Data Analyst Nanodegree

Project-1:

Explore Weather Trends

Ayush Mathur
Term-1, Project-1
Udacity- Data Analyst Nanodegree

Overview:

In this project, I have analysed the local temperature i.e. Bengaluru’s temperature data and global temperature data. I extracted the relevant data from Udacity’s SQL Workspace. I used Jupyter notebook and Python programming language to manipulate and visualize the data.

Goals:

1. Extraction of data from database and export it to CSV.
2. Import CSV and manipulate data.
3. Visualize the data (Create Line Graphs).
4. Make observations.

Steps:

1. The data was extracted using the following SQL query. Here I have joined the two tables namely ‘global_data’ and ‘city_data’ on the column ‘year’ of the tables. Since the tables contained columns with same name, the columns have been renamed to c_avg_temp and g_avg_temp. The data was saved to a CSV file named ‘results.csv’.

   ```
   SELECT g.year, c.avg_temp c_avg_temp, g.avg_temp g_avg_temp
   FROM global_data g
   JOIN city_data c
   ON g.year = c.year AND c.city='Bangalore';
   ```

2. Next, I used python in Jupyter Notebook to read CSV file and store it as Dataframe named ‘dataf’. The function isna() is used to find the null values in the Dataframe and function sum() to have a total count of the null values.

   ```
   import numpy as np
   import pandas as pd
   dataf = pd.read_csv(r"E:\Udacity_Projects\Explore_Weather_Trends\results.csv")
   dataf.isna().sum()
   ```

   Output:

   ```
   year          0
   c_avg_temp    7
   g_avg_temp    0
   dtype: int64
   ```

3. Dataframe contains null values. Null values are present only in the ‘c_avg_temp’ column. Total number of rows is 218 and number of null rows is 7 which accounts to 3 percent. Hence, the rows with null values can be dropped. Otherwise I could have replaced those with the overall mean of the column using replace() and mean() functions. dropna() function is used to drop rows.

```
dataf1 = dataf.dropna(axis=0,inplace=False)
dataf1.reset_index(inplace=True)
```

4. Used the function isna() again to check if there is any null value left.

```
dataf1.isna().sum()
```

Output:

```
year          0
c_avg_temp    0
g_avg_temp    0
dtype: int64
```

5. I have calculated moving average using the functions rolling() and mean(). Rolling() is used to select a set of values and mean() to calculate the average of that set. Parameter window is used to define the size of the moving window. Parameter center is used to shift the average determined to the centre of the window.

