Data Analyses Nanodegree

Project 1
Explore Weather Trends

Elzani Pretorius

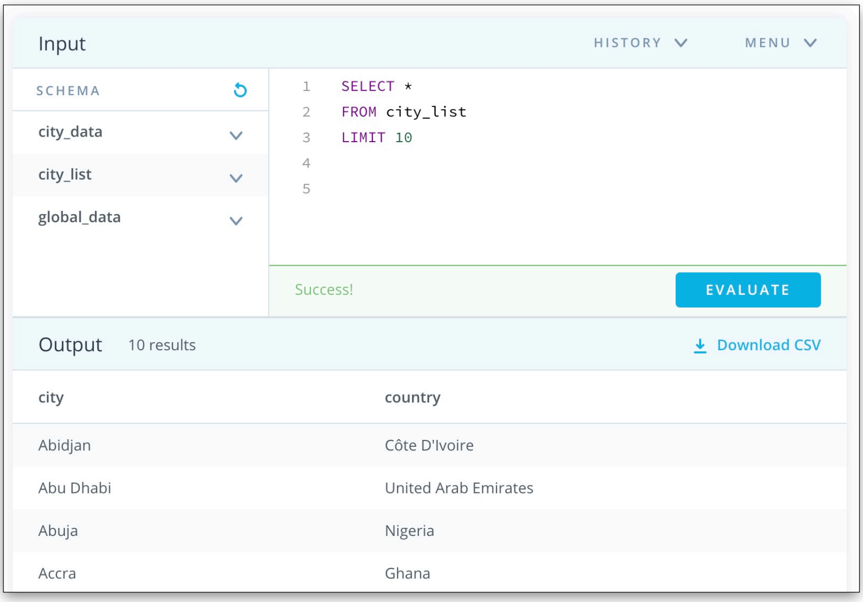
Introduction:

This report will compare temperature data for Boston,USA to global average temperature data. The steps taken and methods used to extract, analyze and visualize the data will be shown in detail.

Analyses steps:

1. Using SQL to inspect and extract data

Before extracting data, the data stored in city_list was inspected. The attributes or columns in the city_list are seen by executing the query in Figure 1. The query in Figure 2 was used to see if there were temperature data for Boston.



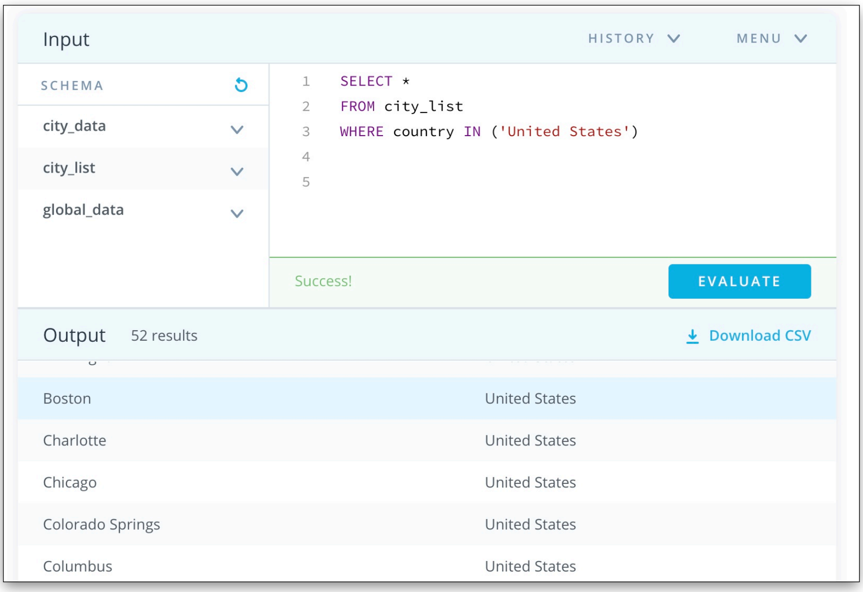
```sql
SELECT *
FROM city_list
LIMIT 10
```

Success!

