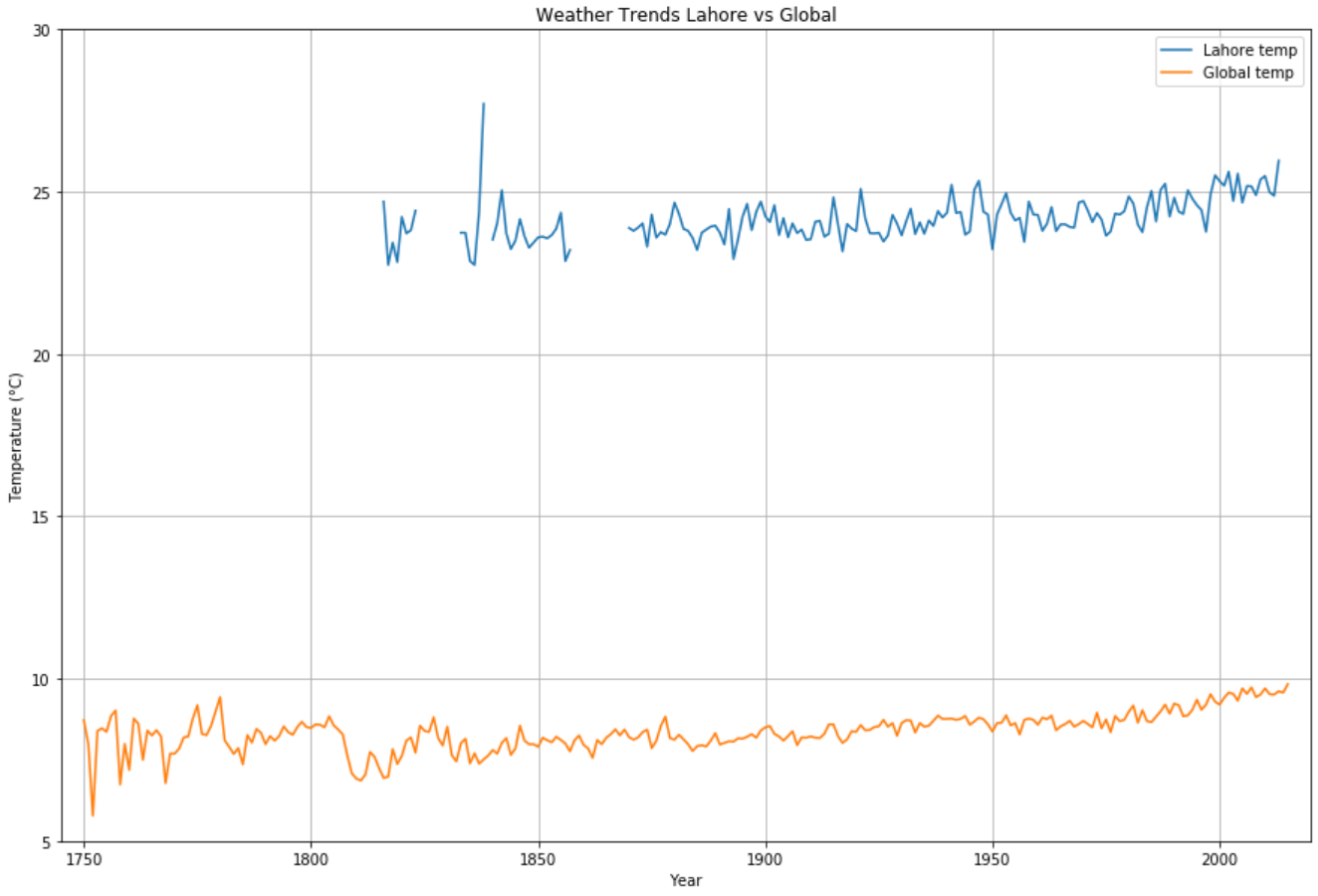
	year	avg_temp
0	1750	8.72
1	1751	7.98
2	1752	5.78
3	1753	8.39
4	1754	8.47

VISUALIZING & CALCULATING MOVING AVERAGES

Visualizing raw data

Firstly, to understand the data and its trends. Drawing a line chart to explore the spikes and fluctuations in the local temperatures (ºC) and global temperatures (ºC) for better comparisons.

```
plt.figure(figsize=[15,10])
plt.plot(df_CityData['year'],
         df_CityData['avg_temp'],
         label='Lahore temp')
plt.plot(df_GlobalData['year'],
         df_GlobalData['avg_temp'],
         label='Global temp')
plt.legend()
plt.grid(True)
plt.axis([1745, 2020, 5, 30])
plt.title('Weather Trends Lahore vs Global')
plt.xlabel('Year')
plt.ylabel('Temperature (°C)')
plt.show()
```



Lahore temperature (ºC) are the blue ones, the average maximum temperature (ºC) ever reached in Lahore is 27.70ºC in the year 1838 and the average minimum temperature (ºC) ever recorded is 22.74ºC in year 1817 & 1836 as per the data. The span of data is from year 1816 to year 2013.

