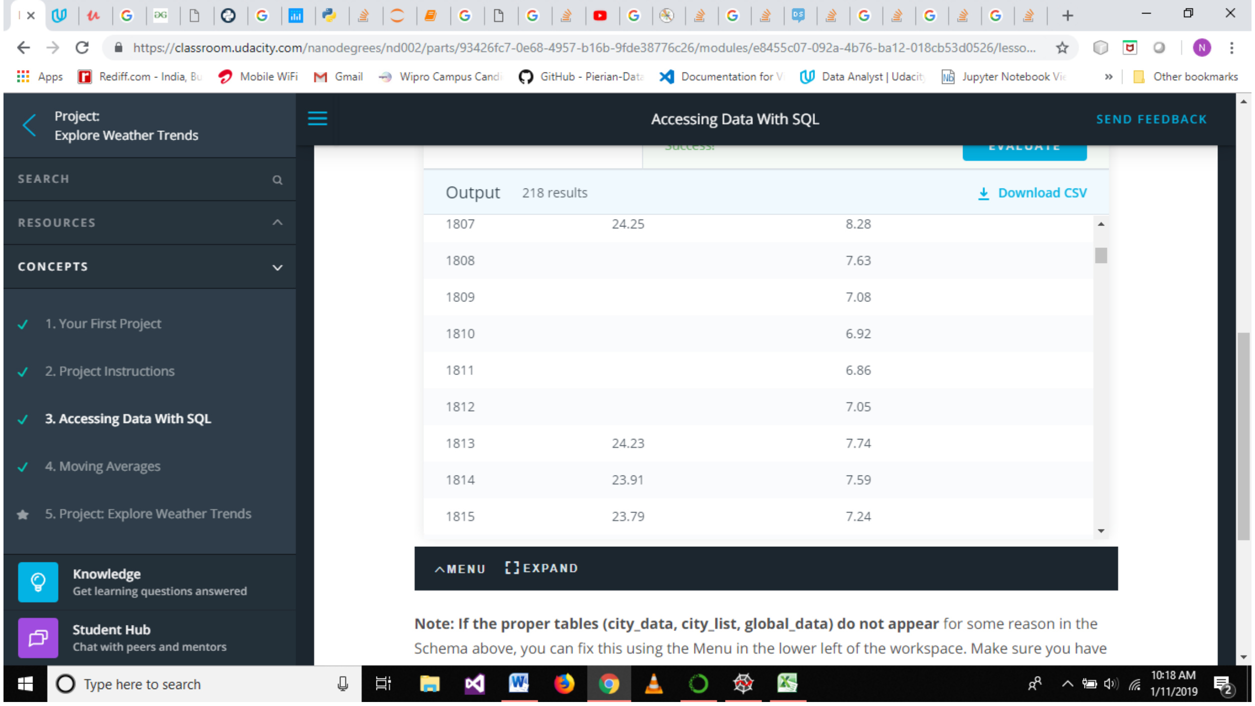
Exploring Weather Trends

For this project I have chosen the city of Bangalore, India.
The SQL Query used to extract the average global temperature and temperature from the local city of Bangalore,India.

select city_data.year,city_data.avg_temp as city_temp,global_data.avg_temp as global_temp from city_data inner join global_data on city_data.year=global_data.year where city_data.city='Bangalore';

Below is a screenshot of the result of this query. Some of the data for the city_temp column is missing.



Output 218 results Download CSV

1807	24.25	8.28
1808		7.63
1809		7.08
1810		6.92
1811		6.86
1812		7.05
1813	24.23	7.74
1814	23.91	7.59
1815	23.79	7.24

Note: If the proper tables (city_data, city_list, global_data) do not appear for some reason in the Schema above, you can fix this using the Menu in the lower left of the workspace. Make sure you have

Step 1: I downloaded the result as csv file.

Step 2: I opened the csv file in Spyder(Python3.7) using the Anaconda software. I opened the csv file into a DataFrame. Pandas library has a two dimensional data structure that can be visualized in a a tabular format, this is called DataFrame. My reason for choosing the DataFrame is that they are easy to work with and also have a lot functions that can be readily used. They also auto populate missing data with nann(null values) . I will attach the whole code at the end of the document.

