Step 1: Import all required libraries to do data analysis

importing all below required libraries to do data analysis

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

Step 2: Loading data tables

Import all 3 csv files one by one 1st: Now importing city_data.csv file 2nd: Now importing city_list.csv file 3rd: Now importing global_data.csv file

In [2]:

```
df_city_data = pd.read_csv('city_data.csv' )

df_global_data =pd.read_csv('global_data.csv')

df_city_list = pd.read_csv('city_list.csv' )
```

Exploring tables

In [3]:

```
#Checking number of rows and columns
print(df_city_data)

# [70792 rows x 4 columns]
```

```
       year     city        country  avg_temp
0      1849  Abidjan  Côte D'Ivoire     25.58
1      1850  Abidjan  Côte D'Ivoire     25.52
2      1851  Abidjan  Côte D'Ivoire     25.67
3      1852  Abidjan  Côte D'Ivoire       NaN
4      1853  Abidjan  Côte D'Ivoire       NaN
...     ...      ...            ...       ...
70787  2009  Zapopan         Mexico     21.76
70788  2010  Zapopan         Mexico     20.90
70789  2011  Zapopan         Mexico     21.55
70790  2012  Zapopan         Mexico     21.52
70791  2013  Zapopan         Mexico     22.19

[70792 rows x 4 columns]
```

In [4]:

```
#Checking number of rows and columns
df_city_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70792 entries, 0 to 70791
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   year      70792 non-null  int64  
 1   city      70792 non-null  object 
 2   country   70792 non-null  object 
 3   avg_temp  68245 non-null  float64
dtypes: float64(1), int64(1), object(2)
memory usage: 2.2+ MB
```

In [5]:

```
#caluclate missing values in the "avg_temp" column
missing_data = df_city_data["avg_temp"].isna()
sum(missing_data)
```

Out[5]:

```
2547
```

In [6]:

```
#calculate the number of affected unique city names
len(df_city_data[missing_data]["city"].unique())
```

Out[6]:

```
235
```

In [7]:

```
#checking the info of the global_data dataframe
df_global_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 266 entries, 0 to 265
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   year      266 non-null    int64  
 1   avg_temp  266 non-null    float64
dtypes: float64(1), int64(1)
memory usage: 4.3 KB
```

In [8]:

```
#checking the head of the global_data dataframe
df_global_data.head()
```

Out[8]:

	year	avg_temp
0	1750	8.72
1	1751	7.98
2	1752	5.78
3	1753	8.39
4	1754	8.47

In [9]:

```
#calculate the mininmal values of the "year" and "avg_temp" column in the global_data dataframe
print(df_global_data["year"].min())
print(df_global_data["avg_temp"].min())
```

```
1750
5.78
```

In [10]:

```
#calculate the maximal values of the "year" and "avg_temp" column in the global_data dataframe
print(df_global_data["year"].max())
print(df_global_data["avg_temp"].max())
```

```
2015
9.83
```

In [11]:

```
#Checking number of rows and columns
df_city_list.info()

#[342 rows x 2 columns]
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 342 entries, 0 to 341
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   city     342 non-null    object
 1   country  342 non-null    object
dtypes: object(2)
memory usage: 5.5+ KB
```

Check asian countries

In [12]:

```
#city_list[city_list.country.isin(['India'])]

Japan = df_city_list[(df_city_list.country=="Japan") & (df_city_list.city== "Tokyo")]
print(Japan)
```

```
      city country
311  Tokyo   Japan
```

In [13]:

```
#Uncomment below if require

# choice=input('about what counry??')
# df_city_list[df_city_list.country.isin([choice])]
```

Now i am checking my city which is New Delhi

In [14]:

```
New_delhi_avg_temp = df_city_data[(df_city_data.country=="India") & (df_city_data.city=
= "New Delhi")]
print(New_delhi_avg_temp)
```

