

- **STEP1:- Extraction of Data from the database**

- To see which cities are available for the country "India"

- **SELECT \* FROM city\_list  
WHERE country = 'India'**

- By joining the database tables(city\_data and global\_data) the required dataset can be achieved,but from the SCHEMA it can be seen that both the tables have a common column i.e **'avg\_temp'**.So we need to change that particular column name

- **ALTER TABLE city\_data  
RENAME COLUMN  
avg\_temp to  
avg temp city;**

- ALTER TABLE city\_data  
RENAME COLUMN  
avg\_temp to  
avg\_temp\_city;
- ALTER TABLE  
global\_data RENAME  
COLUMN avg\_temp to  
avg\_temp\_global;

- Joining the tables(city\_data  
and global\_data) to achieve  
the required dataset

- SELECT global\_data.year,  
avg\_temp\_global,  
avg\_temp\_city FROM  
global\_data  
JOIN city\_data  
ON global\_data.year =  
city\_data.year  
WHERE city ='Bangalore';



+ &lt; &gt; + T

Connect



```
ON global_data.year =  
city_data.year  
WHERE city = 'Bangalore';
```

## STEP2:- Calculating the moving average

```
[ ] from IPython.display import Image  
Image(url= "moving_avg.PNG")
```



## • STEP3:- Python Code to make the Line Chart

- First of all we **import all the required packages** such as numpy,panda,matplotlib
- CSV is Read using the **read\_csv()** method
- The,we use **rolling()** method to calculate the moving average(I have calculated weekly moving avg.).
- Finally,**plot()** method is used to



plot the graph.

```
[ ] #Importing the required packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

df=pd.read_csv('Bangalore.csv') #r
df.tail(10) #having a look at the
```



	year	avg_temp_global	avg_t
--	------	-----------------	-------

208	2004	9.32	
-----	------	------	--

209	2005	9.70	
-----	------	------	--

210	2006	9.53	
-----	------	------	--

211	2007	9.73	
-----	------	------	--

212	2008	9.43	
-----	------	------	--

213	2009	9.51	
-----	------	------	--

214	2010	9.70	
-----	------	------	--

215	2011	9.52	
-----	------	------	--

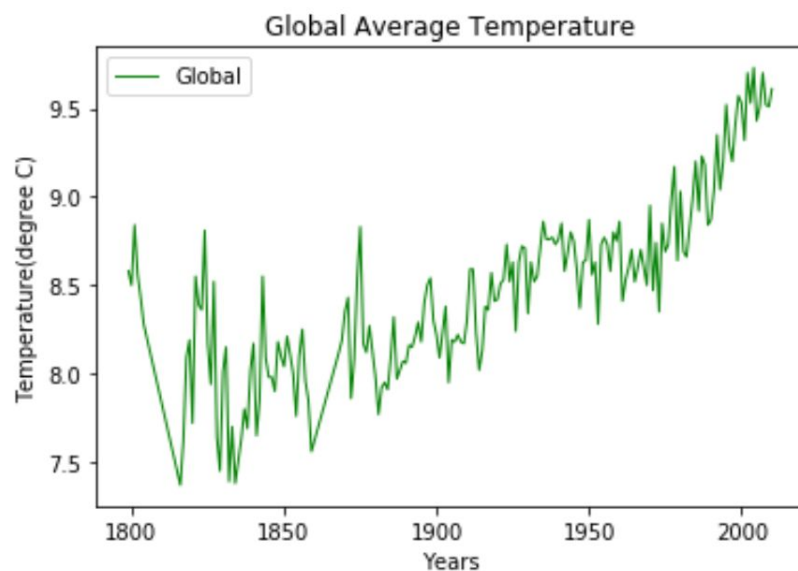
216	2012	9.51	
-----	------	------	--

217	2013	9.61	
-----	------	------	--



```
[ ] #calculating moving average for the
    mA_global_weekly = df.rolling(window=52).mean()
    mA_local_weekly = df.rolling(window=52).mean()
```

```
[ ] #Plotting the global temperature data
    plt.plot(mA_global_weekly['year'], mA_global_weekly['temp'], color='green')
    plt.title("Global Average Temperature")
    plt.xlabel('Years')
    plt.ylabel('Temperature(degree C)')
    plt.legend()
    plt.show()
```



```
[ ] #Plotting the local temperature data
    plt.plot(mA_local_weekly['year'], mA_local_weekly['temp'], color='blue')
    plt.title("Local(Bangalore) Average Temperature")
    plt.xlabel('Years')
```