Command used: data=pd.read_csv('results.csv')

Step 3: Since all the columns are populated as string in the DataFrame, I have to change the year column to Date format. This is done in order to make the column Year as X-axis. If the column is not converted to Date format the X-axis populates default values.

Command used: data['year'] = pd.to_datetime(data['year'],format='%Y')

Step 4: Assign the column year as the x-axis.

Command used: data.set_index(data['year'],inplace=True)

Step 5: I now calculated the 7 day moving average of both columns city_temp and global_temp. For this I have used the function rolling(). Rolling() gives the rolling window calculations and this can be used with sum() or mean().

I could have also calculated the moving average in the Excel sheet and then populated the data in the DataFrame but this would be time consuming and prone to errors. I have calculated the 7-Day Moving Average.

Command used:

data['city_temp_MA7']=data.city_temp.rolling(7).mean()

data['global_temp_MA7']=data.global_temp.rolling(7).mean()

Step 6: I plotted the columns global_temp, city_temp_MA7 and global_temp_MA7. To Make the graph easier to read and understand I have used different colors for each column and labeled them as well.

Command used:

data['city_temp_MA7'].plot(linewidth=1, c='blue',label='Bangalore city temperature in Celcius')

data['global_temp'].plot(linewidth=1, c='red',label='Average global Temperature in Celcius')

data['global_temp_MA7'].plot(linewidth=1, c='green',label=' Moving Average global Temperature in Celcius')

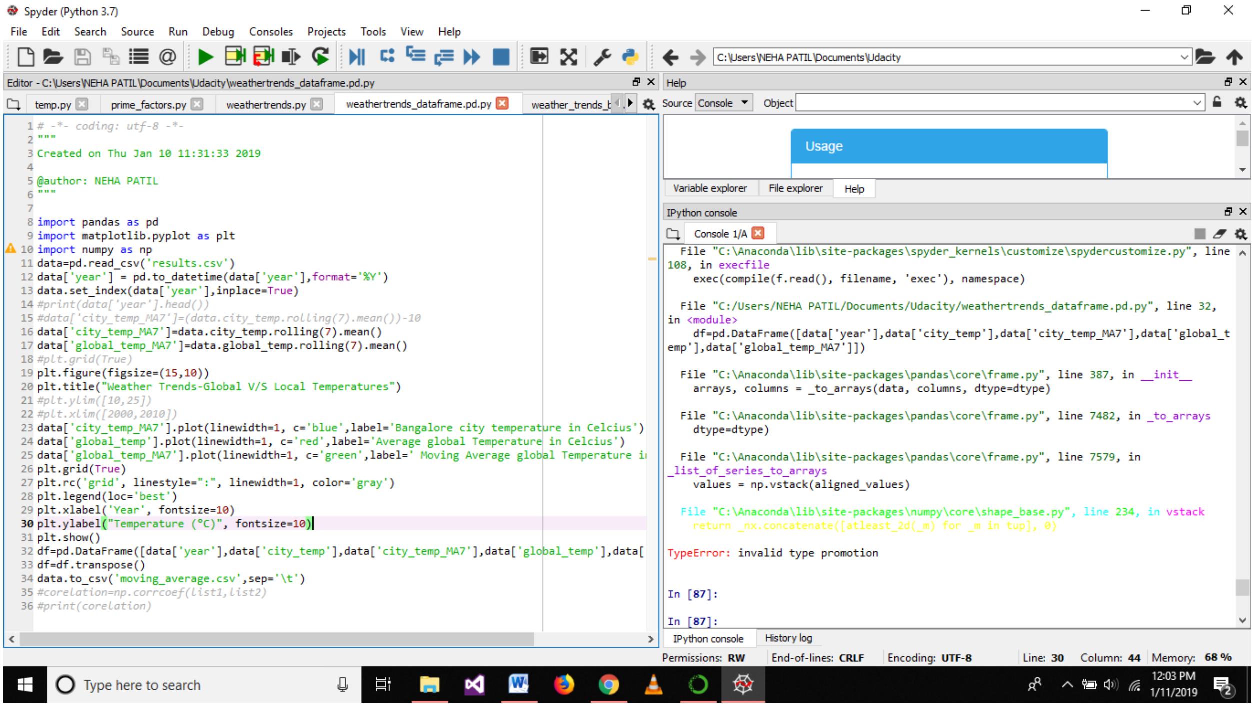
Step 7: Beautify the graph. To make the graph more presentable I have added a title to graph as well as labels to axes.