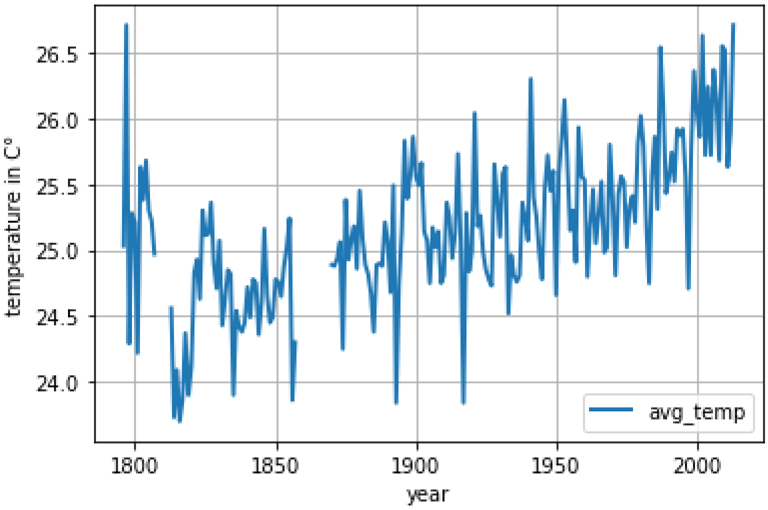
```
       year       city country  avg_temp
45694  1796  New Delhi   India     25.03
45695  1797  New Delhi   India     26.71
45696  1798  New Delhi   India     24.29
45697  1799  New Delhi   India     25.28
45698  1800  New Delhi   India     25.21
...     ...        ...     ...       ...
45907  2009  New Delhi   India     26.55
45908  2010  New Delhi   India     26.52
45909  2011  New Delhi   India     25.63
45910  2012  New Delhi   India     25.89
45911  2013  New Delhi   India     26.71

[218 rows x 4 columns]
```

In [15]:

```
New_delhi_plot=New_delhi_avg_temp.plot('year','avg_temp',grid=True,LineWidth=2)

plt.ylabel('temperature in C°')
plt.xlabel('year')
```

Out[15]:

```
Text(0.5, 0, 'year')
```



I have plotted the avg tem of new delhi and as you can see in plot trend or line is missing its due to missing data.

Now i am checking one more india city which is Banaglore

In [16]:

```
Bangalore_avg_temp = df_city_data[(df_city_data.country=="India") & (df_city_data.city=
= "Bangalore")]
print(Bangalore_avg_temp)
```

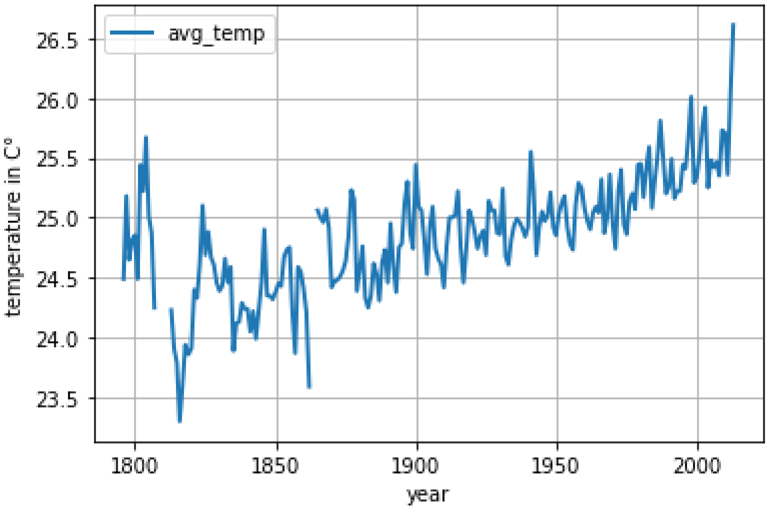
```
      year       city country  avg_temp
6367  1796  Bangalore   India     24.49
6368  1797  Bangalore   India     25.18
6369  1798  Bangalore   India     24.65
6370  1799  Bangalore   India     24.81
6371  1800  Bangalore   India     24.85
...    ...        ...     ...       ...
6580  2009  Bangalore   India     25.73
6581  2010  Bangalore   India     25.71
6582  2011  Bangalore   India     25.36
6583  2012  Bangalore   India     26.04
6584  2013  Bangalore   India     26.61

[218 rows x 4 columns]
```

In [17]:

```
Bangalore_plot=Bangalore_avg_temp.plot('year','avg_temp',grid=True,LineWidth=2)

plt.ylabel('temperature in C°')
plt.xlabel('year')
```