Output 10 results

city	country
Abidjan	Côte D'Ivoire
Abu Dhabi	United Arab Emirates
Abuja	Nigeria
Accra	Ghana

Figure 1



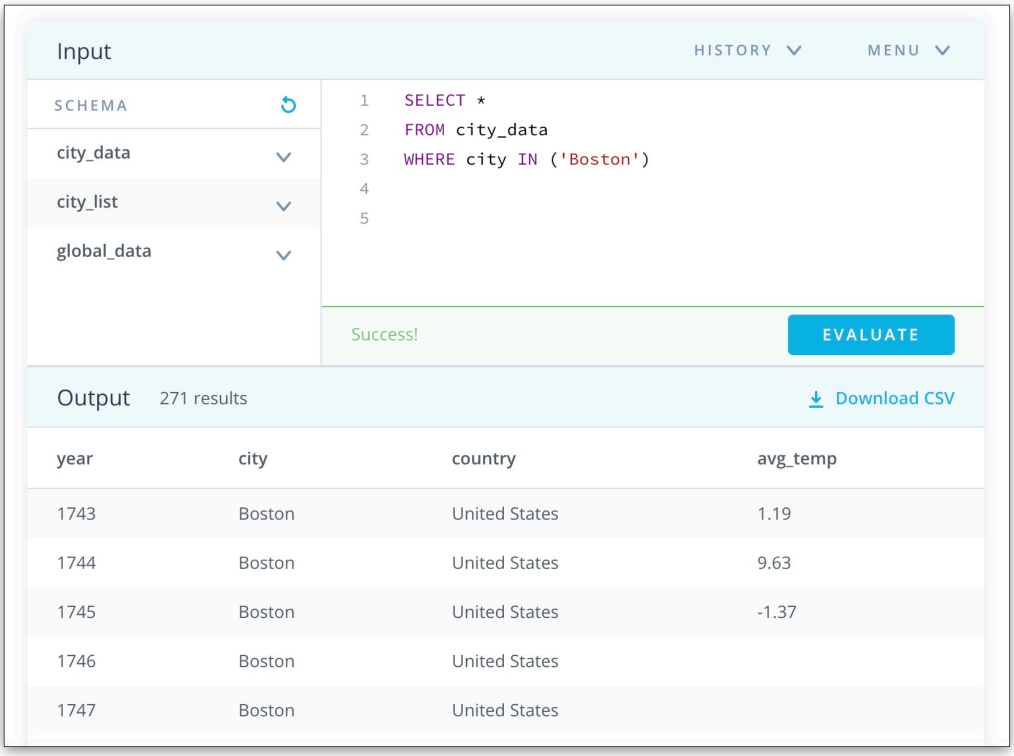
```sql
SELECT *
FROM city_list
WHERE country IN ('United States')
```

Success!

Output 52 results

Boston	United States
Charlotte	United States
Chicago	United States
Colorado Springs	United States
Columbus	United States