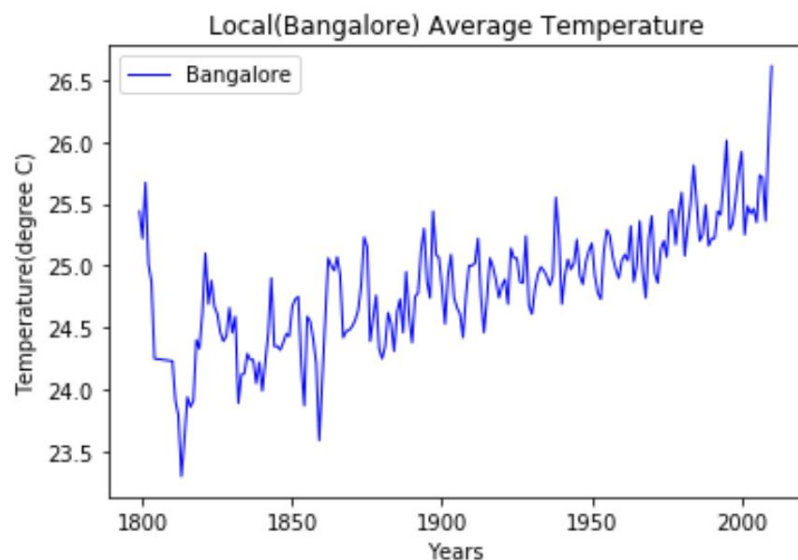
Connect



Years

```
[ ] #Plotting the global temperature da
plt.plot(mA_local_weekly['year'], n

plt.title("Local(Bangalore) Average
plt.xlabel('Years')
plt.ylabel('Temperature(degree C)')
plt.legend()
plt.show()
```



```
[ ] #Plotting the global temperature da
plt.plot(mA_global_weekly['year'],
plt.plot(mA_local_weekly['year'], n

plt.title("(Global vs Bangalore) Av
plt.xlabel('Years')
plt.ylabel('Temperature(degree C)')
plt.legend()
```



1800

1850

1900

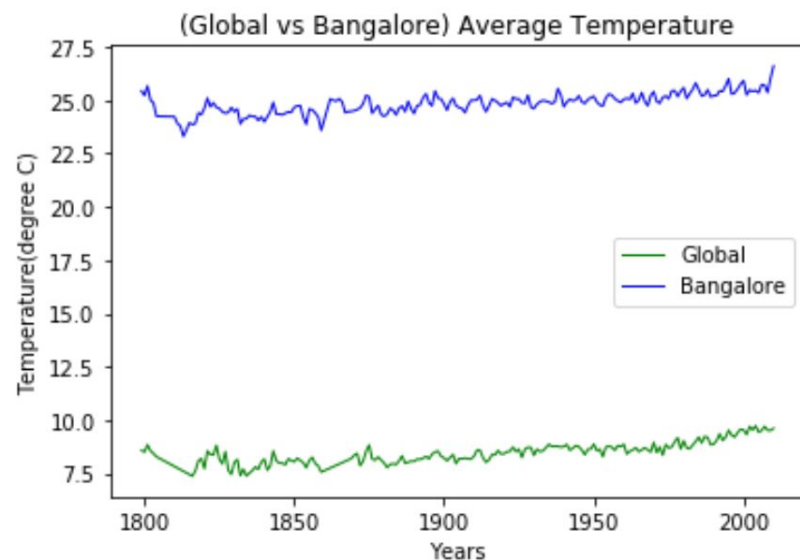
1950

2000

Years

```
[ ] #Plotting the global temperature da
plt.plot(mA_global_weekly['year'],
plt.plot(mA_local_weekly['year'], m

plt.title("(Global vs Bangalore) Av
plt.xlabel('Years')
plt.ylabel('Temperature(degree C)')
plt.legend()
plt.show()
```



```
[ ] avg_global_temp=df['avg_temp_global]
avg_bangalore_temp=df['avg_temp_cit

objects = ('Bangalore', 'Global')
```



1800

1850

1900

1950

2000

Years

```
[ ] avg_global_temp=df['avg_temp_global']
    avg_bangalore_temp=df['avg_temp_cit

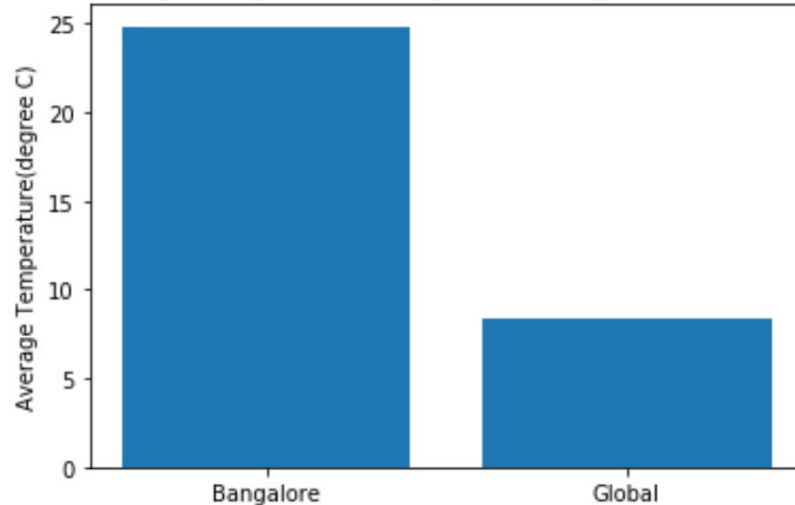
objects = ('Bangalore', 'Global')
y_pos = np.arange(len(objects))
performance = [avg_bangalore_temp, a

plt.bar(y_pos, performance, align='
plt.xticks(y_pos, objects)
plt.ylabel('Average Temperature(deg
plt.title('Average Temperature Comp

plt.show()
```



Average Temperature Comparision(Bangalore vs Global)



```
[ ] df2=df.apply(lambda x: x.fillna(x.n
    bangalore temp=df2['avg temp citv']
```