Out[17]:

```
Text(0.5, 0, 'year')
```



This is the plot for Japan,Tokyo

In [18]:

```
Tokyo_avg_temp = df_city_data[(df_city_data.country=="Japan") & (df_city_data.city== "Tokyo")]
print(Tokyo_avg_temp)
```

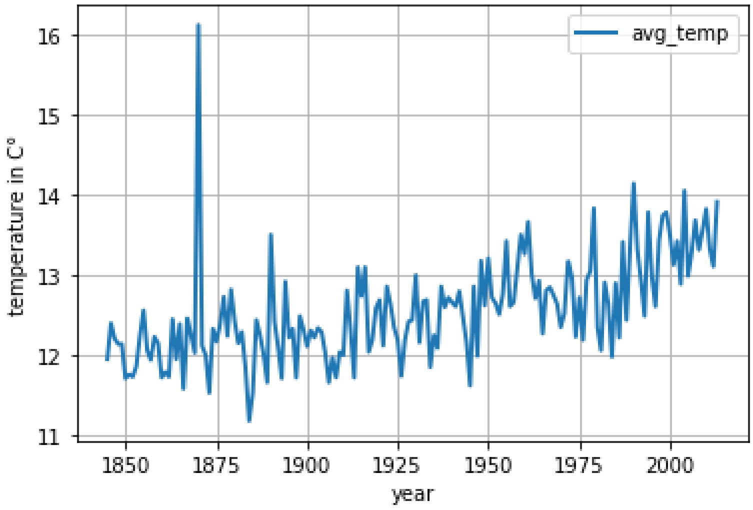
```
       year   city country  avg_temp
63890  1845  Tokyo   Japan     11.95
63891  1846  Tokyo   Japan     12.40
63892  1847  Tokyo   Japan     12.21
63893  1848  Tokyo   Japan     12.14
63894  1849  Tokyo   Japan     12.14
...     ...    ...     ...       ...
64054  2009  Tokyo   Japan     13.55
64055  2010  Tokyo   Japan     13.82
64056  2011  Tokyo   Japan     13.32
64057  2012  Tokyo   Japan     13.11
64058  2013  Tokyo   Japan     13.91

[169 rows x 4 columns]
```

In [19]:

```
Tokyo_plot = Tokyo_avg_temp.plot('year','avg_temp',grid=True,LineWidth=2)
plt.ylabel('temperature in C°')
plt.xlabel('year')
```

Out[19]:

```
Text(0.5, 0, 'year')
```



This is the plot for China, Shanghai

In [20]:

```
Shanghai_avg_temp = df_city_data[(df_city_data.country=="China") & (df_city_data.city==
"Shanghai")]
print(Shanghai_avg_temp)
```

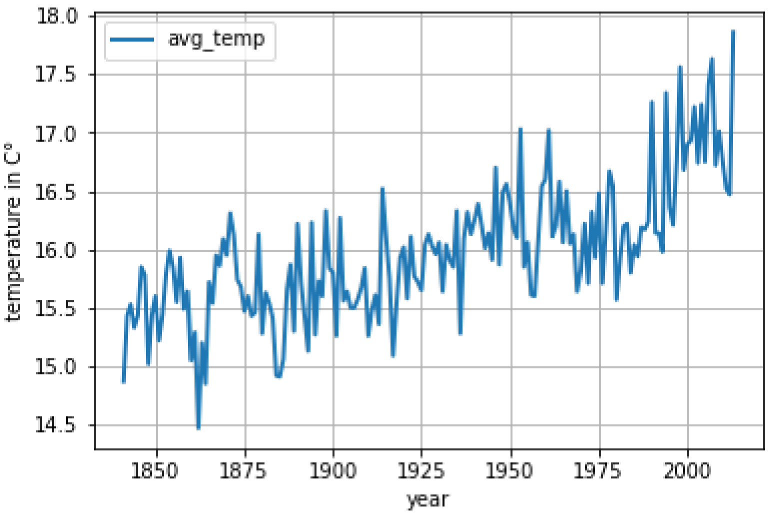
```
       year      city country  avg_temp
59174  1841  Shanghai   China     14.87
59175  1842  Shanghai   China     15.43
59176  1843  Shanghai   China     15.53
59177  1844  Shanghai   China     15.33
59178  1845  Shanghai   China     15.43
...     ...       ...     ...       ...
59342  2009  Shanghai   China     17.01
59343  2010  Shanghai   China     16.75
59344  2011  Shanghai   China     16.52
59345  2012  Shanghai   China     16.47
59346  2013  Shanghai   China     17.86

[173 rows x 4 columns]
```