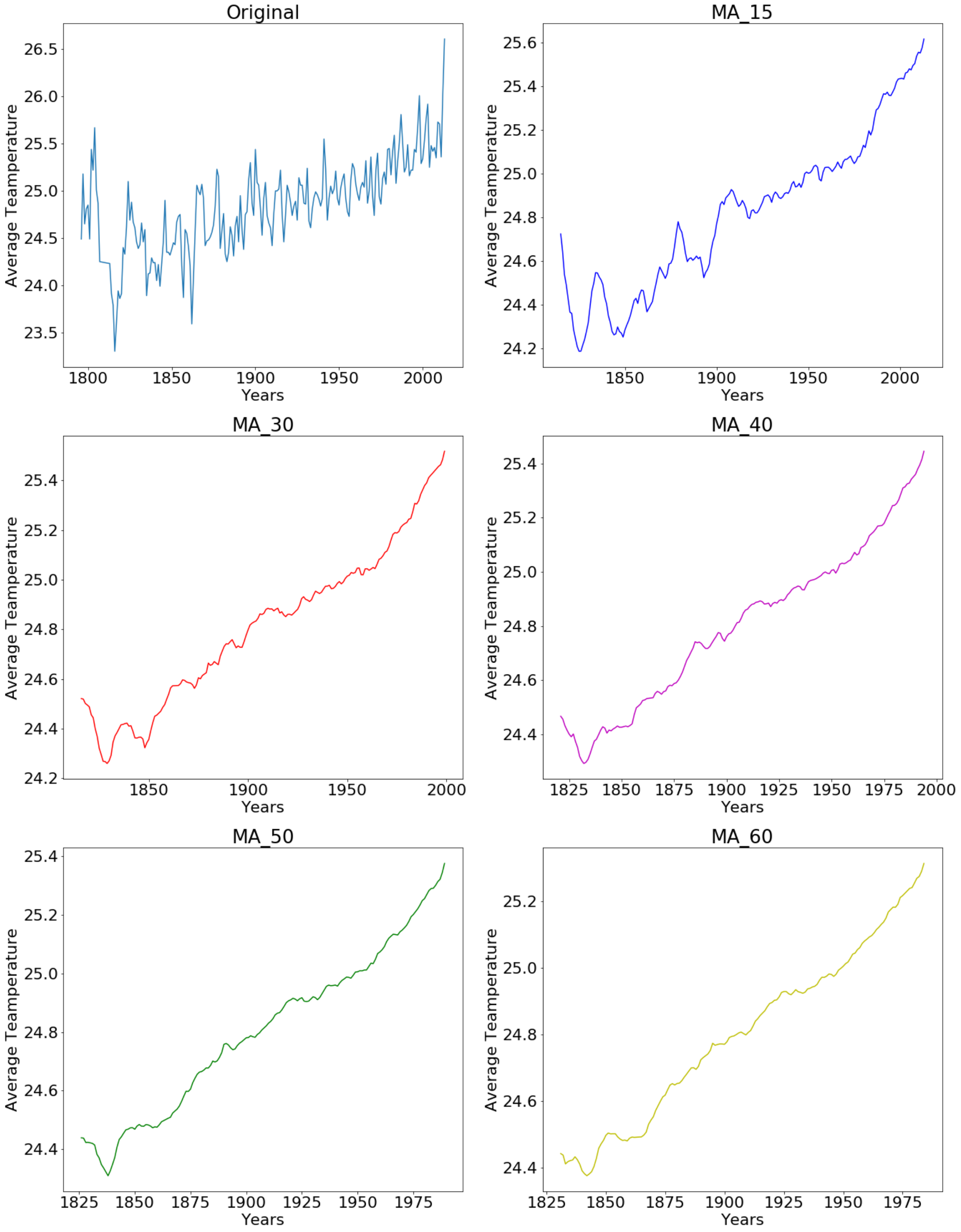
```
df40 = dataf1[['c_avg_temp','g_avg_temp']].rolling(40,center=True).mean()
df40['year'] = dataf1['year']
```

6. I have used different values for the moving window: 15, 30, 40, 50 and 60 and stored the results in dataframes df15, df30, df40 df50 and df60. Line graph were plotted for these dataframes to determine the reasonable value for moving window.

7. Python library matplotlib is used to plot graphs. Python Code for plotting graphs. I wrote a similar code to plot graphs for global average temperature using the same moving average values.

```
%matplotlib inline                      #Required to run matplotlib in Jupyter
import matplotlib.pyplot as plt         #Import Library

plt.figure(figsize=(12,18))             #Set the figure dimensions
plt.subplot(326)                        #Provide space for subplot
plt.plot(df60['year'],df60['c_avg_temp'],'y-')              #Plot 1st Graph for MA=60
plt.title('MA_60')                                          #Sub-plot Title
plt.xlabel('Years')                                         #X-Label of subplot
plt.ylabel('Average Teamperature')                          #Y-Label of subplot

plt.subplot(325)                                            #Subplot for MA=50