Command Used:

```
plt.figure(figsize=(15,10))

plt.title("Weather Trends-Global V/S Local Temperatures")

plt.grid(True)

plt.rc('grid', linestyle=":", linewidth=1, color='gray')

plt.legend(loc='best')

plt.xlabel('Year', fontsize=10)

plt.ylabel("Temperature (°C)", fontsize=10)

plt.show()
```

Below are the screenshots of the code and the graph.

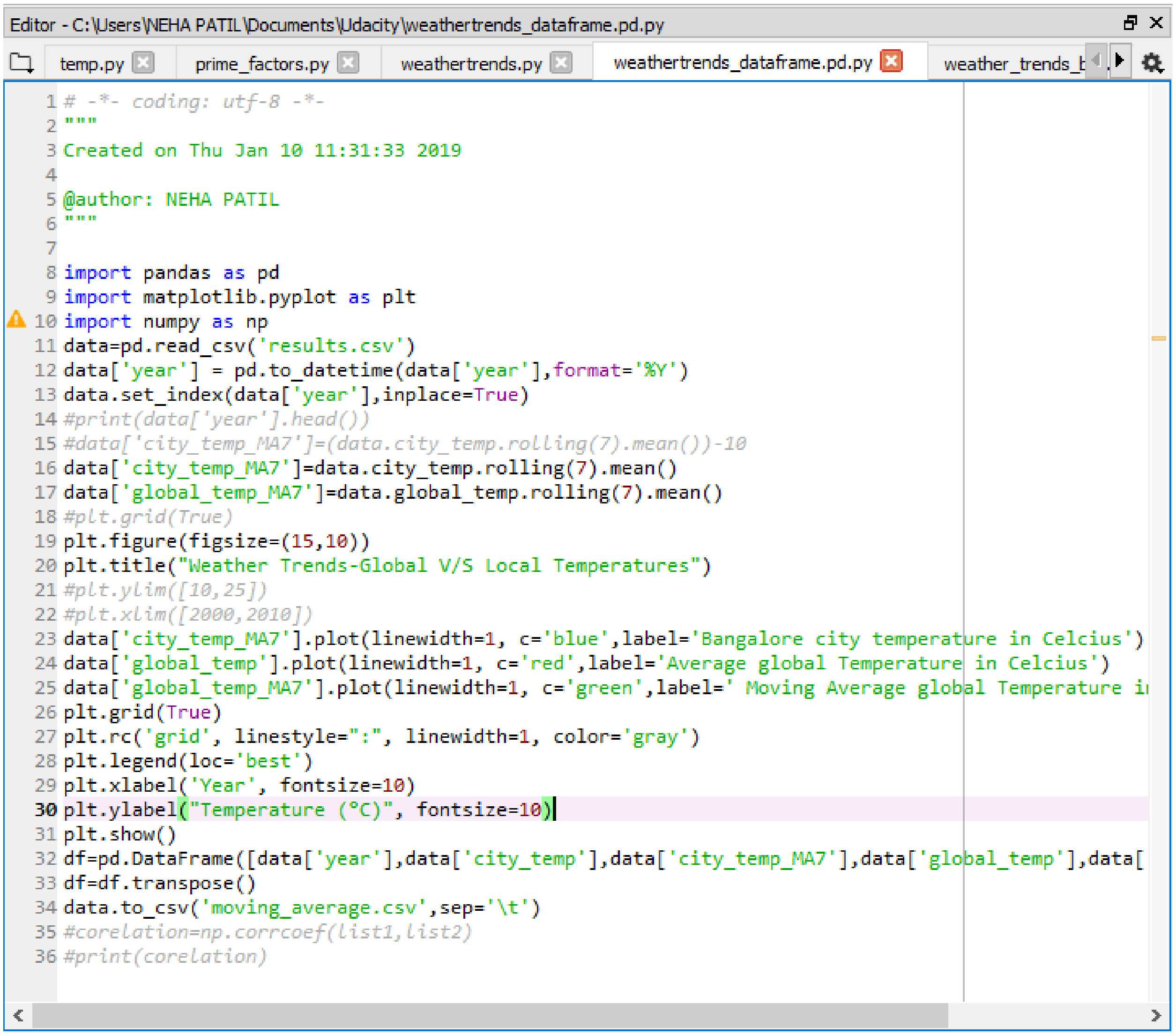
Fig1: Snapshot of the Python code. IDE being used is Spyder.



```
# -*- coding: utf-8 -*-
"""
Created on Thu Jan 10 11:31:33 2019

@author: NEHA PATIL
"""

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
data=pd.read_csv('results.csv')
data['year'] = pd.to_datetime(data['year'],format='%Y')
data.set_index(data['year'],inplace=True)
#print(data['year'].head())
#data['city_temp_MA7']=(data.city_temp.rolling(7).mean())-10
data['city_temp_MA7']=data.city_temp.rolling(7).mean()
data['global_temp_MA7']=data.global_temp.rolling(7).mean()
#plt.grid(True)
plt.figure(figsize=(15,10))
plt.title("Weather Trends-Global V/S Local Temperatures")
#plt.ylim([10,25])
#plt.xlim([2000,2010])
data['city_temp_MA7'].plot(linewidth=1, c='blue',label='Bangalore city temperature in Celcius')
data['global_temp'].plot(linewidth=1, c='red',label='Average global Temperature in Celcius')
data['global_temp_MA7'].plot(linewidth=1, c='green',label=' Moving Average global Temperature i
plt.grid(True)
plt.rc('grid', linestyle=":", linewidth=1, color='gray')
plt.legend(loc='best')
plt.xlabel('Year', fontsize=10)
plt.ylabel("Temperature (°C)", fontsize=10)
plt.show()
df=pd.DataFrame([data['year'],data['city_temp'],data['city_temp_MA7'],data['global_temp'],data[
df=df.transpose()
data.to_csv('moving_average.csv',sep='\t')
#corelation=np.corrcoef(list1,list2)
#print(corelation)
```