In [21]:

```
Shanghai_plot = Shanghai_avg_temp.plot('year','avg_temp',grid=True,LineWidth=2)
plt.ylabel('temperature in C°')
plt.xlabel('year')
```

Out[21]:

```
Text(0.5, 0, 'year')
```



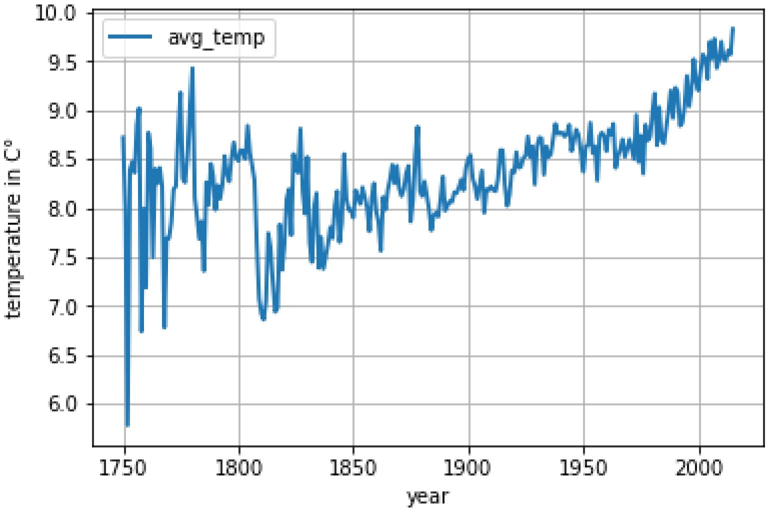
This is the average Temerature of entire planet which means it inculdes all countries and its city

In [22]:

```
Earth_Avg_temp_plot = df_global_data.plot('year','avg_temp',grid=True,LineWidth=2)

plt.ylabel('temperature in C°')
plt.xlabel('year')
```

Out[22]:

```
Text(0.5, 0, 'year')
```



below is the comparisions of majors cities with overall earth avg temp

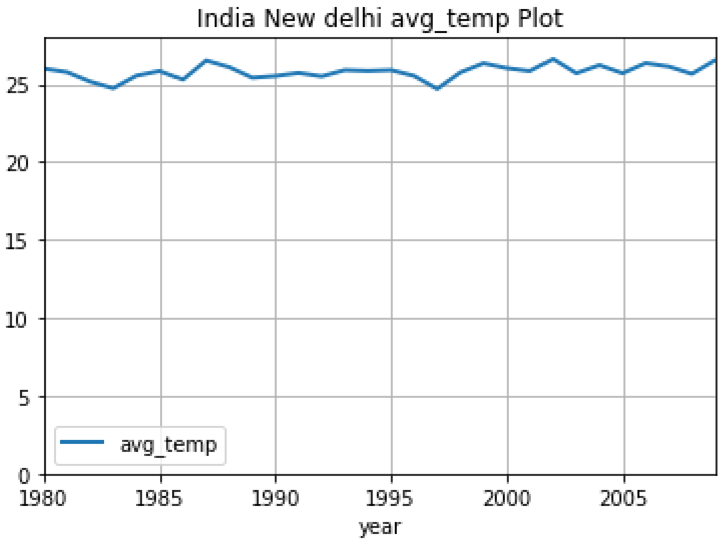
In [23]:

