Untitled

January 2, 2019

0.0.1 Data Extracted through SQL Queries

0.0.2 Tool/Tech used to create PDF and visualization: Jupyter Notebook, Python

0.0.3 Step 0 : Importing Libraries

```
In [1]: import pandas as pd
        import seaborn as sns
        import numpy as np
        import math as math
        import matplotlib.pyplot as plt

In [2]: df_bang = pd.read_csv('./data/city_bangalore.csv')
        df_global = pd.read_csv('./data/global_data.csv')

In [3]: df_bang.shape, df_global.shape

Out[3]: ((218, 4), (266, 2))

In [4]: df_global.describe()

Out[4]:                year    avg_temp
        count   266.000000  266.000000
        mean   1882.500000    8.369474
        std      76.931788    0.584747
        min    1750.000000    5.780000
        25%    1816.250000    8.082500
        50%    1882.500000    8.375000
        75%    1948.750000    8.707500
        max    2015.000000    9.830000
```

0.0.4 Inference a: Global Data is from year 1750 to 2015

0.0.5 Inference b: Minimum Average Global Temperature is around 5.7 degrees

0.0.6 Inference c: Maximum Average Global Temperature is around 9.8 degrees

```
In [5]: df_global.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 266 entries, 0 to 265
Data columns (total 2 columns):
year        266 non-null int64
avg_temp    266 non-null float64
dtypes: float64(1), int64(1)
memory usage: 4.2 KB
```

```
In [7]: df_global.head()
```

Out[7]:

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

0.0.7 Note: Filtering Year from 1796 to 2013 as the City I have chosen is Bangalore and it has data for the following years only

```
In [8]: df_global = df_global[( df_global['year'] >= 1796) & (df_global['year'] <= 2013)]
```

```
In [6]: df_bang.head()
```

Out[6]:

	year	city	country	avg_temp
0	1796	Bangalore	India	24.49
1	1797	Bangalore	India	25.18
2	1798	Bangalore	India	24.65
3	1799	Bangalore	India	24.81
4	1800	Bangalore	India	24.85

```
In [11]: df_bang.describe()
```

Out[11]:

	year	avg_temp
count	218.000000	211.000000
mean	1904.500000	24.853081
std	63.075352	0.485181
min	1796.000000	23.300000
25%	1850.250000	24.530000
50%	1904.500000	24.880000
75%	1958.750000	25.165000
max	2013.000000	26.610000

0.0.8 Inference d: Global Data is from year 1796 to 2013

0.0.9 Inference e: Minimum Average Temperature for Bangalore is around 23.3 degrees

0.0.10 Inference f: Maximum Average Temperature for Bangalore is around 26.6 degrees

```
In [13]: df_bang.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 218 entries, 0 to 217
Data columns (total 4 columns):
year        218 non-null int64
city        218 non-null object
country     218 non-null object
avg_temp    211 non-null float64
dtypes: float64(1), int64(1), object(2)
memory usage: 6.9+ KB
```

0.0.11 Note: Filling Null values with mean of the values

```
In [14]: df_bang['avg_temp'] = df_bang['avg_temp'].fillna(df_bang['avg_temp'].mean())
```

```
In [71]: df_bang.head(5)
```

Out[71]:

	year	city	country	avg_temp	MA
0	1796	Bangalore	India	24.49	24.490
1	1797	Bangalore	India	25.18	25.180
2	1798	Bangalore	India	24.65	24.650
3	1799	Bangalore	India	24.81	24.810
4	1800	Bangalore	India	24.85	24.796