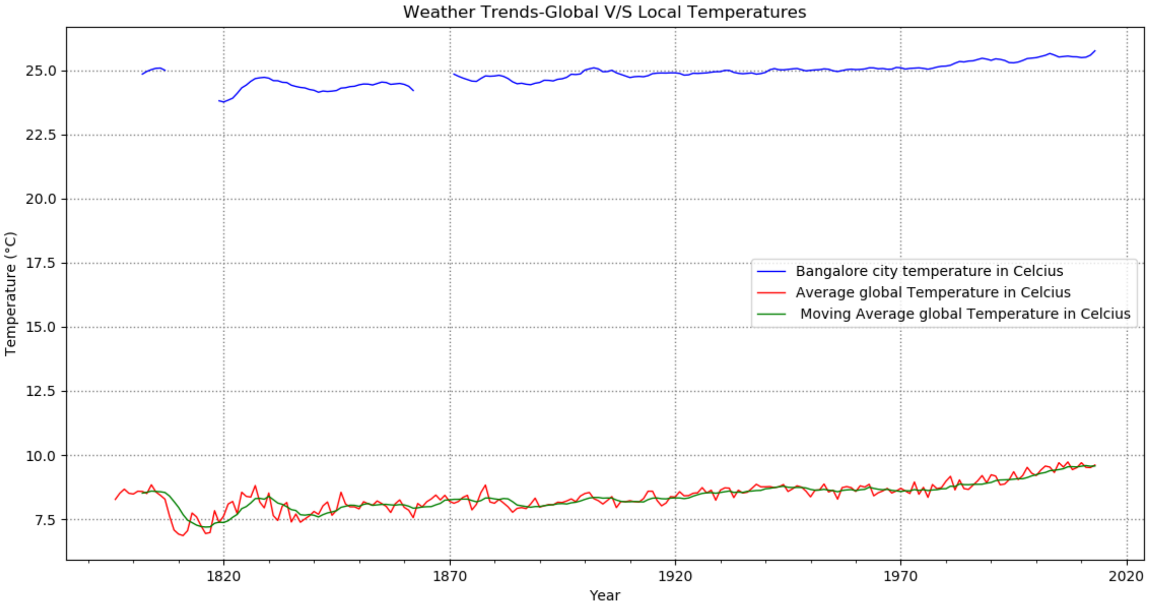
```
  File "C:\Anaconda\lib\site-packages\spyder_kernels\customize\spydercustomize.py", line
108, in execfile
    exec(compile(f.read(), filename, 'exec'), namespace)

  File "C:/Users/NEHA PATIL/Documents/Udacity/weathertrends_dataframe.pd.py", line 32,
in <module>
    df=pd.DataFrame([data['year'],data['city_temp'],data['city_temp_MA7'],data['global_t
emp'],data['global_temp_MA7']])

  File "C:\Anaconda\lib\site-packages\pandas\core\frame.py", line 387, in __init__
    arrays, columns = _to_arrays(data, columns, dtype=dtype)

  File "C:\Anaconda\lib\site-packages\pandas\core\frame.py", line 7482, in _to_arrays
    dtype=dtype)

  File "C:\Anaconda\lib\site-packages\pandas\core\frame.py", line 7579, in
_list_of_series_to_arrays
    values = np.vstack(aligned_values)

  File "C:\Anaconda\lib\site-packages\numpy\core\shape_base.py", line 234, in vstack
    return _nx.concatenate([atleast_2d(_m) for _m in tup], 0)

TypeError: invalid type promotion


In [87]:

In [87]:
```