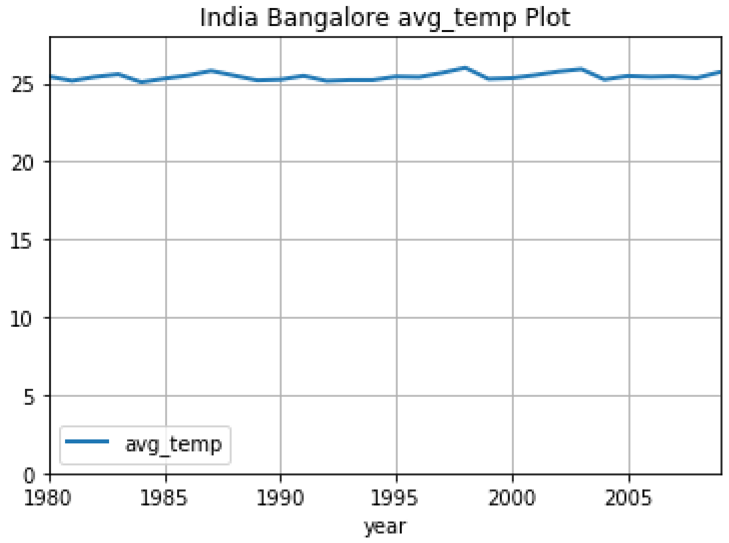
```
print('1)India,New Dehli:')
first=New_delhi_avg_temp.plot('year','avg_temp',grid=True,LineWidth=2)
plt.axis([1980, 2009,0,28])
plt.title("India New delhi avg_temp Plot")

print('2)India,Bangalore:')
first=Bangalore_avg_temp.plot('year','avg_temp',grid=True,LineWidth=2)
plt.axis([1980, 2009,0,28])
plt.title("India Bangalore avg_temp Plot")

print('3)China,Shanghai:')
Scnd = Shanghai_avg_temp.plot('year','avg_temp',grid=True,LineWidth=2)
plt.axis([1980, 2009,0,28])
plt.title("China Shanghai avg_temp Plot")

print('4)Japan,Tokyo:')
Scnd = Tokyo_avg_temp.plot('year','avg_temp',grid=True,LineWidth=2)
plt.axis([1980, 2009,0,28])
plt.title("Japan Tokyo avg_temp Plot")

print('5)Earth total average Temp:')
third =df_global_data.plot('year','avg_temp',grid=True,LineWidth=2)
plt.axis([1980, 2009,0,28])
plt.title("Earth total avg_temp Plot")
```

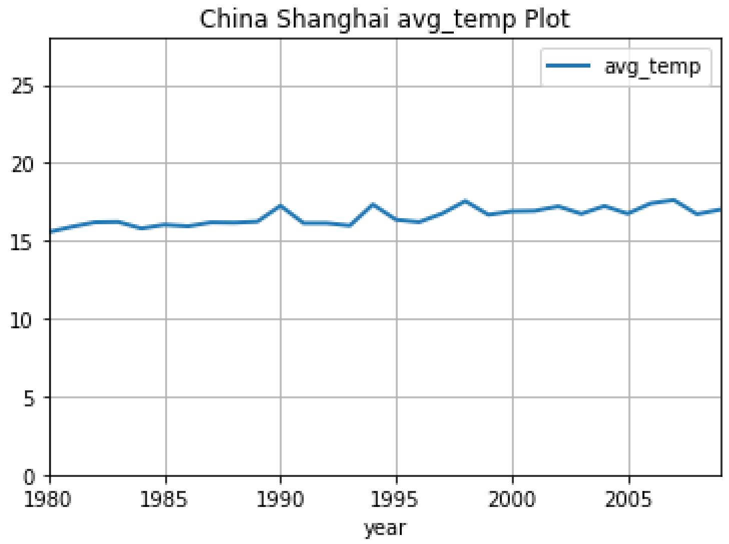
```
1)India,New Dehli:
2)India,Bangalore:
3)China,Shanghai:
4)Japan,Tokyo:
5)Earth total average Temp:
```

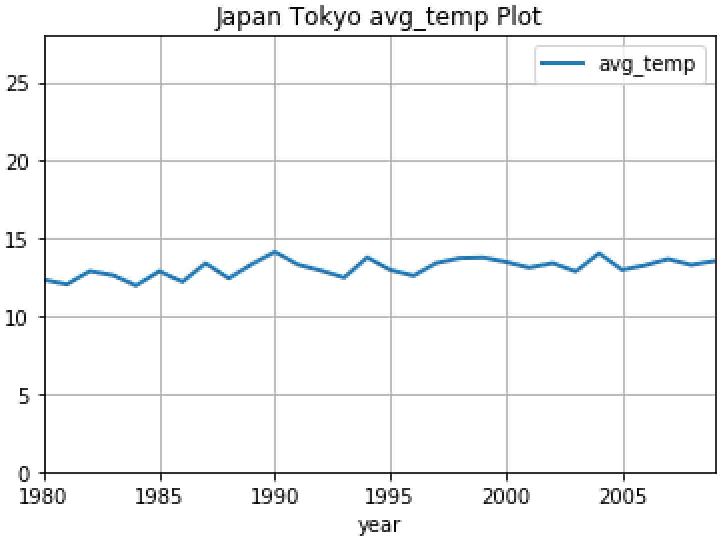
Out[23]:

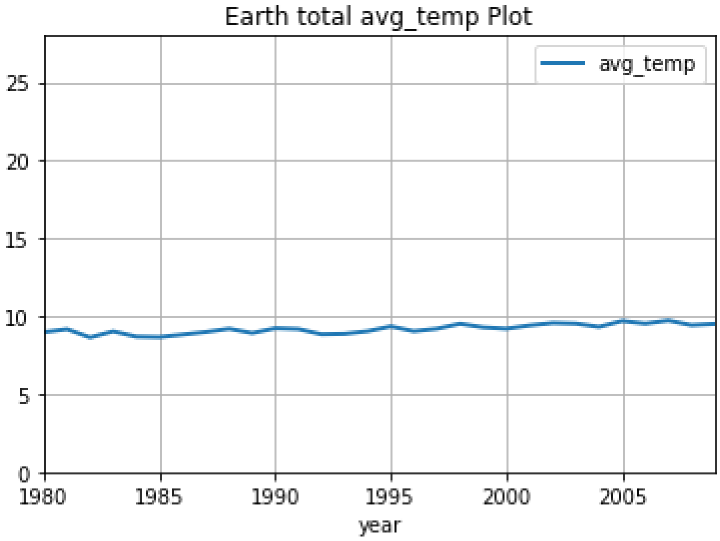
```
Text(0.5, 1.0, 'Earth total avg_temp Plot')
```











Here I created the moving avarge.You can put in how accurate you want it to be and than press enter.

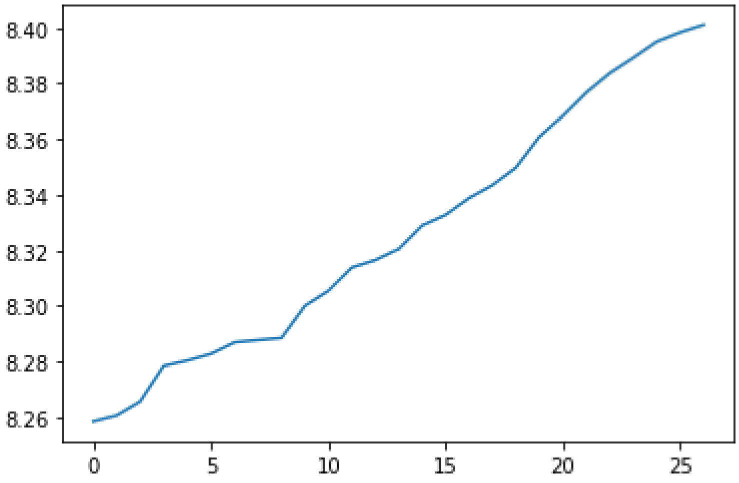
In [24]:

```
numbers = df_global_data.avg_temp
window_size = int(input("how exact do whant your moving avarage plot to be?(265 years)"
))

i = 0
moving_averages = []
while i < len(numbers) - window_size + 1:
    this_window = numbers[i : i + window_size]


    window_average = sum(this_window) / window_size
    moving_averages.append(window_average)
    i += 1

plt.plot(moving_averages)
#plt.axis([1980, 2009,0,10])
```

```
how exact do whant your moving avarage plot to be?(265 years)240
```

Out[24]:

```
[<matplotlib.lines.Line2D at 0x17a085a3848>]
```



Insights

- We can clearly see that the earth is getting warmer.
- It is visible that New Dehli and bangalore is warmer than the averge Temperature.
- Japan,tokyo is colder than the average temperature.
- That is very warm during the industrial times and that it went down from there and than it came back higher than ever.

End of analysis and code