EXPLORE WEATHER TRENDS

UDACITY-DATA ANALYST

BERLIN 11/11/2019

Step One: Extract Data from Database with SQL

1. **Find the city where I live**

```
select*
from city_list
where city like 'Berlin' and country like 'Germany';
```

#I live in Berlin, the capital of Germany. It's no surprise that I could find the city easily on the "city_list".

2. **Select data with variable "Berlin" and download CSV file named "city_data"**

```
select*
from city_data
where city like 'Berlin';
```

3. **Select global data and download CSV file named "global_data"**

```
select*
from global_data
```

Step Two: Create Line Chart with Python

1. **Set up**

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
```

2. **Import datasets saved on local computer**

```
global_data = pd.read_csv("global_data.csv")
city_data = pd.read_csv("city_data.csv")
```

3. **Calculate moving average of temperature by using rolling function and setting window value of 10 (years)**

```
mA_city = city_data.cat.rolling(10).mean()
mA_global = global_data.avg_temp.rolling(10).mean()
```

#10 is choosed randomly, it turned out that the lines have been smoothed so I did not

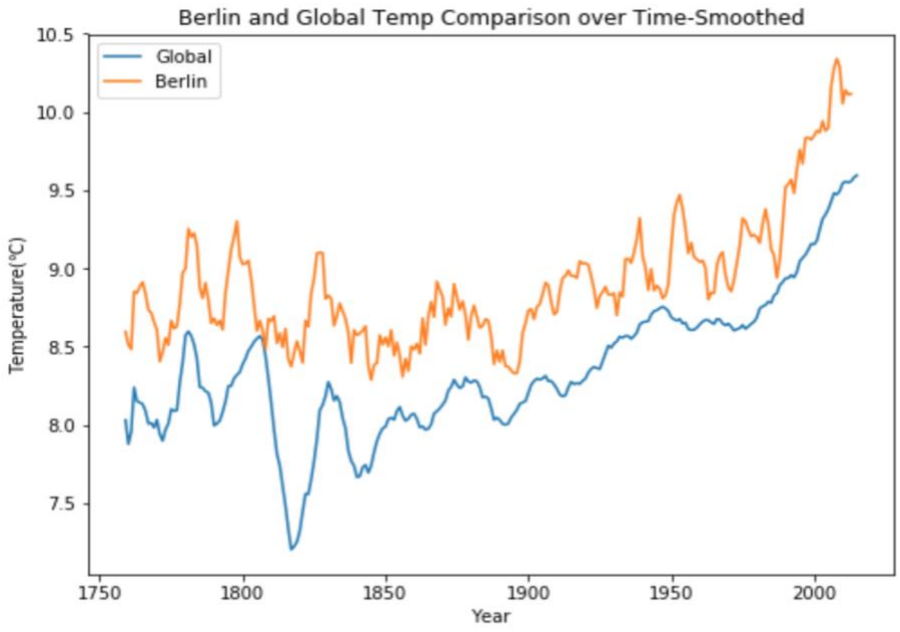
try different window value.

4. Create line chart

I used Jupyter Notebook to write codes and visualize data considering facility of managing large datasets (with time series as variable) with numpy and pandas. Rolling function could be easily applied to calculate moving average and generate graph afterwards.

(Code Below)

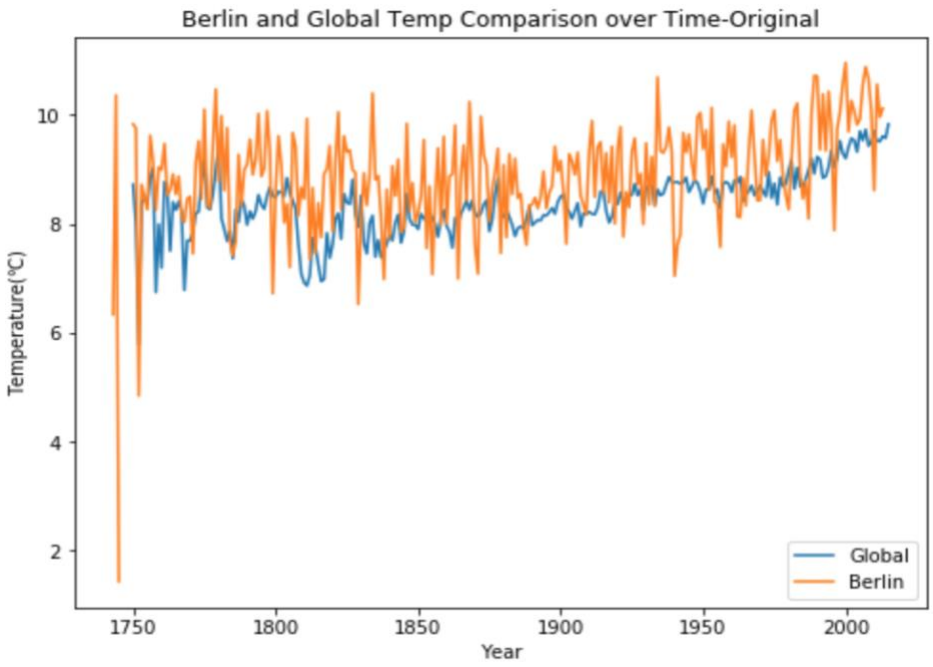
```
plt.figure(figsize=(8,6)) #The size of the figure has been adjusted several times to find good setting
plt.title("Berlin and Global Temp Comparison over Time-Smoothed")
plt.plot(global_data.year, mA_global, label="Global")
plt.plot(city_data.year, mA_city, label="Berlin")
plt.legend()
plt.xlabel("Year")
plt.ylabel("Temperature(℃)")
plt.show()
```



5. Extra: Original version of comparison without rolling

(Code Below)

```
plt.figure(figsize=(8,6))
plt.title("Berlin and Global Temp Comparison over Time-Original")
plt.plot(global_data.year, global_data.avg_temp, label="Global")
plt.plot(city_data.year, city_data.cat, label="Berlin")
plt.legend()
plt.xlabel("Year")
plt.ylabel("Temperature(℃)")
plt.show()
```



Step Three: Observations

1. The average temperature in Berlin has overpassed 10 degrees centigrade for the first time around Y2000.
2. The average temperature in Berlin is constantly higher than that observed globally and the difference has been consistent.
3. The temperature in Berlin has recorded more brutal fluctuations compared to global average.

4. The figure reveals global warming trend since Y1750 (roughly increased by 1.5 degrees centigrade).