Figure 2



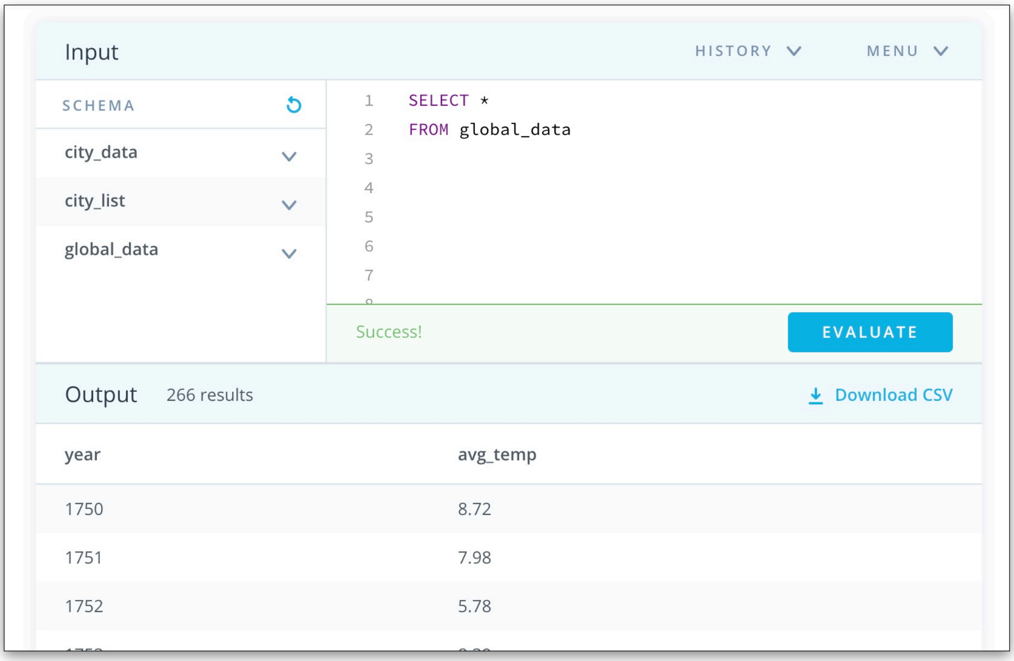
```sql
SELECT *
FROM city_data
WHERE city IN ('Boston')
```

year	city	country	avg_temp
1743	Boston	United States	1.19
1744	Boston	United States	9.63
1745	Boston	United States	-1.37
1746	Boston	United States	
1747	Boston	United States	

Figure 3

All data relating to Boston in the city_data table was selected, as shown in Figure 3, and this data was then downloaded as a csv file.

Similarly, the global_data was downloaded as a csv file, shown in Figure 4 below.



```sql
SELECT *
FROM global_data
```

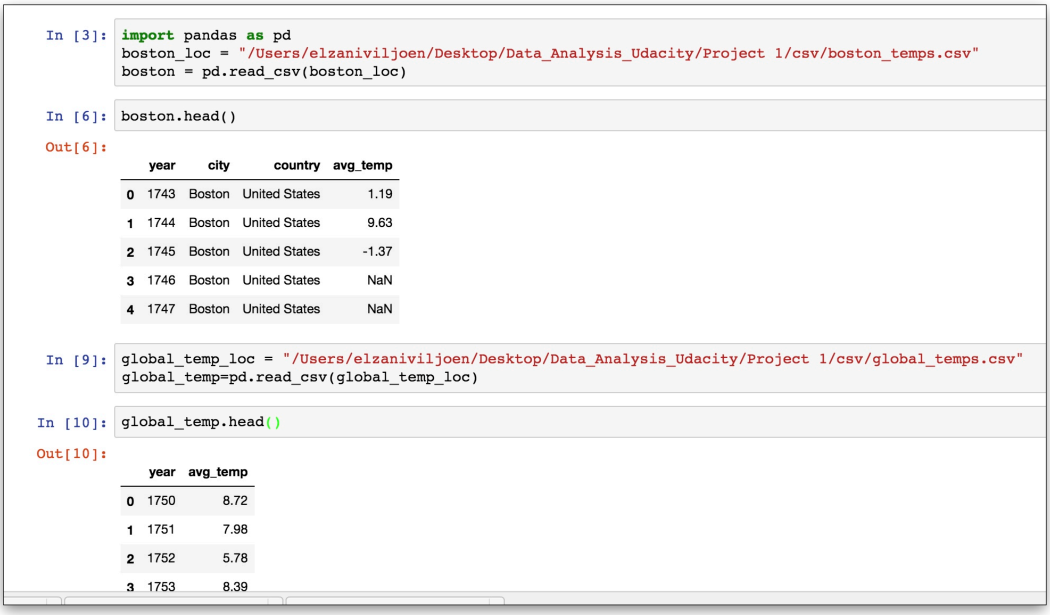
year	avg_temp
1750	8.72
1751	7.98
1752	5.78

Figure 4

2. Using Python and Jupyter Notebook to analyze data

Next, the pandas library was imported and panda dataframes were made out of the csv's obtained in the previous step.

