

Exploring Weather Trends

MAY 31, 2020

Mridul Sharma

New Delhi, 201301

Overview

I have been provided the temperature database from the udacity's project space from where I have extracted the data related to global temperature and my city temperature. I analyzed the temperature around the global with the city I live in by extracting the data from the database.

Goals

1. Extraction of data from the database and export to CSV file
2. Making a chart visualization based on extracted data
3. Observation based on chart

Tools Used:

1. **SQL:** To extract the data from the database
2. **Python:** For calculating moving average and plotting line chart (Pandas library for data processing and **matplotlib** for data visualization.
3. **Jupyter Notebook:** For writing python code and making observations
4. **Google Sheets:** Having a look at the data and writing project

STEP 1 - Extraction of Data from provided Database

```
SELECT *  
  
FROM city_list  
  
WHERE country = 'India'  
  
{NOTE: i think the most preferable state is new delhi}  
  
SELECT  
  
c.year "Year",  
  
c.avg_temp "New Delhi",  
  
g.avg_temp "Global"  
  
FROM global_data g  
  
JOIN city_data c ON c.year = g.year  
  
WHERE c.city = 'New Delhi'
```

The data was obtained as a CSV file named results.csv and the same was used for further analysis.

STEP 2 -Analyzing the CSV File

I selected Python for data processing and visualisation for this task.

I chose pandas library for data processing and matplotlib for visualisation. All operations were done in Jupyter Notebooks to create this document.

A- Importing the libraries

```
import matplotlib.pyplot as plt
```

```
import pandas as pd
```

```
%matplotlib inline
```

B- Creating Pandas Dataframe

```
py = pd.read_csv("results.csv")
```

C- Getting insights from the data

```
py.head()
```

	Year	New Delhi	Global
0	1796	25.03	8.27
1	1797	26.71	8.51
2	1798	24.29	8.67
3	1799	25.28	8.51
4	1800	25.21	8.48

```
py.columns.unique()
```

Removing Whitespace from Column names:

```
py.columns = ['Year', 'NewDelhi', 'Global']
```

```
py.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 218 entries, 0 to 217
Data columns (total 3 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Year        218 non-null    int64
1   NewDelhi    201 non-null    float64
2   Global      218 non-null    float64
dtypes: float64(2), int64(1)
memory usage: 5.2 KB
```

```
py.describe()
```

	Year	NewDelhi	Global
count	218.000000	201.000000	218.000000
mean	1904.500000	25.166269	8.403532
std	63.075352	0.594003	0.548662
min	1796.000000	23.700000	6.860000
25%	1850.250000	24.800000	8.092500
50%	1904.500000	25.140000	8.415000
75%	1958.750000	25.550000	8.727500
max	2013.000000	26.710000	9.730000

Removing NA values using dropna function:

```
py.dropna(axis=0, how='any', thresh=None, subset=None, inplace=True)
```

D- Calculating Moving Average using Rolling and Mean function

Using Window=10 sets a 10-year window for the moving average. This value was used because it smoothed the result chart optimally for further analysis among a set of other values.

```
py['NewDelhi'] = py.rolling(window=10)['NewDelhi'].mean()  
py['Global'] = py.rolling(window=10)['Global'].mean()
```

E- Configuring Parameters of the Graph

```
params = {  
    'legend.fontsize': 24,  
    'figure.figsize': (18, 10),  
    'axes.labelsize': 20,  
    'axes.titlesize': 20,  
    'xtick.labelsize': 'x-large',  
    'ytick.labelsize': 'x-large'  
}  
  
plt.rcParams.update(params)  
plt.title("\nLine Chart of New Delhi vs. Global Temperature")  
plt.ylabel("Temperature")  
plt.xlabel("Year")  
plt.xticks(np.arange(1800, 2013, step=10))  
plt.plot(py.Year, py.NewDelhi, linewidth=4, color='Red', ls = 'dashdot', label='New Delhi')  
plt.plot(py.Year, py.Global, linewidth=4, color='green', ls = 'dashdot', label='Global')  
plt.grid(True)  
plt.legend(loc='upper left', bbox_to_anchor=(1, 1))  
plt.show()
```

F- Plotting the Line Chart

```
plt.plot(py.Year, py.NewDelhi, linewidth=4, color='Red', ls = '-', label='New Delhi')
```