0.0.12 Calculating Moving Average for the past 5 years

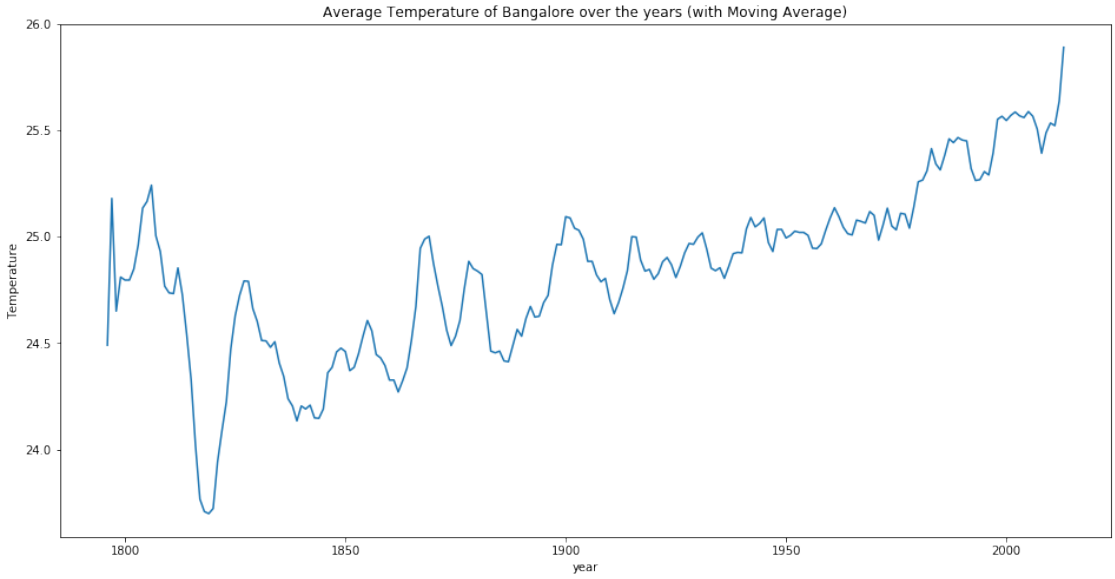
```
In [73]: def calculate_moving_average(df, index):

             list_previous = []
             list_avg = []

             for i in df.iterrows():

                 list_previous.insert(0 , i[1][index])

                 if(len(list_previous) ==5):
                     roll_avg = np.mean(list_previous)
                     list_avg.append(roll_avg)

                     list_previous.pop()
                 else:
                     list_avg.append(i[1][index])

             return list_avg
```

```
In [24]: list_avg_bang = calculate_moving_average(df_bang,3)
```

```
In [25]: df_bang['MA'] = list_avg_bang
```

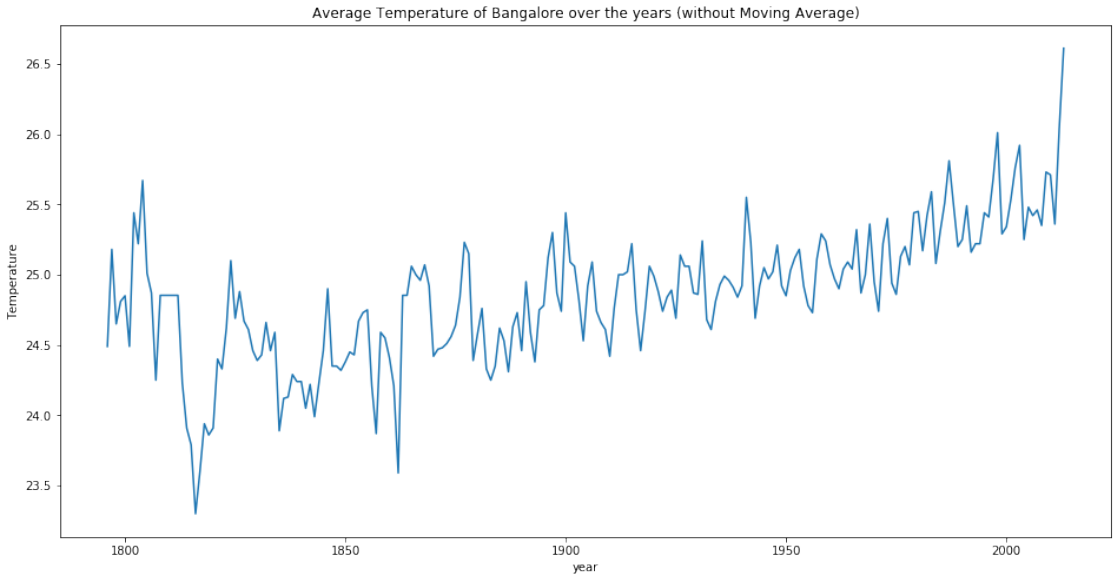
In [85]:
```
plt.subplots(figsize=(16,8))

plt.title('Average Temperature of Bangalore over the years (with Moving Average)')
sns.lineplot(df_bang['year'], df_bang['MA'])
plt.ylabel('Temperature')
```