The figure below shows how this was done in Jupiter Notebook, for both csv's. The first few lines of the dataframes , or head, were displayed to check the content.



```
In [3]: import pandas as pd
        boston_loc = "/Users/elzaniviljoen/Desktop/Data_Analysis_Udacity/Project 1/csv/boston_temps.csv"
        boston = pd.read_csv(boston_loc)

In [6]: boston.head()
Out[6]:
```

	year	city	country	avg_temp
0	1743	Boston	United States	1.19
1	1744	Boston	United States	9.63
2	1745	Boston	United States	-1.37
3	1746	Boston	United States	NaN
4	1747	Boston	United States	NaN

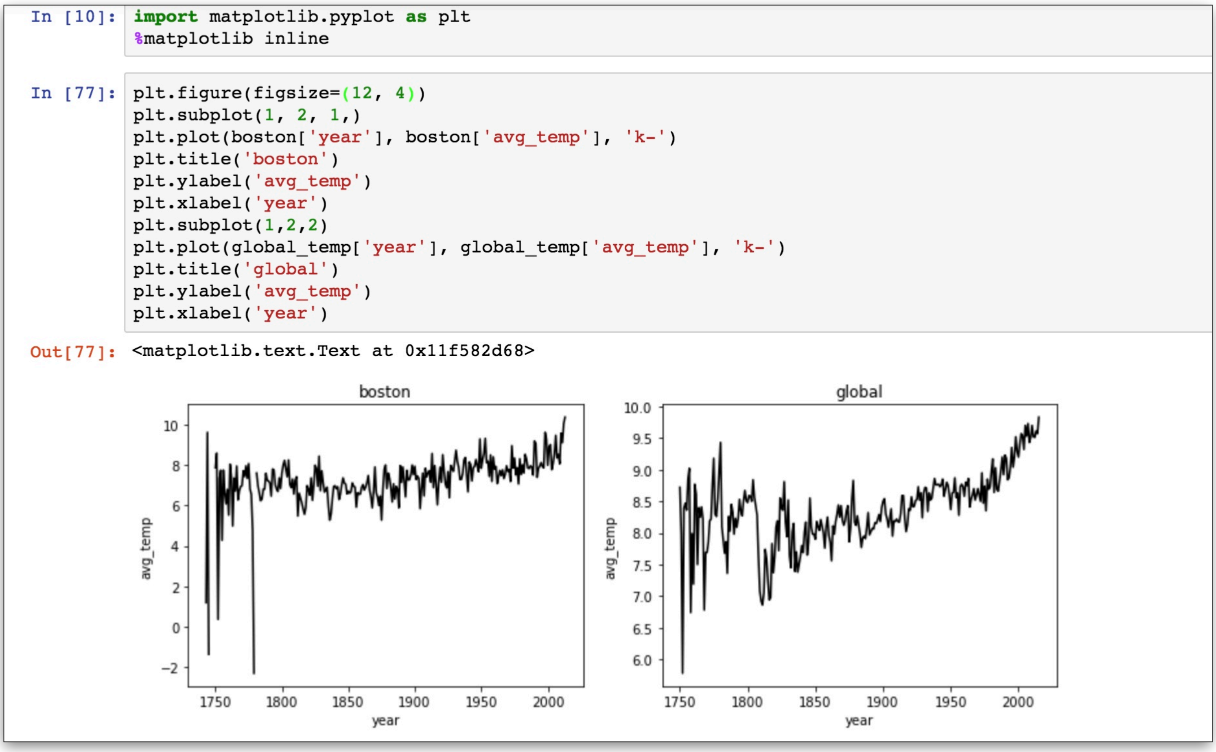
```
In [9]: global_temp_loc = "/Users/elzaniviljoen/Desktop/Data_Analysis_Udacity/Project 1/csv/global_temps.csv"
        global_temp=pd.read_csv(global_temp_loc)

In [10]: global_temp.head()
Out[10]:
```

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

Figure 5

Two figures were plotted, with average temperature (avg_temp) on the y-axis and year on the x-axis. To do this, matplotlib.pyplot had to be imported first, as seen in Figure 5.