Fig2: Snapshot of the code.


Editor - C:\Users\NEHA PATIL\Documents\Udacity\weathertrends_dataframe.pd.py
temp.py
prime_factors.py
weathertrends.py
weathertrends_dataframe.pd.py
weather_trends_b

```python
# -*- coding: utf-8 -*-
"""
Created on Thu Jan 10 11:31:33 2019

@author: NEHA PATIL
"""

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
data=pd.read_csv('results.csv')
data['year'] = pd.to_datetime(data['year'],format='%Y')
data.set_index(data['year'],inplace=True)
#print(data['year'].head())
#data['city_temp_MA7']=(data.city_temp.rolling(7).mean())-10
data['city_temp_MA7']=data.city_temp.rolling(7).mean()
data['global_temp_MA7']=data.global_temp.rolling(7).mean()
#plt.grid(True)
plt.figure(figsize=(15,10))
plt.title("Weather Trends-Global V/S Local Temperatures")
#plt.ylim([10,25])
#plt.xlim([2000,2010])
data['city_temp_MA7'].plot(linewidth=1, c='blue',label='Bangalore city temperature in Celcius')
data['global_temp'].plot(linewidth=1, c='red',label='Average global Temperature in Celcius')
data['global_temp_MA7'].plot(linewidth=1, c='green',label=' Moving Average global Temperature i
plt.grid(True)
plt.rc('grid', linestyle=":", linewidth=1, color='gray')
plt.legend(loc='best')
plt.xlabel('Year', fontsize=10)
plt.ylabel("Temperature (°C)", fontsize=10)
plt.show()
df=pd.DataFrame([data['year'],data['city_temp'],data['city_temp_MA7'],data['global_temp'],data[
df=df.transpose()
data.to_csv('moving_average.csv',sep='\t')
#corelation=np.corrcoef(list1,list2)
#print(corelation)
```

Fig3: Snapshot of the graphs.