Global temperature (ºC) are the orange ones, the average maximum temperature (ºC) ever reached globally is 9.83ºC in the year 2015 and the average minimum temperature (ºC) ever recorded is 5.78ºC in year 1752 as per the data. The span of data is from year 1750 to year 2015.

But the temperatures fluctuations are so rapid around the years, so we must calculate the moving average over the years temperature (ºC). To predict better results.

Calculating Moving Averages on Raw Data & Visualizing

Calculating moving averages for both local & global data temperatures (ºC) for the span of a decade.

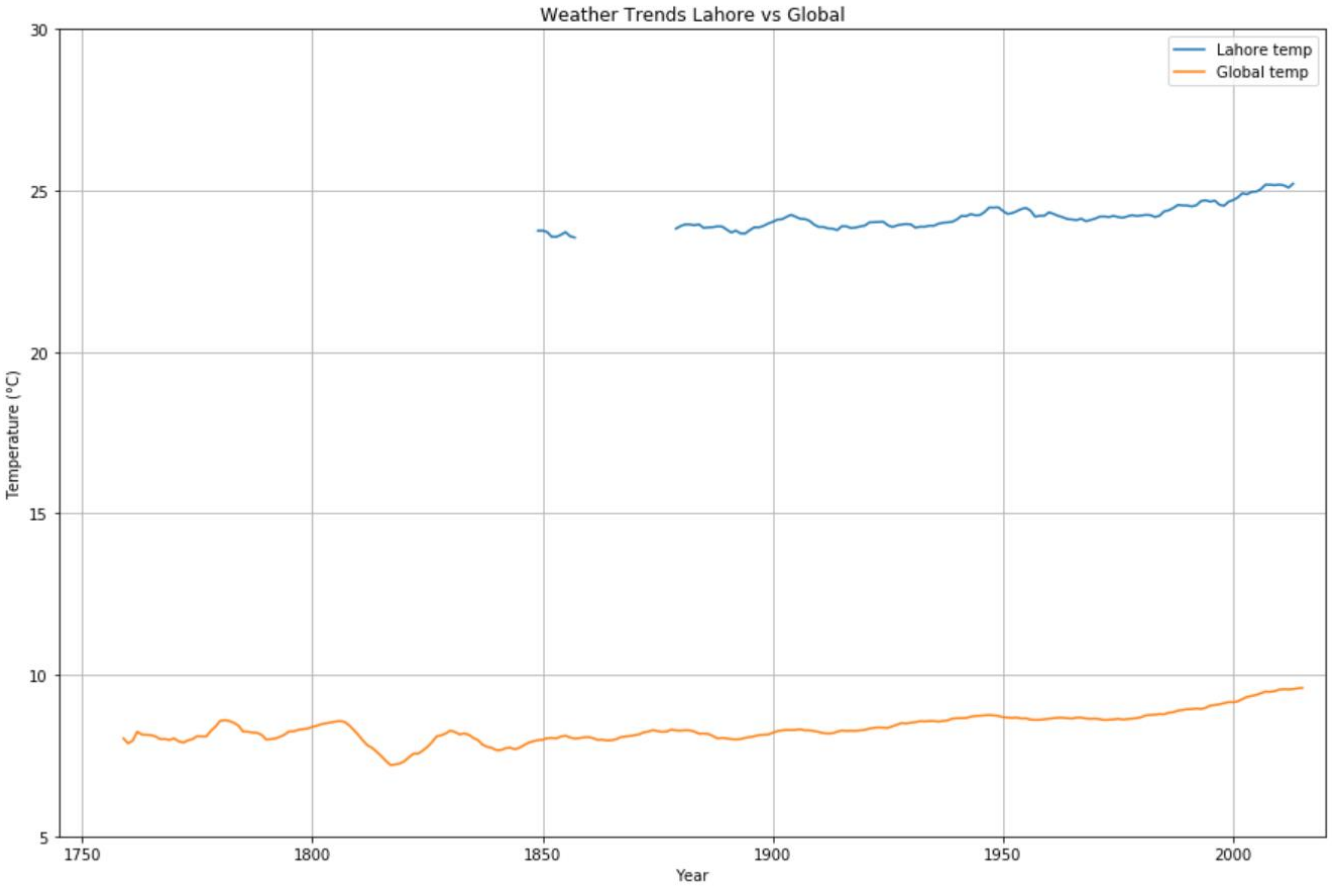
Why the window size or moving average calculating for 10 years or decade?