Out[85]: Text(0, 0.5, 'Temperature')



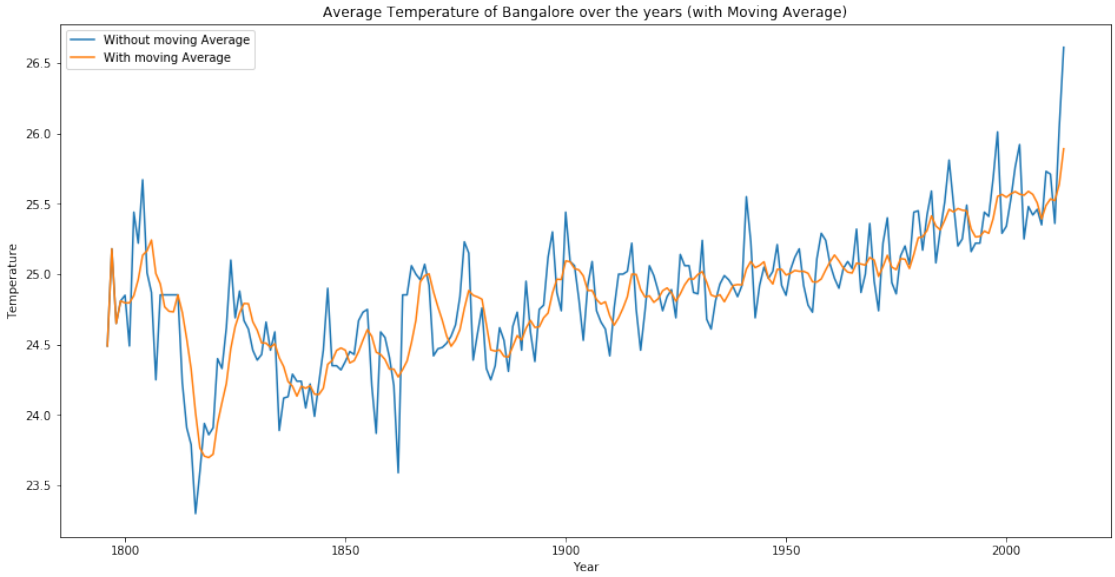
In [86]:
```
plt.subplots(figsize=(16,8))

plt.title('Average Temperature of Bangalore over the years (without Moving Average)')

sns.lineplot(df_bang['year'], df_bang['avg_temp'])
plt.ylabel('Temperature')
```

Out[86]: Text(0, 0.5, 'Temperature')