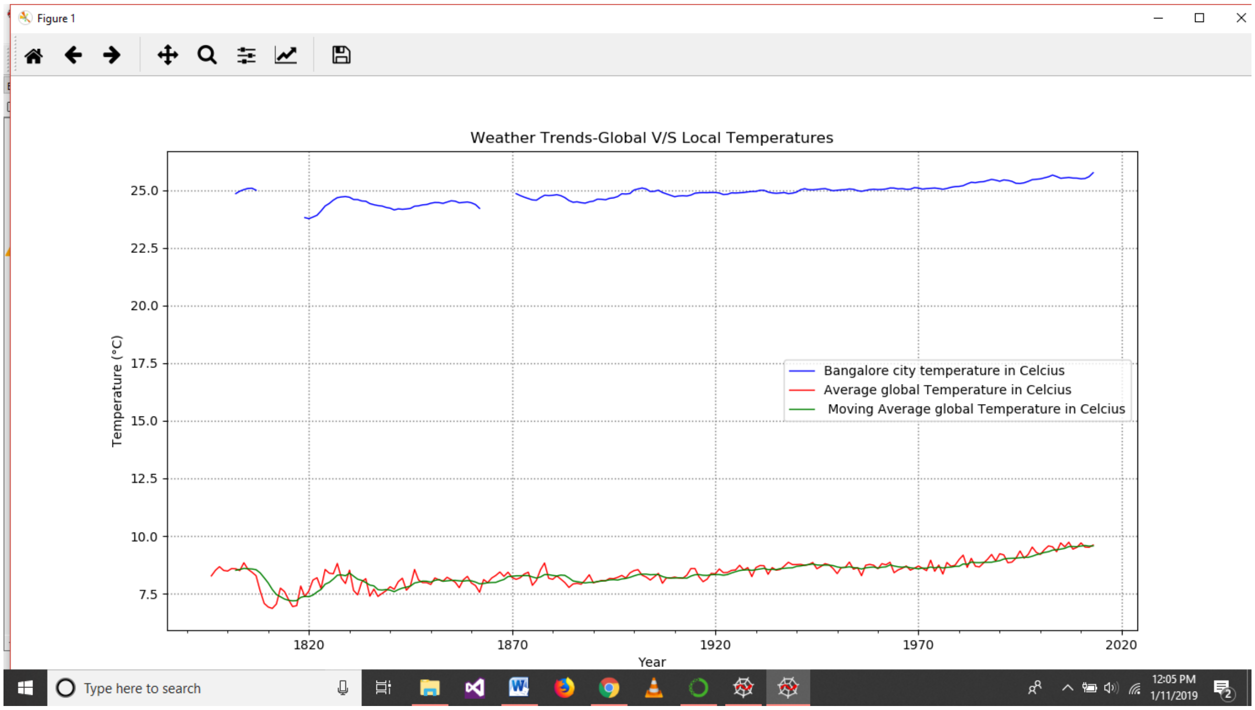


Figure 1
Weather Trends-Global V/S Local Temperatures
Temperature (°C)
25.0
22.5
20.0
17.5
15.0
12.5
10.0
7.5
1820
1870
1920
1970
2020
Year
Bangalore city temperature in Celcius
Average global Temperature in Celcius
Moving Average global Temperature in Celcius
Type here to search
12:05 PM
1/11/2019