plt.plot(df50['year'],df50['c_avg_temp'],'g-')
plt.xlabel('Years')
plt.ylabel('Average Teamperature')
plt.title('MA_50')

plt.subplot(324)                                            #Subplot for MA=40
plt.plot(df40['year'],df40['c_avg_temp'],'m-')
plt.xlabel('Years')
plt.ylabel('Average Teamperature')
plt.title('MA_40')

plt.subplot(323)                                            #Subplot for MA=30
plt.plot(df30['year'],df30['c_avg_temp'],'r-')
plt.xlabel('Years')
plt.ylabel('Average Teamperature')
plt.title('MA_30')
plt.subplot(322)                                            #Subplot for MA=15
plt.plot(df15['year'],df15['c_avg_temp'],'b-')
plt.xlabel('Years')
plt.ylabel('Average Teamperature')
plt.title('MA_15')
plt.subplot(321)                                            #Subplot without Moving Average
plt.plot(dataf1['year'],dataf1['c_avg_temp'])
plt.xlabel('Years')
plt.ylabel('Average Teamperature')
plt.title('Original')

plt.rcParams.update({'font.size': 22})                      #Update the Font Size of Figure Title
plt.suptitle('City Temperature using Different MA Values')               #Figure Title
plt.savefig(r"E:\Udacity_Projects\Explore_Weather_Trends\City_temp_ma.png")     #Save as png
plt.show()
```