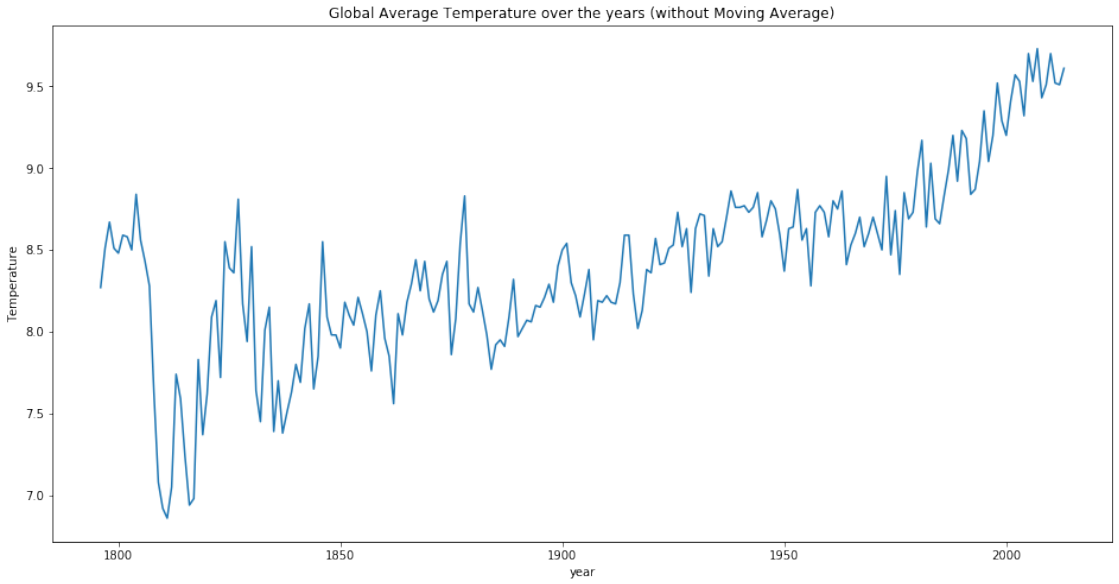


In [83]:
```
plt.subplots(figsize=(16,8))
plt.ylabel('Temperature')
plt.xlabel('Year')
plt.title('Average Temperature of Bangalore over the years (with Moving Average)')

plt.plot(df_bang['year'], df_bang['avg_temp'], label='Without moving Average')
plt.plot(df_bang['year'] , df_bang['MA'], label='With moving Average')
plt.legend(loc='upper left')
```

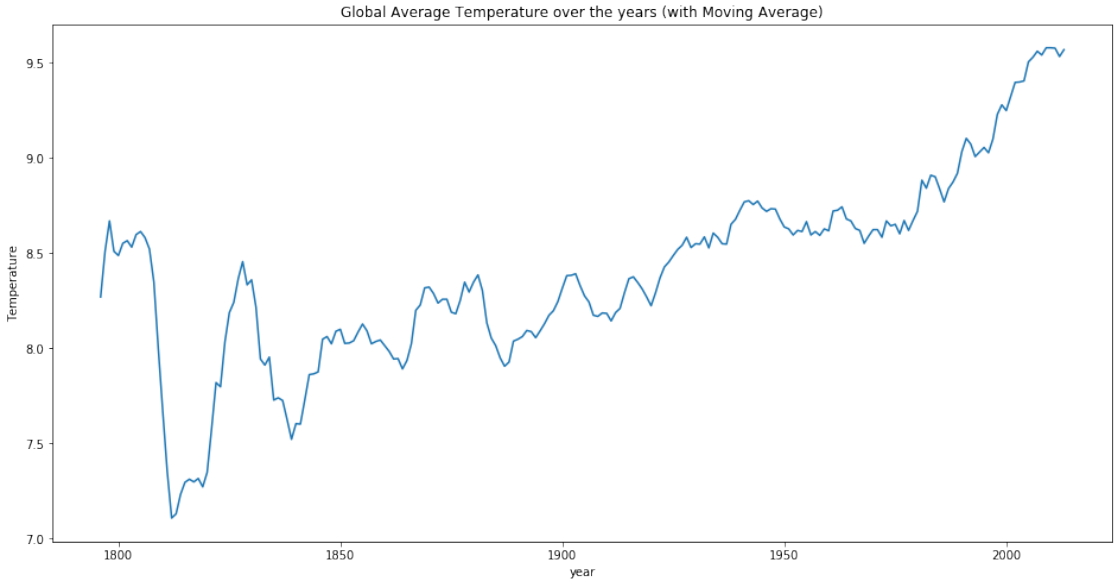
Out[83]: <matplotlib.legend.Legend at 0x1e2ae704080>