As the dispersion of data will be reduced nearly by 25%, and lesser the dispersion of data the trend line will be smoother.

```
df_CityData['m_avg_temp'] = df_CityData['avg_temp'].rolling(10).mean()
df_GlobalData['m_avg_temp'] = df_GlobalData['avg_temp'].rolling(10).mean()
```

Now plotting the data of moving averages calculated for 10 years span of both local and global temperatures (ºC)

```
plt.figure(figsize=[15,10])
plt.plot(df_CityData['year'],
         df_CityData['m_avg_temp'],
         label='Lahore temp')
plt.plot(df_GlobalData['year'],
         df_GlobalData['m_avg_temp'],
         label='Global temp')
plt.legend()
plt.grid(True)
plt.axis([1745, 2020, 5, 30])
plt.title('Weather Trends Lahore vs Global')
plt.xlabel('Year')
plt.ylabel('Temperature (°C)')
plt.show()
```



The moving average temperature (°C) drops for local & global data is 23.53°C & 7.2°C respectively. And the highest moving average temperature (°C) for global & local data is 9.59°C & 25.2°C respectively.

Imputing Missing values in Raw Data & Visualizing

The inconsistencies in data leads to the misleading decisions. To minimize the risks leading to inapropriate decisions, I applied imputer to the missing data for the local temperatures (°C).

Finding the count of missing values in each column of data for local temperatures (°C) and global temperatures (°C) as well. Code and results shown side by side.

```
1. df_CityData.isnull().sum()
```

```
year           0
city           0
country        0
avg_temp      22
m_avg_temp    54
dtype: int64
```

```
1. df_GlobalData.isnull().sum()
```

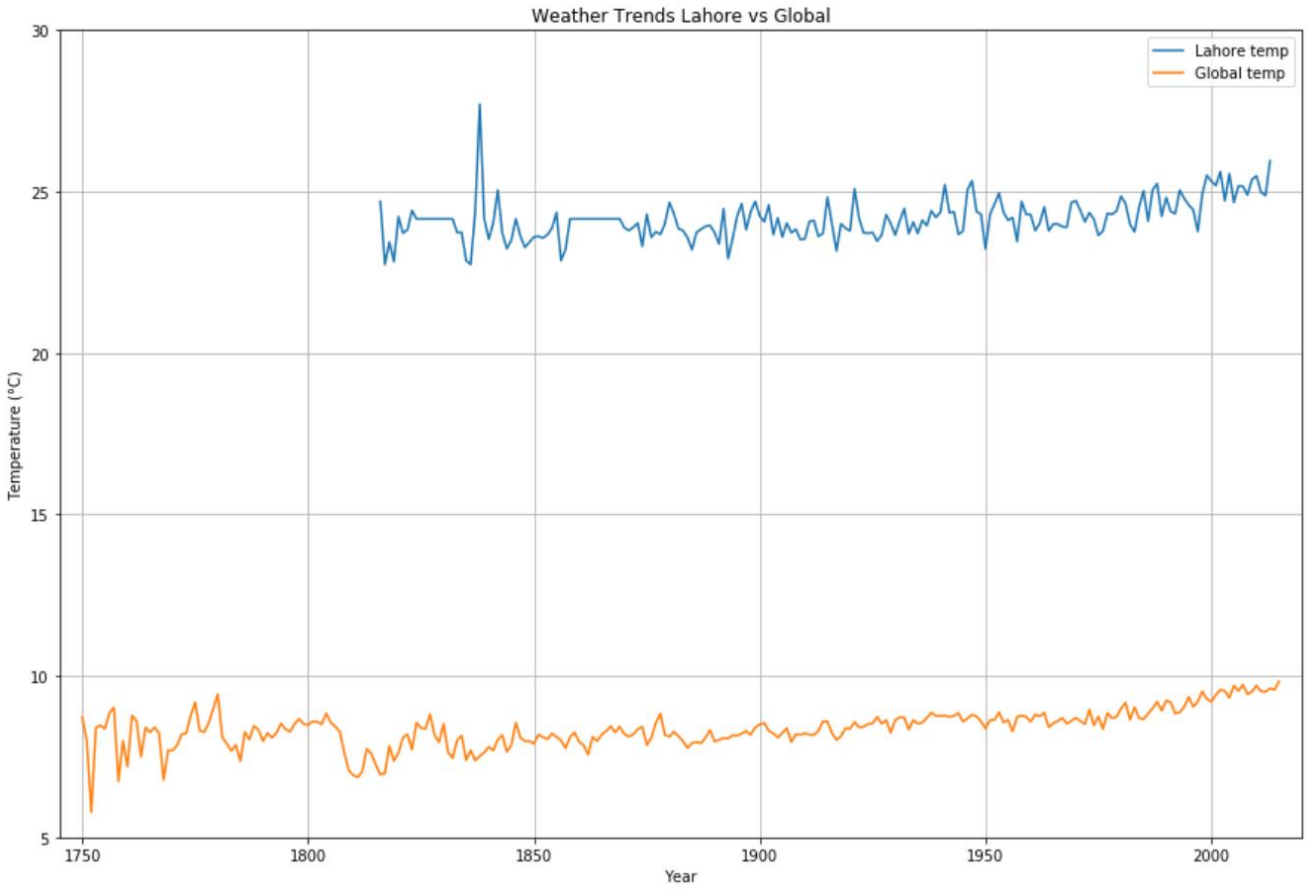
```
year          0
avg_temp      0
m_avg_temp    9
dtype: int64
```