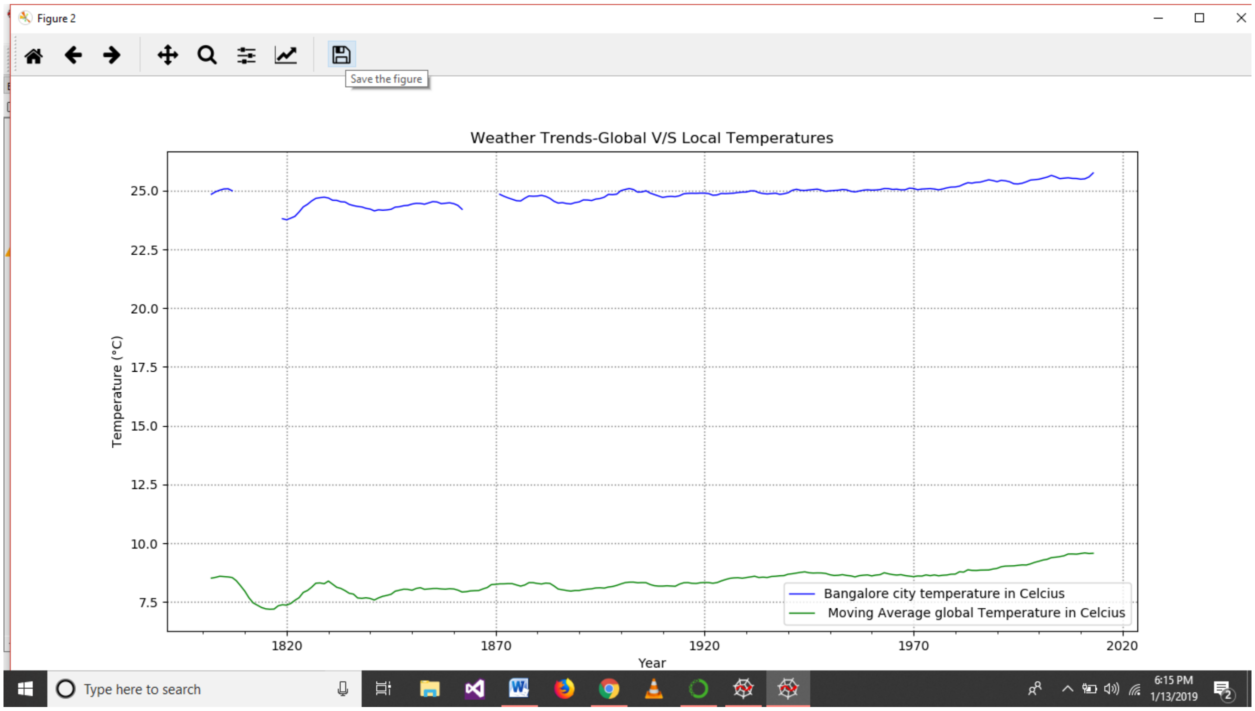
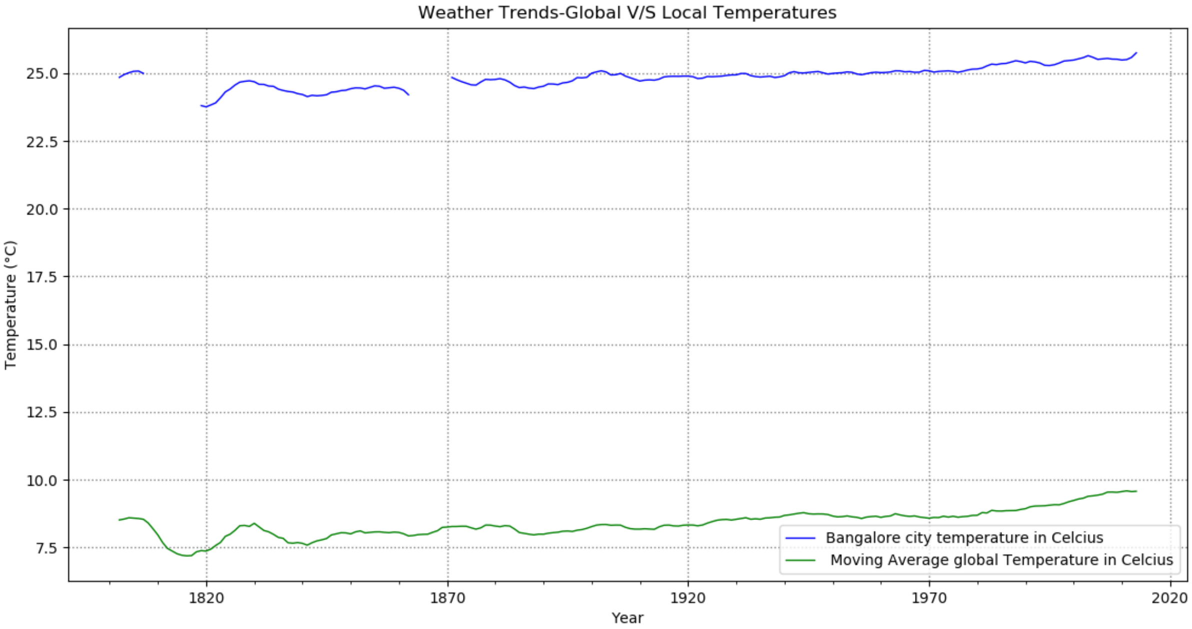
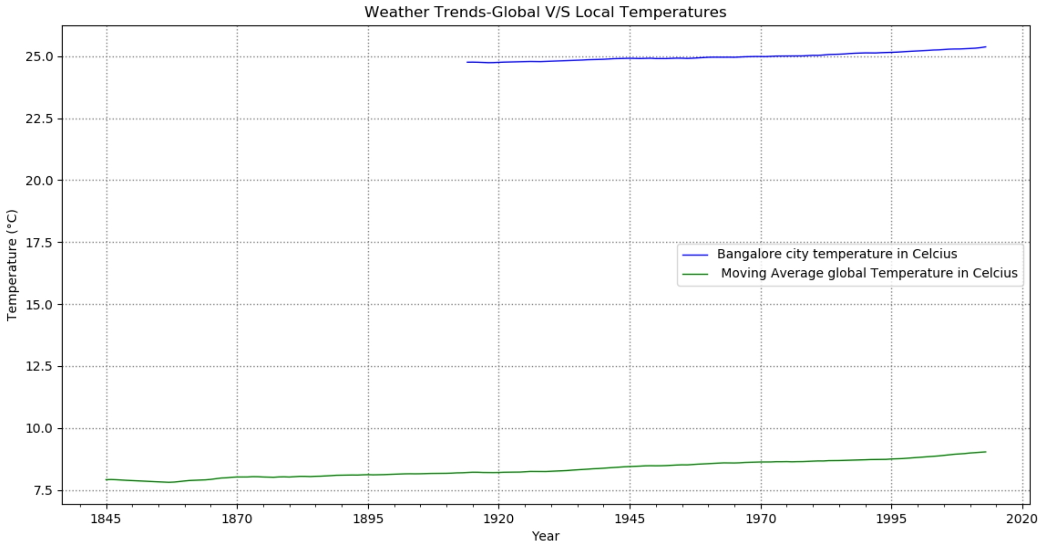