City Temperature using Different MA Values



Original

MA_15

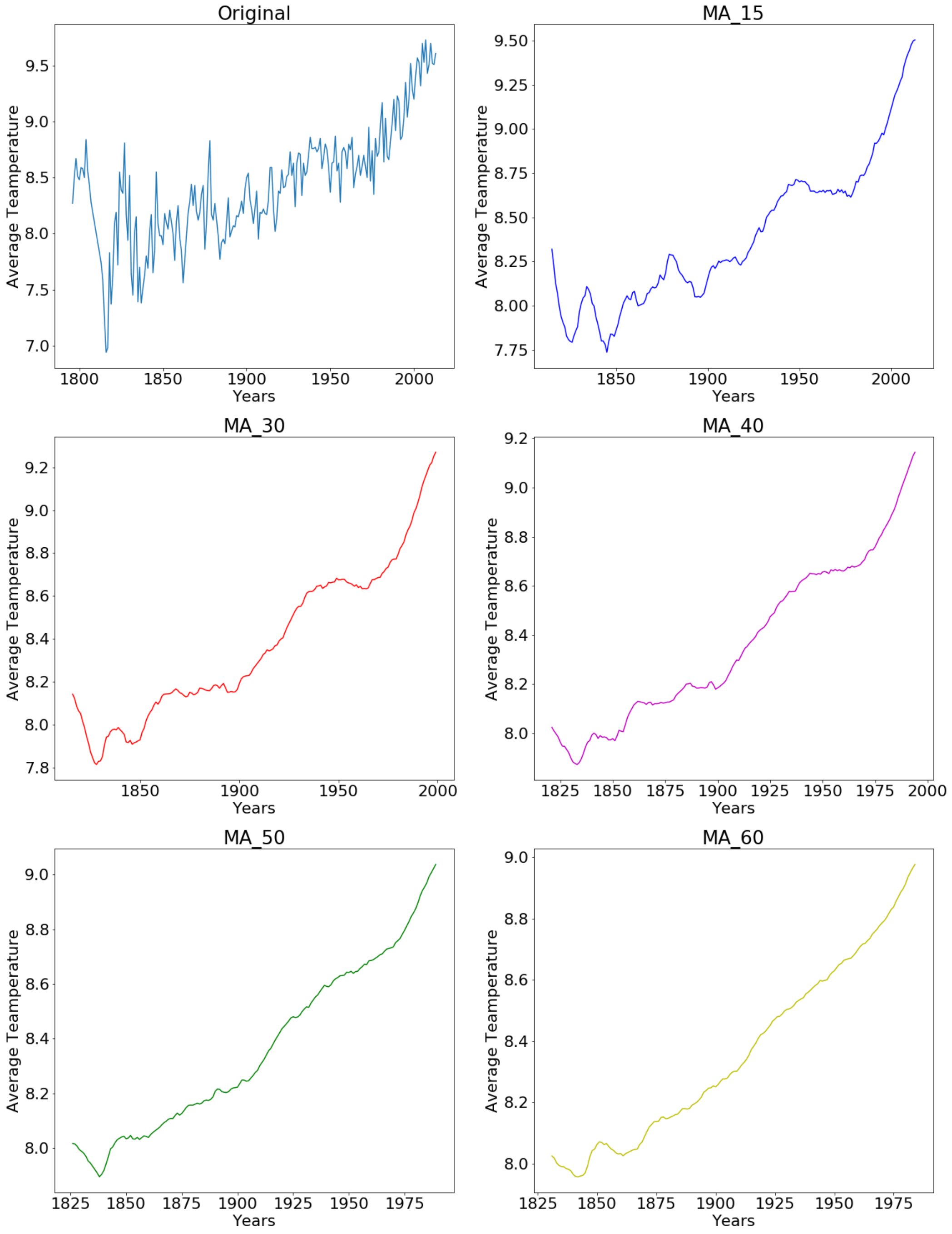
MA_30

MA_40

MA_50

MA_60

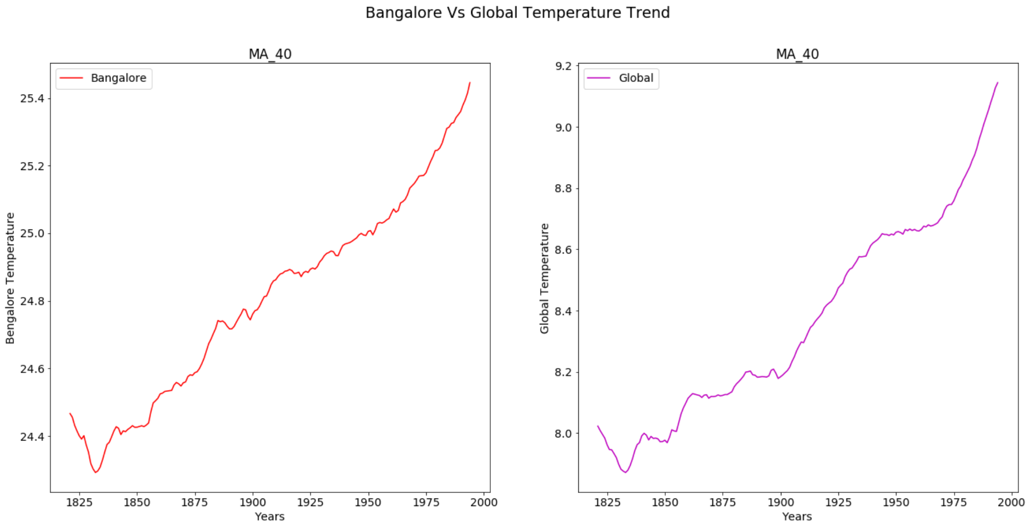
Global Temperature using Different MA Values

