Data Analyst Nanodegree

Udacity

Explore Weather Trends

By

Shivam Jain

DATA EXTRACTION FROM DATABASE

(Using SQL)

1.To find out all the cities of India:

Select * from city_list where country='India';

2.To alter the name "avg_temp" in both city_data and global_data for better understanding:

Alter table city_data rename column avg_temp to city_avg_temp;
Alter table global_data rename column avg_temp to global_avg_temp;

3.To join two tables "global_data" and "city_data" and extract three columns of relevant data i.e. "year", "global_avg_temp" and "city_avg_temp" for a city (Nagpur) in India:

Select gd.year, gd.global_avg_temp, cd.city_avg_temp from global_data AS gd INNER JOIN city_data as cd ON gd.year = cd.year where city = 'Nagpur';

I performed the above SQL commands to extract data from the database in the form of a schema of three columns, namely "year", "city_avg_temp" and "global_avg_temp" . After downloading the csv file I performed analysis using python.

ANALYSIS

- After downloading csv file I imported that in my jupyter notebook as follows:

```
In [21]: import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
         %matplotlib inline
```

```
In [22]: df = pd.read_csv(r"C:\Users\shivam\Desktop\Udatasets\final.csv")
         df.head()
```

Out[22]:

	year	global_avg_temp	city_avg_temp
0	1796	8.27	25.43
1	1797	8.51	26.58
2	1798	8.67	24.63
3	1799	8.51	25.71
4	1800	8.48	25.69

- Now to check the data for any missing values or NaN or any kind of discrepancies I followed this:

```
In [23]: df.describe()
```

Out[23]:

	year	global_avg_temp	city_avg_temp
count	218.000000	218.000000	211.000000
mean	1904.500000	8.403532	25.638768
std	63.075352	0.548662	0.625609
min	1796.000000	6.860000	21.100000
25%	1850.250000	8.092500	25.325000
50%	1904.500000	8.415000	25.640000
75%	1958.750000	8.727500	26.015000
max	2013.000000	9.730000	27.140000

- From the above table, I could conclude that the “year” and “global_avg_temp” had same number of values but the column for “city_avg_temp” had 7 missing values. So, in order to balance the data, I decided to drop the rows with “NaN” for the “city_avg_temp” values.

- I coded a function to calculate the Moving Averages for “global_avg_temp” and “city_avg_temp” columns to see the trends in temperature readings over years using rolling function from pandas as follows:

```
In [24]: def moving_averages(data=None, window_size=None):
             df_roll_avg = data.rolling(window=window_size, center=False, on='year').mean().dropna()
             return df_roll_avg
```

```
In [26]: roll_window_size = 9

         mov_avg = moving_averages(data=df, window_size=roll_window_size)
```

- In above function, the inputs to the “moving_averages” function are “data” i.e. the original dataframe and “window_size” i.e. size of the moving window. This is the number of observations used for calculating the statistic. Each window is of a fixed size. Moreover, this function removes the rows for which the “city_avg_temp” is a “NaN” by using “dropna()”.
- For this data, I tried plotting the graph for different values for the rolling window sizes for the smoothness and then set it to a value of 9.
- The data then looked like this:

```
In [27]: mov_avg.head(10)
```

Out[27]:

	year	global_avg_temp	city_avg_temp
8	1804	8.550000	25.668889
9	1805	8.582222	25.711111
10	1806	8.573333	25.613333
11	1807	8.530000	25.654444
25	1821	7.488889	24.612222
26	1822	7.538889	24.646667
27	1823	7.553333	24.767778
28	1824	7.698889	24.907778
29	1825	7.860000	25.056667
30	1826	8.013333	25.196667