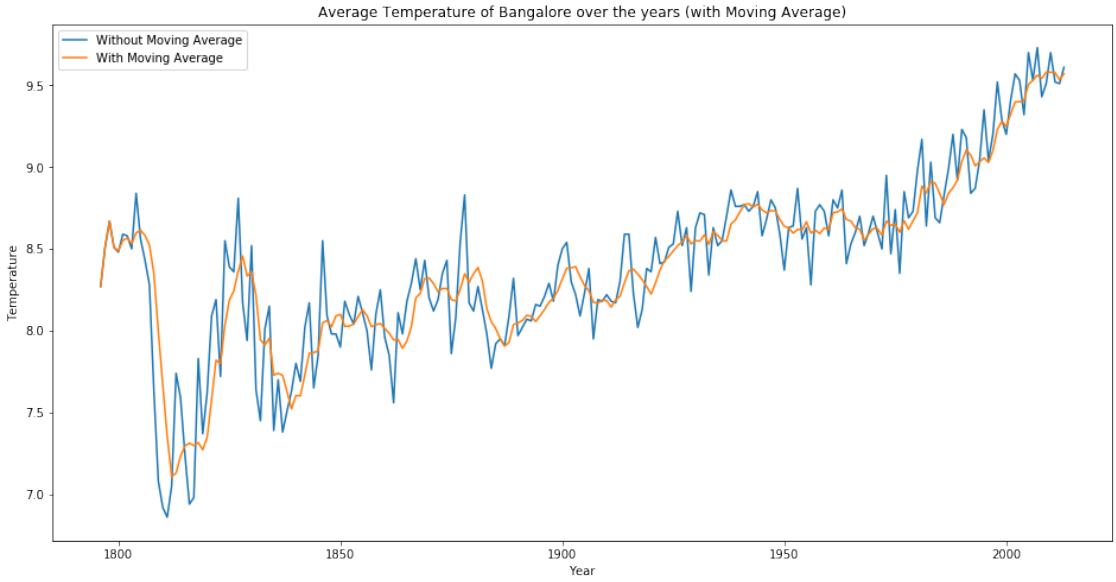
```
In [29]: avg_global = calculate_moving_average(df_global,1)

In [31]: df_global['MA'] = avg_global

In [87]: plt.subplots(figsize=(16,8))

         plt.title('Global Average Temperature over the years (without Moving Average)')

         sns.lineplot(x = df_global['year'] , y=df_global['avg_temp'])
         plt.ylabel('Temperature')

Out[87]: Text(0, 0.5, 'Temperature')
```




```
In [88]: plt.subplots(figsize=(16,8))

         plt.title('Global Average Temperature over the years (with Moving Average)')

         sns.lineplot(x = df_global['year'] , y=df_global['MA'])
         plt.ylabel('Temperature')

Out[88]: Text(0, 0.5, 'Temperature')
```