You can see local temperatures (ºC) have ‘22’ missing values in ‘avg_temp’ which results in missing of ‘54’ values for ‘m_avg_temp’. These missing values breaks the continuity of data trend in chart. Now I’ll impute the missing values with the **mean** of the ‘avg_temp’ in the new data frame.

```
df_CityData_imp = df_CityData.copy()
df_CityData_imp['imp_avg_temp'] =
df_CityData_imp['avg_temp'].fillna(df_CityData['avg_temp'].mean())
```

After replacing missing values with the mean temperature. Plotting the data.

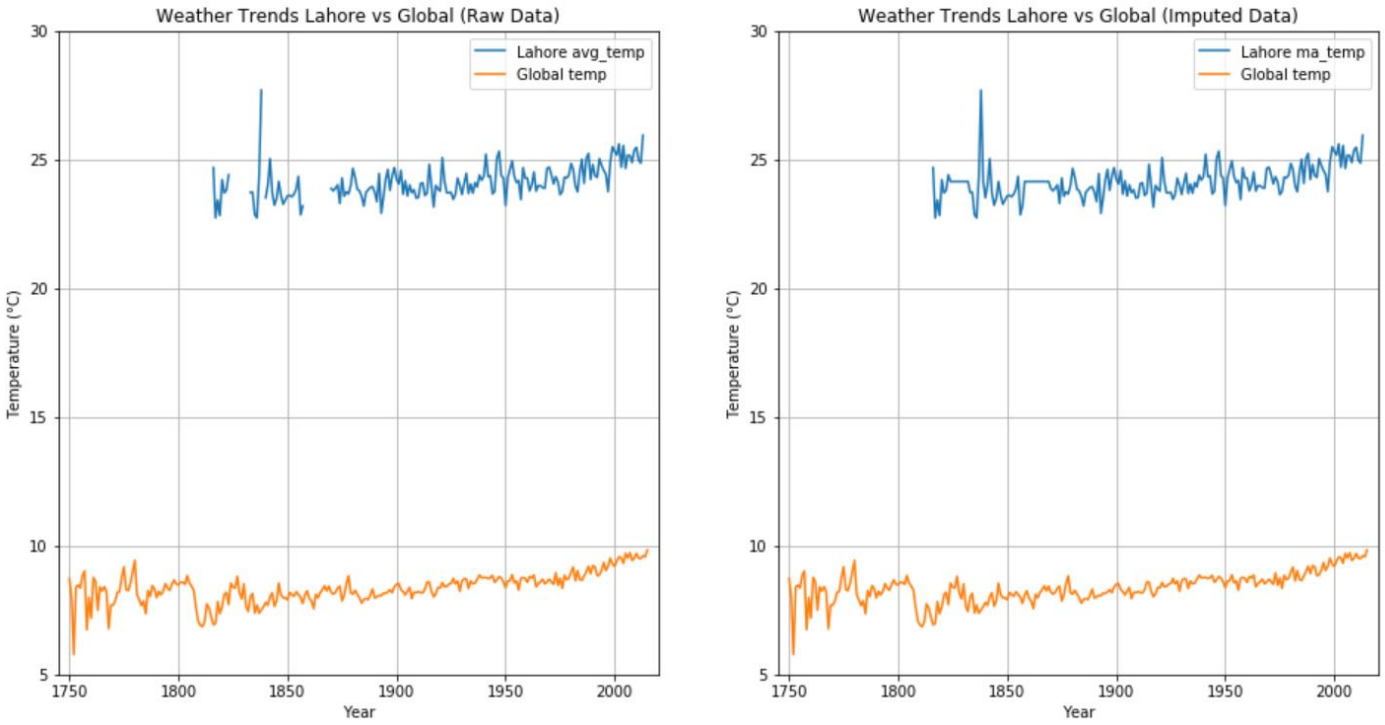
```
plt.figure(figsize=[15,10])
plt.plot(df_CityData_imp['year'],
         df_CityData_imp['imp_avg_temp'],
         label='Lahore temp')
plt.plot(df_GlobalData['year'],
         df_GlobalData['avg_temp'],
         label='Global temp')
plt.legend()
plt.grid(True)
plt.axis([1745, 2020, 5, 30])
plt.title('Weather Trends Lahore vs Global')
plt.xlabel('Year')
plt.ylabel('Temperature (°C)')
plt.show()
```