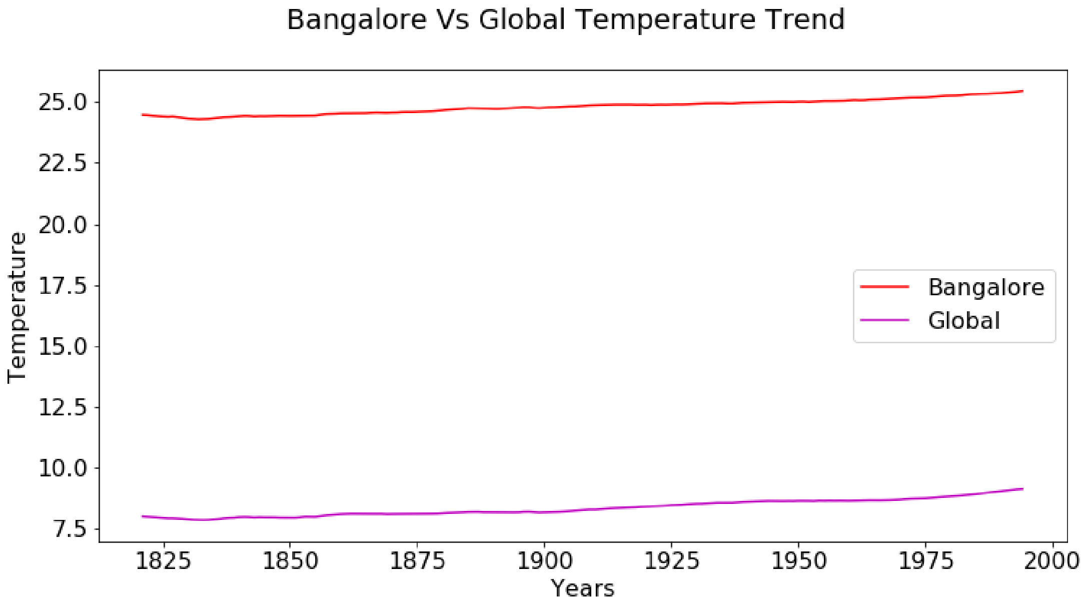


8. **Observations from graphs**:

From the above graphs we can see that the original data without any moving average had a lot of fluctuations. The graphs with moving average window of 15 and 30 still have some amount of fluctuations. Graphs with moving average window of 50 and 60 are quite smooth but tend to miss out on some information. Whereas moving average window of 40 seems to have a balance between both information and fluctuations. Information here means small changes in value. These values can be handy when analyzing time period of small duration. Therefore, I chose to continue with the value 40.

Using the moving average value of 40, I have plotted the following graphs for Bangalore vs Global temperature trends.

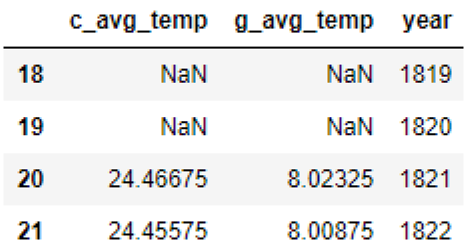




From the above graphs, it is evident that the global temperature plot has higher slope than Bangalore temperature plot. The overall temperature rise is higher for the global temperature than that of Bangalore’s temperature. The above graphs give us an overall idea about the temperature changes. The first graph above has better information but it is a combination of two separate graphs. And in the second graph, the range of temperature varies by great amount to make conclusions. To get a better view of this, the following graph depicts percentage change in temperature. Thereby, the two plots are in the same graph and in the same range.

To find the percentage change, the entire column was divided by the first occurrence of value. Since we have used moving average of 40 and the values are centered. The first value is in 21st row. I used following python code to confirm my assumption.

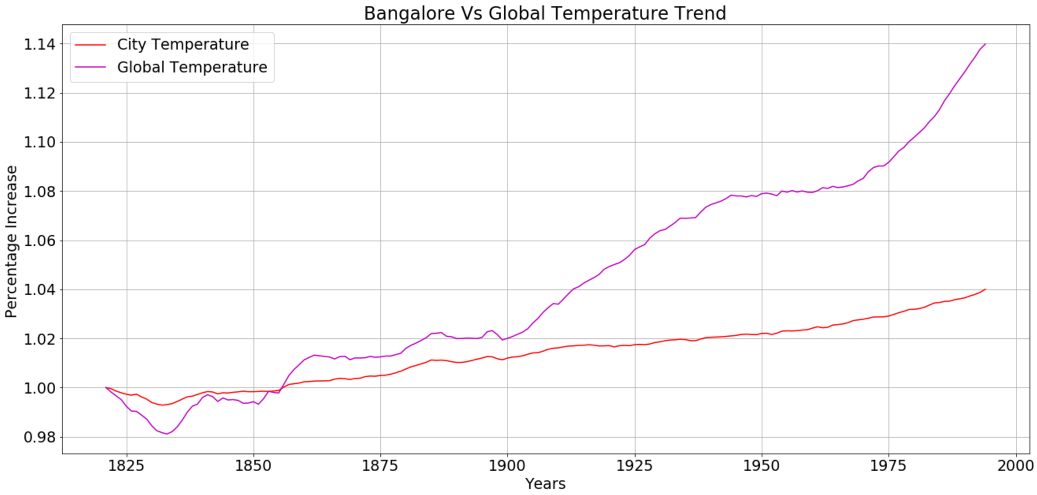
df40.iloc[18:22,]
Output:



	c_avg_temp	g_avg_temp	year
18	NaN	NaN	1819
19	NaN	NaN	1820
20	24.46675	8.02325	1821
21	24.45575	8.00875	1822

9. **Python code for Final Graph :**

```
plt.figure(figsize=(15,6))
c1 = df40['c_avg_temp'][20]
g1 = df40['g_avg_temp'][20]
plt.plot(df40['year'],df40['c_avg_temp']/c1,'r-',label='City Temperature')
plt.plot(df40['year'],df40['g_avg_temp']/g1,'m-',label='Global Temperature')
plt.xlabel('Years')
plt.ylabel('Percentage Increase ')
plt.title('Bangalore Vs Global Temperature Trend')
plt.legend()
plt.grid()
plt.savefig(r"E:\Udacity_Projects\Explore_Weather_Trends\Bangalore_vs_Global_final.png")
plt.show()
```



The graph ranges on Y-axis for the two plots are now similar. There is an overall increase of 4 percent in Bangalore's temperature whereas an overall increase of 14 percent in Global average temperature. The Global temperature kept on decreasing from 1820 to 1834 and has seen a significant rise since then. Temperature decreased by 2 percent in just over a decade time. During this time period, Bangalore's temperature decreased by around 0.65%. Both Bangalore and global temperatures have seen a constant rise after 1855 especially after 1900. When global temperatures have seen some drastic increase compared to Bangalore.

Result- Conclusion:

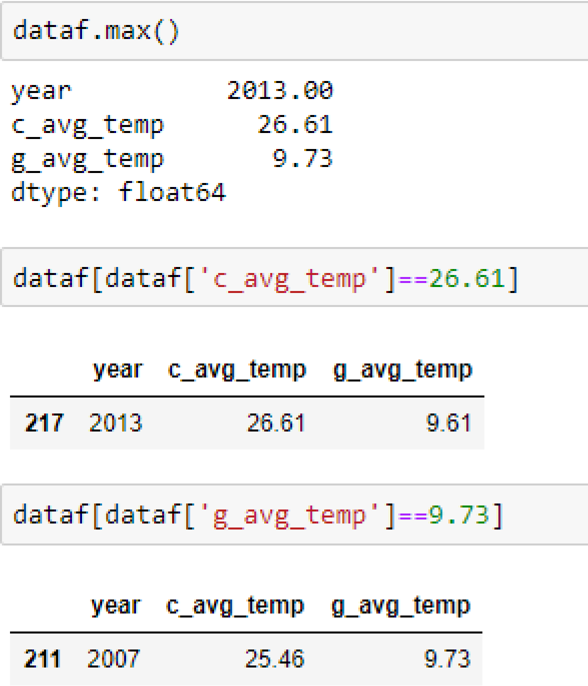
1. Bangalore is hotter compared to global average temperature. This has been consistent over time and the temperature difference is large. On average Bangalore has 3 times the temperature compared to global average.

2. The temperature variance trends have been similar over time. The percentage increase in Bangalore's temperature was less than the global average temperature increase. Temperature decreased uniformly from 1820 to 1834 both for Bangalore and global average. It increased from 1834 to 1841 for both Bangalore and global average. Almost stable from 1841 to 1855 for Bangalore and small fluctuations for global average. Similar trend was observed from 1855 to 1900. Therefore, temperature changes were similar till 1900. If we divide the timeline into time periods of 5 years, We see similar trends. The story changes after this.