Figure 2
Save the figure
Weather Trends-Global V/S Local Temperatures
Temperature (°C)
25.0
22.5
20.0
17.5
15.0
12.5
10.0
7.5
1820
1870
1920
1970
2020
Year
Bangalore city temperature in Celcius
Moving Average global Temperature in Celcius
Type here to search
6:15 PM
1/13/2019


Weather Trends-Global V/S Local Temperatures
Temperature (°C)
25.0
22.5
20.0
17.5
15.0
12.5
10.0
7.5
1820
1870
1920
1970
2020
Year
Bangalore city temperature in Celcius
Moving Average global Temperature in Celcius

Observations Made:

1. The city of Bangalore is hotter when compared to the global average temperature.
2. The difference between the city average temperature of Bangalore and the global average temperature has been pretty consistent over the years.
3. The city temperature average temperature of Bangalore and the global average temperature are positively co-related. This means if the global average temperature increase, the city temperature also increases with it and if the global average temperature decreases then the city temperature also decreases.
4. According to the trend the seen in the plot, the global average temperature is increasing steadily over the years. When the window of the Moving Average is increased to 50, we can see a smooth line that is steadily increases over time.



5. The above observations imply that the world is getting hotter as time goes by. The average temperature of Bangalore city is also increasing as the years go by.
6. This trend of the average global and city temperature increasing as the years go by has been consistent over the past century.

ACKNOWLEDGEMENT:

I have taken some assistance/support from the following sources:

1. https://www.pythonforbeginners.com/csv/using-the-csv-module-in-python
2. https://docs.python.org/3.3/
3. https://stackoverflow.com/

4. https://www.geeksforgeeks.org/program-find-correlation-coefficient/
5. https://pandas.pydata.org/pandas-docs/version/0.23.4/generated/pandas.DataFrame.html
6. https://matplotlib.org/users/pyplot_tutorial.html