Now you can see the trend line is continuous from year 1816 to year 2013 for ‘Lahore temp’ despite of its spikes and fluctuations.

Now, plotting raw data along with the imputed data side by side.

```
plt.figure(figsize=(16,8))

plt.subplot(121)
plt.plot(df_CityData['year'],
         df_CityData['avg_temp'],
         label='Lahore avg_temp')
plt.plot(df_GlobalData['year'],
         df_GlobalData['avg_temp'],
         label='Global temp')
plt.legend()
plt.grid(True)
plt.axis([1745, 2020, 5, 30])
plt.title('Weather Trends Lahore vs Global (Raw Data)')
plt.xlabel('Year')
plt.ylabel('Temperature (°C)')

plt.subplot(122)
plt.plot(df_CityData_imp['year'],
          df_CityData_imp['imp_avg_temp'],
          label='Lahore ma_temp')
plt.plot(df_GlobalData['year'],
          df_GlobalData['avg_temp'],
          label='Global temp')
plt.legend()
plt.grid(True)
plt.axis([1745, 2020, 5, 30])
plt.title('Weather Trends Lahore vs Global (Imputed Data)')
plt.xlabel('Year')
plt.ylabel('Temperature (°C)')
plt.show()
```



Calculating Moving Averages on Imputed Data & Visualizing

Calculating moving averages for both local & global data temperatures (ºC) for the span of a decade.

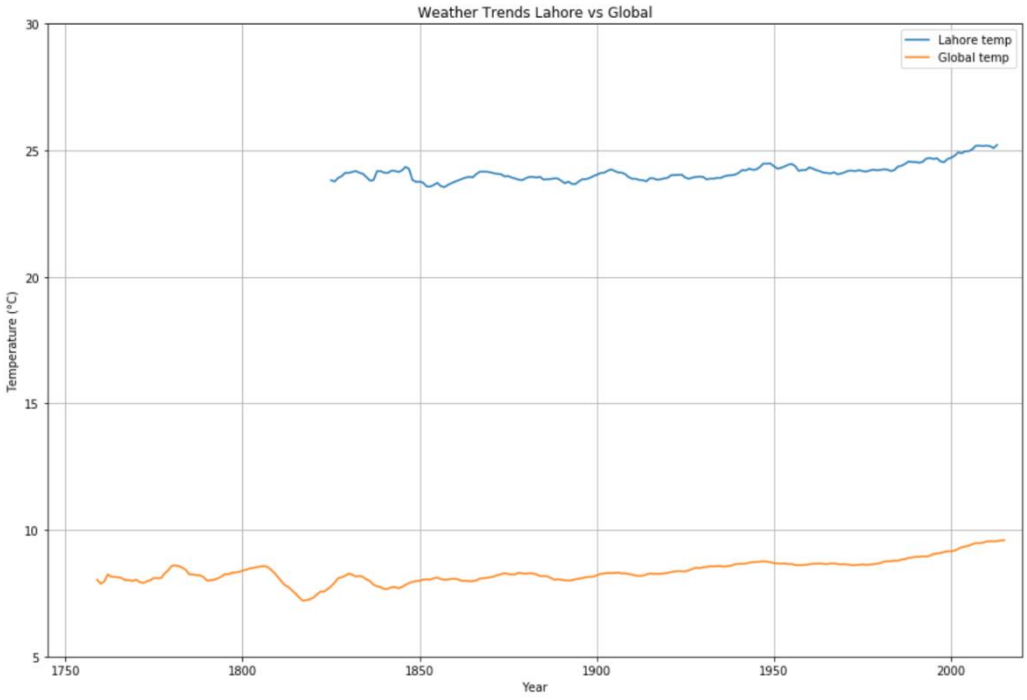
Why the window size or moving average calculating for 10 years or decade?