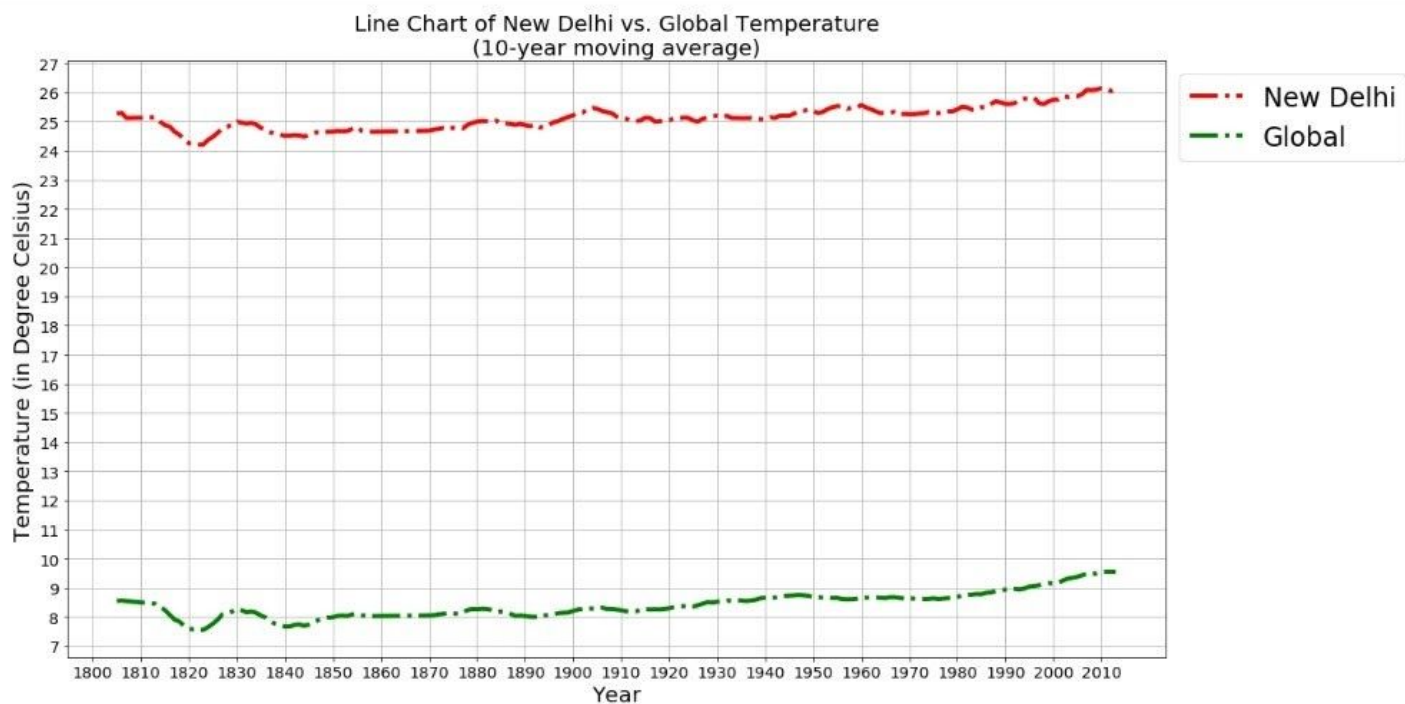
```
plt.plot(py.Year, py.Global, linewidth=4, label='Global')
```

```
plt.grid(True)
```

Legend:

```
plt.legend(loc='upper left', bbox_to_anchor=(1, 1))
```

```
plt.show()
```



OBSERVATIONS

1. In contrary to Global average, the city's temperature is considerably higher i.e. on an average, it is almost 17 degrees hotter than global average.
2. However, after observing the graph, it is clear that the temperature of the city as well as the Global temperature have risen steadily throughout the years.
3. The temperature has risen 3 degrees while the global temperature has also increased by an equivalent amount. This is evident from the Min and Max temperature recordings.
4. The ratio of the temperatures of the city and global average is also almost equal to 3 degrees i.e. the city's average temperature is almost 3 times the global average.
5. The Correlation coefficient between city's and Global temperature is 0.76 so we can conclude it is a positive but weak relationship and hence the temperature of the city cannot be estimated based on Global temperature.

References:

https://en.wikipedia.org/wiki/Climate_of_Delhi

https://www.yr.no/place/India/Delhi/New_Delhi/statistics.html