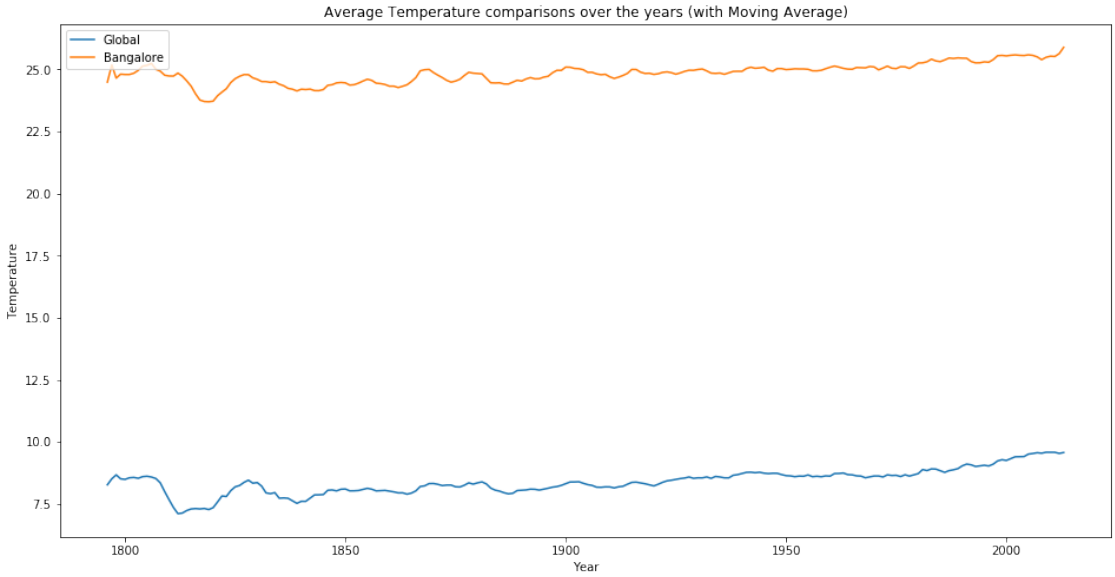

```
In [90]: plt.subplots(figsize=(16,8))
         plt.ylabel('Temperature')
         plt.xlabel('Year')
         plt.title('Average Temperature of Bangalore over the years (with Moving Average)')

         plt.plot(df_global['year'] , df_global['avg_temp'], label='Without Moving Average')
         plt.plot(df_global['year'] , df_global['MA'], label='With Moving Average')
         plt.legend(loc='upper left')

Out[90]: <matplotlib.legend.Legend at 0x1e2b06bee48>
```



In [35]:
```
### Analysis vs global
```

In [92]:
```
plt.subplots(figsize=(16,8))
plt.ylabel('Temperature')
plt.xlabel('Year')
plt.title('Average Temperature comparisons over the years (with Moving Average)')


plt.plot(df_global['year'] , df_global['MA'], label = 'Global')
plt.plot(df_bang['year'], df_bang['MA'], label='Bangalore')
plt.legend(loc='upper left')
```

Out[92]:
```
<matplotlib.legend.Legend at 0x1e2b08de080>
```



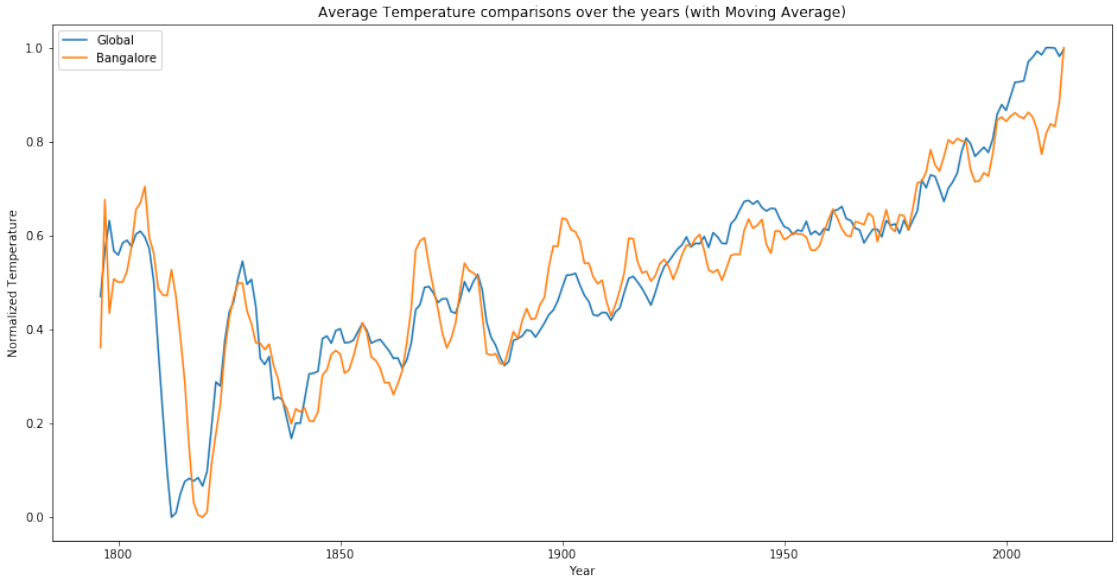
In [96]:
```
from sklearn.preprocessing import MinMaxScaler
ms_global = MinMaxScaler()
ms_bang = MinMaxScaler()

df_global['scaled_temp'] = ms_global.fit_transform(np.array(df_global['MA']).reshape(-1
df_bang['scaled_temp'] = ms_bang.fit_transform(np.array(df_bang['MA']).reshape(-1,1))
```