As the dispersion of data will be reduced nearly by 25%, and lesser the dispersion of data the trend line will be smoother.

```
df_CityData_imp['m_imp_avg_temp'] =
df_CityData_imp['imp_avg_temp'].rolling(10).mean()
```

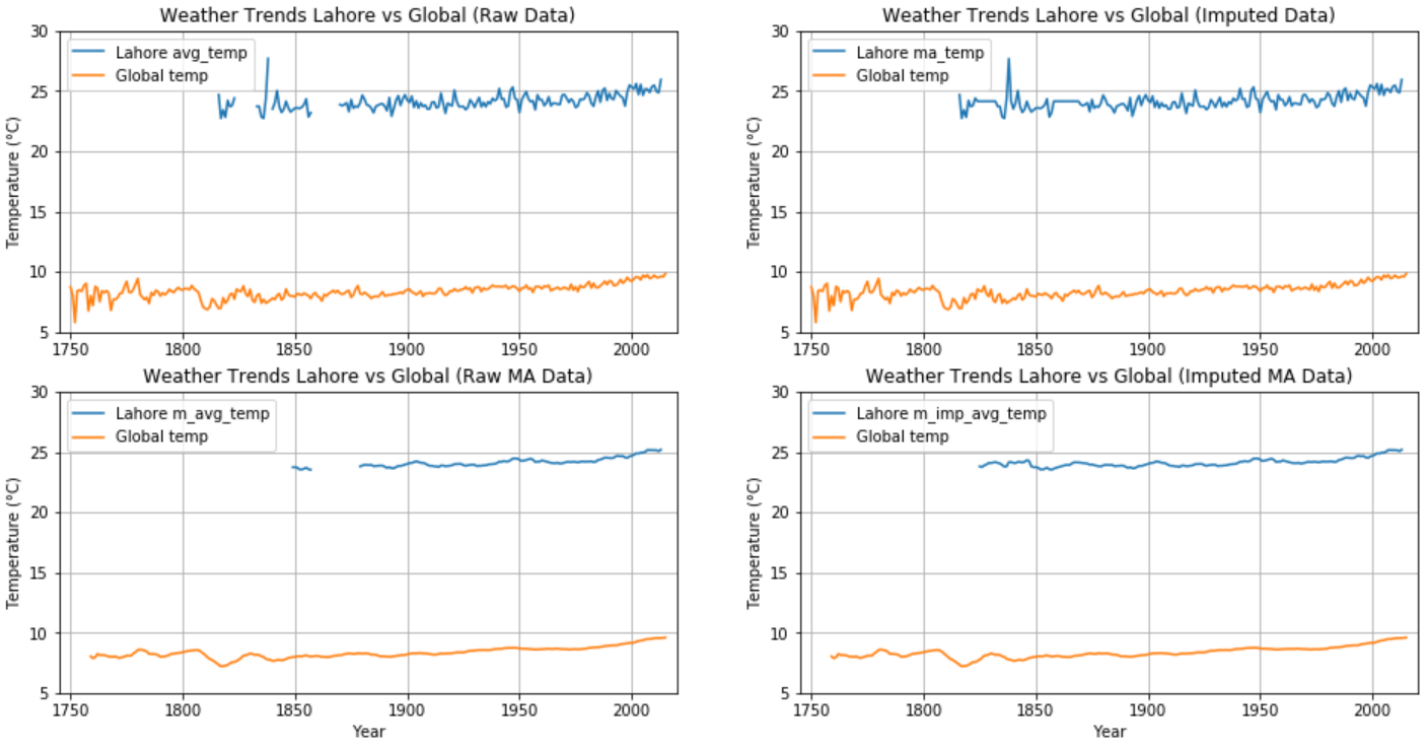
Now plotting the data with no missing values and with a moving average span of a decade.

```
plt.figure(figsize=[15,10])
plt.plot(df_CityData_imp['year'],
         df_CityData_imp['m_imp_avg_temp'],
         label='Lahore temp')
plt.plot(df_GlobalData['year'],
         df_GlobalData['m_avg_temp'],
         label='Global temp')
plt.legend()
plt.grid(True)
plt.axis([1745, 2020, 5, 30])
plt.title('Weather Trends Lahore vs Global')
plt.xlabel('Year')
plt.ylabel('Temperature (°C)')
plt.show()
```