3. Temperature rise from 1820 to 1900 for Bangalore was 1.2% whereas for global average was 2%. After 1900, Bangalore had an overall linear increase in temperature and global had an exponential increase. The overall increase in temperature amounts to overall 4% for Bangalore and 14% for Global temperature. The world is getting hotter year by year and so is Bangalore but the degree of increase is large for global temperature.

4. On observing the data for the last 14 years (2000-2013), there is an average change of ±0.15 to ±0.25 per year.

5. The hottest year for Bangalore and global were 2013 and 2007 respectively. And the coolest year for Bangalore and global were 1816 and 1811 respectively. So we had the coolest years around 200 years ago. This was also obvious from the last graph but to get exact years I used functions min(), max() and logical properties of pandas Dataframe .



```
dataf.max()
```

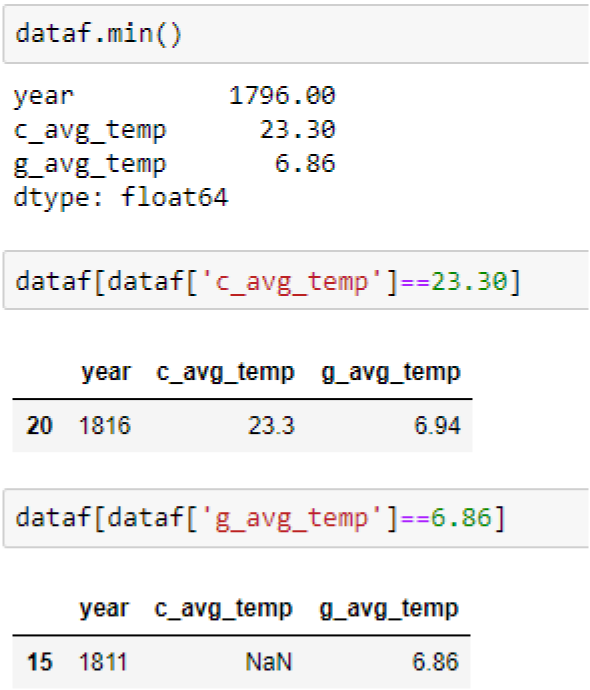
```
year          2013.00
c_avg_temp      26.61
g_avg_temp       9.73
dtype: float64
```

```
dataf[dataf['c_avg_temp']==26.61]
```

	year	c_avg_temp	g_avg_temp
217	2013	26.61	9.61

```
dataf[dataf['g_avg_temp']==9.73]
```

	year	c_avg_temp	g_avg_temp
211	2007	25.46	9.73



```
dataf.min()
```

```
year          1796.00
c_avg_temp      23.30
g_avg_temp       6.86
dtype: float64
```

```
dataf[dataf['c_avg_temp']==23.30]
```

	year	c_avg_temp	g_avg_temp
20	1816	23.3	6.94

```
dataf[dataf['g_avg_temp']==6.86]
```

	year	c_avg_temp	g_avg_temp
15	1811	NaN	6.86

6. To find out the years with temperature less than the overall average temperature, I used the function mean() and logical properties of pandas dataframe.

dataf1.mean()

```
year          1907.127962
c_avg_temp      24.853081
g_avg_temp       8.437630
dtype: float64
```

dataf1[(dataf1['c_avg_temp']<=24.853081)]

This command resulted in 99 rows and the last 7 rows were for years 1934, 1939, 1943, 1950, 1955, 1956 and 1971.

dataf1[(dataf1['g_avg_temp']<8.437630)]

This command resulted in 106 rows with the last 7 rows for years 1923, 1929, 1939, 1950, 1956, 1964, 1976.

It is evident from these results that we have not had a cooler year in recent decades.

Key Considerations:

1. Both tables' global_data and city_data must share common years.
2. Use of moving average concept for smoother plot and better analysis.
3. Null values should be eliminated.
4. Temperatures are plot on Y-axis and years on X-axis.