0.0.13 Scaled Temperature Comparison

In [98]:
```
plt.subplots(figsize=(16,8))
plt.ylabel('Normalized Temperature')
plt.xlabel('Year')
```

```
plt.title('Average Temperature comparisons over the years (with Moving Average)')


plt.plot(df_global['year'] , df_global['scaled_temp'], label = 'Global')
plt.plot(df_bang['year'], df_bang['scaled_temp'], label='Bangalore')
plt.legend(loc='upper left')
```

```
Out[98]: <matplotlib.legend.Legend at 0x1e2b0f699e8>
```



0.0.14 Modelling Data

```
In [39]: X = np.array(df_global['MA'])
         y = np.array(df_bang['MA'])
```

```
In [41]: y.shape
```

```
Out[41]: (218,)
```

```
In [51]: X_train = X[:200]
         X_test  = X[200:]
         y_train = y[:200]
         y_test = y[200:]
```

```
In [45]: from sklearn.linear_model import LinearRegression
```

```
In [56]: lr = LinearRegression()
         lr.fit(X_train.reshape(-1,1), y_train.reshape(-1,1))
```

```
Out[56]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
                  normalize=False)
```

```
In [57]: lr.coef_

Out[57]: array([[0.74115827]])

In [60]: y_test_pred = lr.predict(X_test.reshape(-1,1))

In [68]: y_test_pred.reshape(-1)

Out[68]: array([25.31584555, 25.36920895, 25.46555952, 25.50261744, 25.48038269,
                25.5352284 , 25.59007411, 25.59155643, 25.59600338, 25.67011921,
                25.687907  , 25.71162407, 25.6968009 , 25.72496492, 25.72496492,
                25.7234826 , 25.69087164, 25.71755333])

In [70]: df_pred = pd.DataFrame(np.array([y_test_pred.reshape(-1), y_test]))
         df_pred.head()

Out[70]:           0          1         2          3          4          5          6   \
         0  25.315846  25.369209  25.46556  25.502617  25.480383  25.535228  25.590074
         1  25.290000  25.394000  25.55200  25.566000  25.546000  25.570000  25.586000

                   7          8          9          10         11         12  \
         0  25.591556  25.596003  25.670119  25.687907  25.711624  25.696801
         1  25.568000  25.560000  25.588000  25.566000  25.506000  25.392000

                   13         14         15         16         17
         0  25.724965  25.724965  25.723483  25.690872  25.717553
         1  25.488000  25.534000  25.522000  25.638000  25.890000
```

0.1 Observations

0.1.1 Observation 1: Average Rise in Temperature of Bangalore is 2 Degrees from 1800 to 2013 (Bang is getting hotter year by year)

0.1.2 Observation 2: Average Rise in Global Temperature is around 7.1 to 9.6 from 1815 to 2013 (We are getting hotter around the world day by day)

0.1.3 Observation 3: The rise in global temperature is somewhat consistent with the rise of temperature in Bangalore. (As seen from the normalized scaled graph)

0.1.4 Observation 4: Winter came and conquered around the world from years 1815 to 1825. I tried to google out the reason for this and it turns out that this was due to 1815 eruption of Mount Tambora. This sudden cool climate change approximately killed 90,000 people around the world. Link: https://en.wikipedia.org/wiki/1815_eruption_of_Mount_Tambora

Even Bangalore was affected by this and remain cool through this period

0.1.5 Bonus: I used Linear Regression to find out relation between Bangalore Temperature and Global Temperature