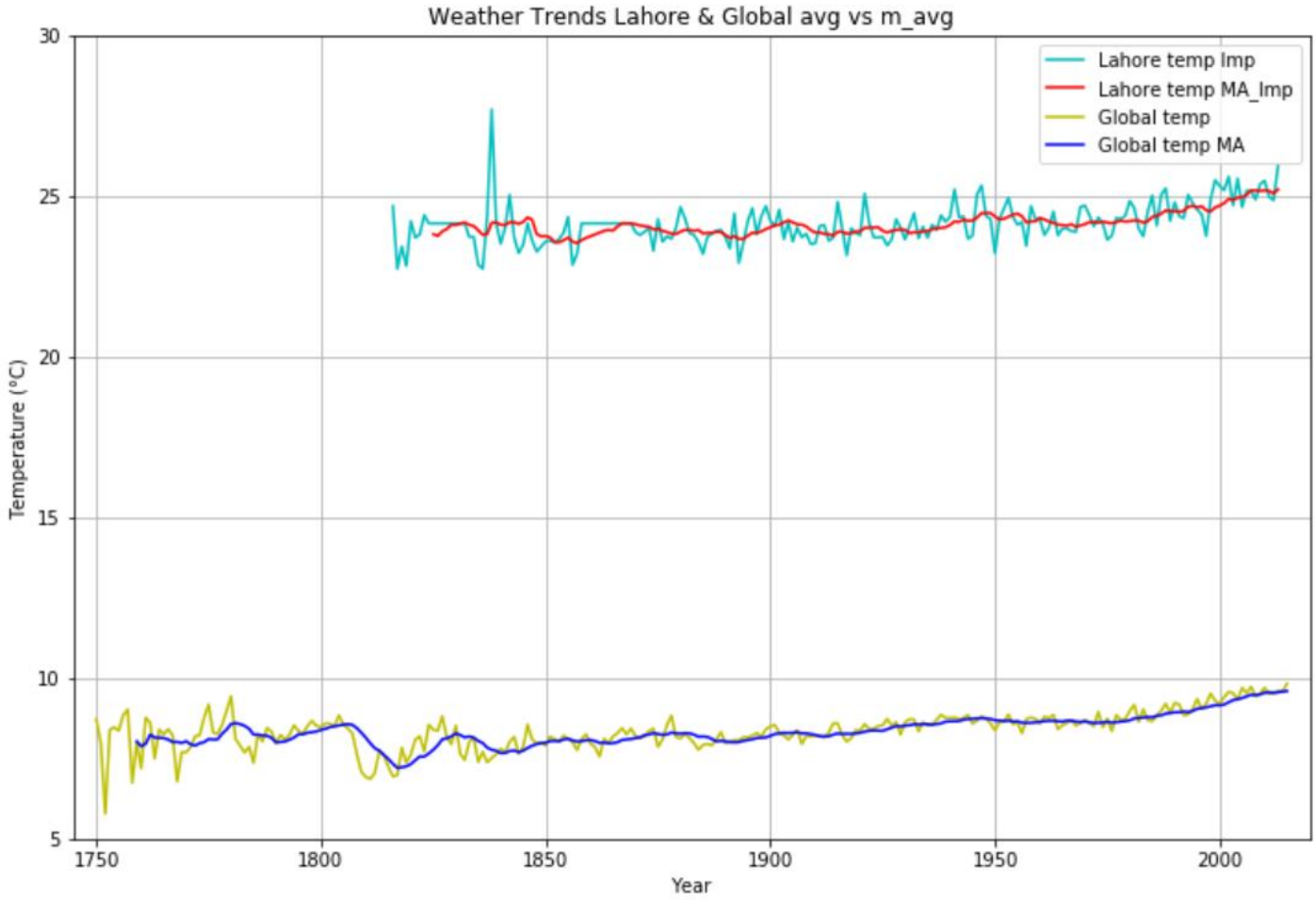
All the plots in one chart raw data along with its moving averages and imputed data along with its moving averages.

```
plt.figure(figsize=(16,8))

plt.subplot(221)
plt.plot(df_CityData['year'],
         df_CityData['avg_temp'],
         label='Lahore avg_temp')
plt.plot(df_GlobalData['year'],
         df_GlobalData['avg_temp'],
         label='Global temp')
plt.legend()
plt.grid(True)
plt.axis([1745, 2020, 5, 30])
plt.title('Weather Trends Lahore vs Global (Raw Data)')
plt.ylabel('Temperature (°C)')

plt.subplot(222)
plt.plot(df_CityData_imp['year'],
         df_CityData_imp['imp_avg_temp'],
         label='Lahore ma_temp')
plt.plot(df_GlobalData['year'],
         df_GlobalData['avg_temp'],
         label='Global temp')
plt.legend()
plt.grid(True)
plt.axis([1745, 2020, 5, 30])
plt.title('Weather Trends Lahore vs Global (Imputed Data)')
plt.ylabel('Temperature (°C)')

plt.subplot(223)
plt.plot(df_CityData['year'],
         df_CityData['m_avg_temp'],
         label='Lahore m_avg_temp')
plt.plot(df_GlobalData['year'],
         df_GlobalData['m_avg_temp'],
         label='Global temp')
plt.legend()
plt.grid(True)
plt.axis([1745, 2020, 5, 30])
plt.title('Weather Trends Lahore vs Global (Raw MA Data)')
plt.xlabel('Year')
plt.ylabel('Temperature (°C)')

plt.subplot(224)
plt.plot(df_CityData_imp['year'],
         df_CityData_imp['m_imp_avg_temp'],
         label='Lahore m_imp_avg_temp')
plt.plot(df_GlobalData['year'],
         df_GlobalData['m_avg_temp'],
         label='Global temp')
plt.legend()
plt.grid(True)
plt.axis([1745, 2020, 5, 30])
plt.title('Weather Trends Lahore vs Global (Imputed MA Data)')
plt.xlabel('Year')
plt.ylabel('Temperature (°C)')
plt.show()
```