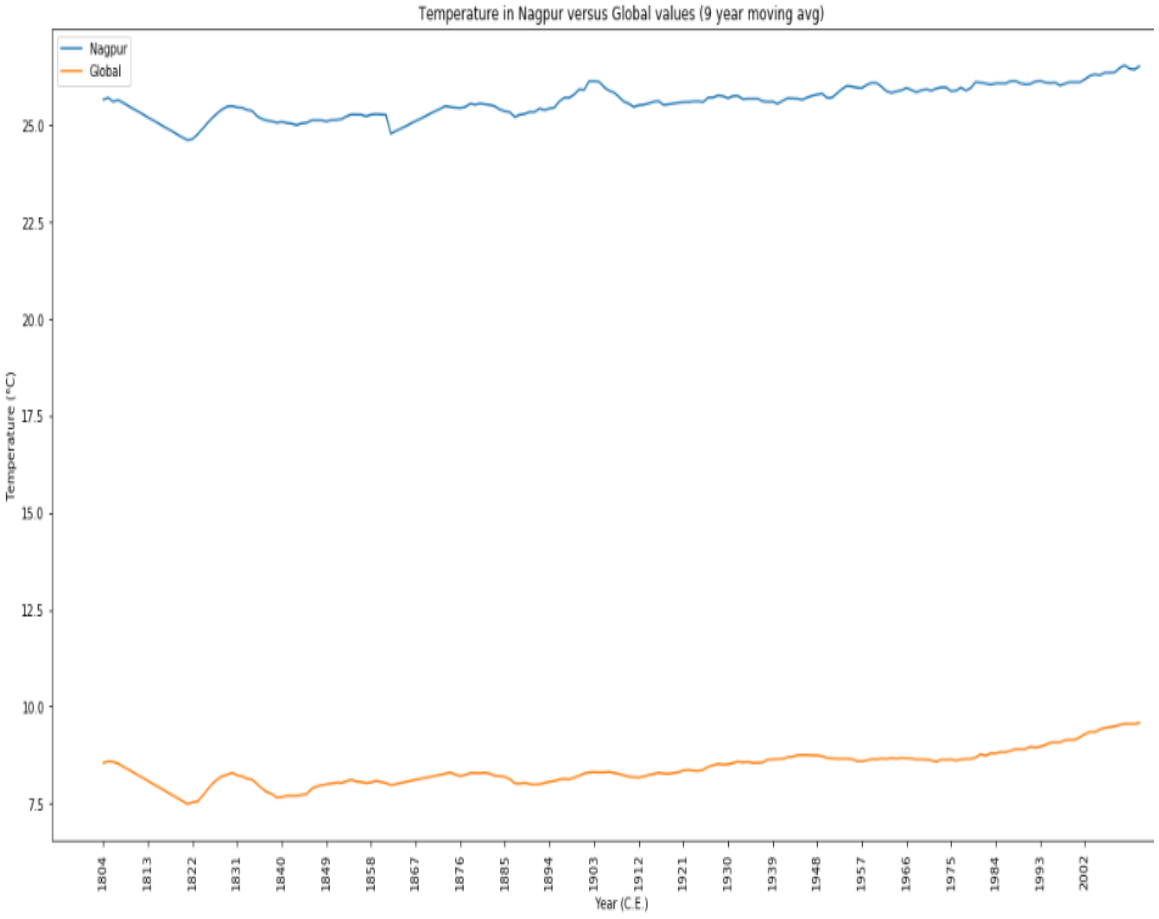
- Then I plotted the moving average temperature using matplotlib:

In [29]:
```
min_year = mov_avg['year'].iloc[0]
max_year = mov_avg['year'].iloc[176]

plt.figure(figsize=(20,10))
plt.plot(mov_avg['year'], mov_avg['city_avg_temp'], label='Nagpur')
plt.plot(mov_avg['year'], mov_avg['global_avg_temp'], label='Global')
plt.legend(loc='best')

plt.xticks(np.arange(min_year, max_year, 9.0), rotation='vertical')
plt.xlabel("Year (C.E.)")
plt.ylabel("Temperature (°C)")
plt.title("Temperature in Nagpur versus Global values ({} year moving avg)".format(roll_window_size))
```

Out[29]: Text(0.5, 1.0, 'Temperature in Nagpur versus Global values (9 year moving avg)')



Observations

- From the above plot, we can see that the global moving average temperature lies between 7.15 to 8.67 Degrees Celsius.
- For Nagpur, the moving average temperature lies in the range of 24.2 to 26.7 Degree Celsius.
- During the period of 1806 to 1821, both the Global as well as Delhi temperatures went down.
- Since then, there have been some small ups and downs in both the Global and Delhi temperatures as well as their moving averages but mostly, the temperature has been increasing consistently.
- The above plot shows that the moving average temperature for Delhi is more than the Global moving average temperature which infers that Delhi is hotter in comparison of the Global temperature.
- This proves that both the Global and Delhi's temperature is rising at a high rate based on the increasing trend as seen in the plot above starting from about 1846 till today.