Last, but not least. Plotting the data with no missing values and average temperatures along with calculated moving averages.

```
plt.figure(figsize=[12,8])
plt.plot(df_CityData_imp['year'],
         df_CityData_imp['imp_avg_temp'],
         label='Lahore temp Imp', color= 'c')
plt.plot(df_CityData_imp['year'],
         df_CityData_imp['m_imp_avg_temp'],
         label='Lahore temp MA_Imp', color= 'r')
plt.plot(df_GlobalData['year'],
         df_GlobalData['avg_temp'],
           label='Global temp', color= 'y')
 plt.plot(df_GlobalData['year'],
           df_GlobalData['m_avg_temp'],
           label='Global temp MA', color= 'b')
 plt.legend()
 plt.grid(True)
 plt.axis([1745, 2020, 5, 30])
 plt.title('Weather Trends Lahore & Global avg vs m_avg')
 plt.xlabel('Year')
 plt.ylabel('Temperature (°C)')
 plt.show()
```


Weather Trends Lahore & Global avg vs m_avg
Lahore temp Imp
Lahore temp MA_Imp
Global temp
Global temp MA
Temperature (°C)
Year
30
25
20
15
10
5
1750
1800
1850
1900
1950
2000

CONCLUSION

From the extraction of the data from the database, transforming it into a csv file and loading it into a data-frame. Then performing the analysis with several perspectives. I come with some discrete finding from the data which will govern your decisions.

- Today, we are standing at the highest level of temperatures (°C) globally from last two centuries.
- Lahore's temperatures (°C) are relatively twice than the global average temperatures (°C).
- Over few decades, local temperatures (°C) are increasing consistently.
- Over time, the city temperatures have changed similarly with the global average.
- The world is getting hotter day by day, as the global and local temperatures (°C) exhibit an uphill trend.
- Over few decades, global temperatures (°C) are increasing consistently.

REFERENCES

1. https://www.datacamp.com/community/tutorials/moving-averages-in-pandas (moving averages)
2. https://towardsdatascience.com/stop-using-mean-to-fill-missing-data-678c0d396e22 (imputing data)
3. https://matplotlib.org/3.1.1/tutorials/introductory/pyplot.html#sphx-glr-tutorials-introductory-pyplot-py (plotting the data)
4. https://knowledge.udacity.com/questions/372272 (Plotting data)