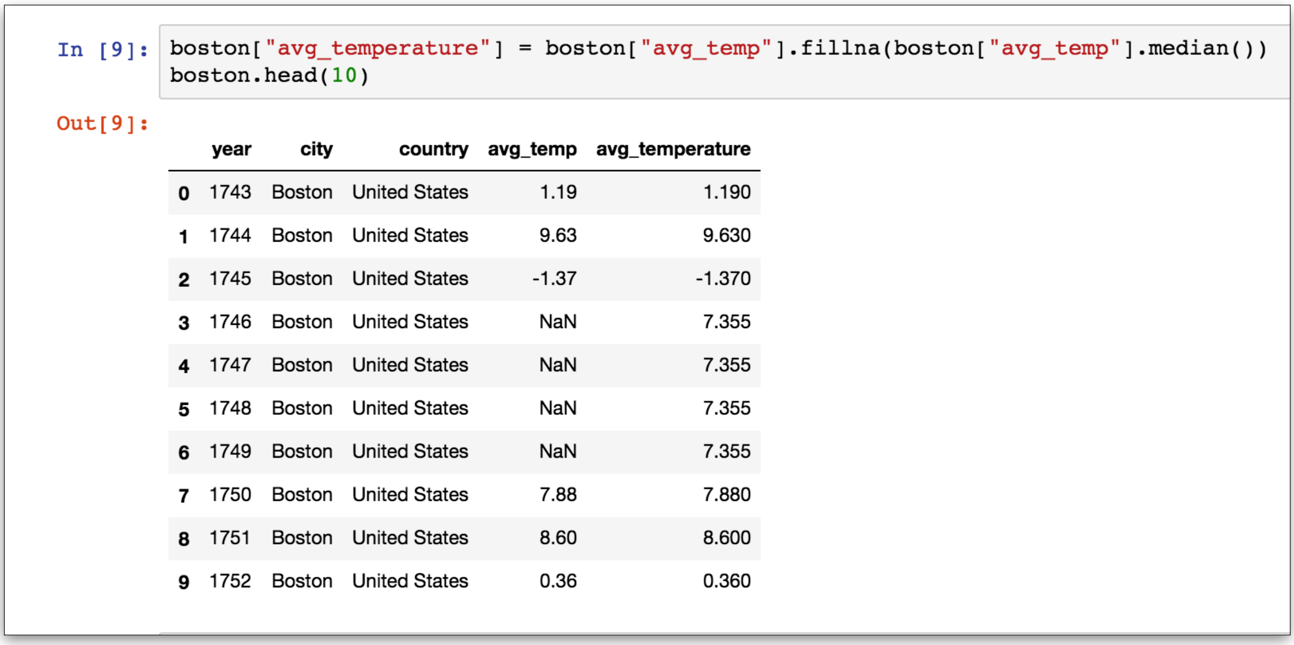
```
In [10]: import matplotlib.pyplot as plt
         %matplotlib inline

In [77]: plt.figure(figsize=(12, 4))
         plt.subplot(1, 2, 1,)
         plt.plot(boston['year'], boston['avg_temp'], 'k-')
         plt.title('boston')
         plt.ylabel('avg_temp')
         plt.xlabel('year')
         plt.subplot(1,2,2)
         plt.plot(global_temp['year'], global_temp['avg_temp'], 'k-')
         plt.title('global')
         plt.ylabel('avg_temp')
         plt.xlabel('year')

Out[77]: <matplotlib.text.Text at 0x11f582d68>
```

Figure 5

The Boston dataset had some missing temperature data. A simple imputation technique was used to substitute each missing value with the median value of the data that was available.
This is shown in Figure 6 below.



```
In [9]: boston["avg_temperature"] = boston["avg_temp"].fillna(boston["avg_temp"].median())
        boston.head(10)
```

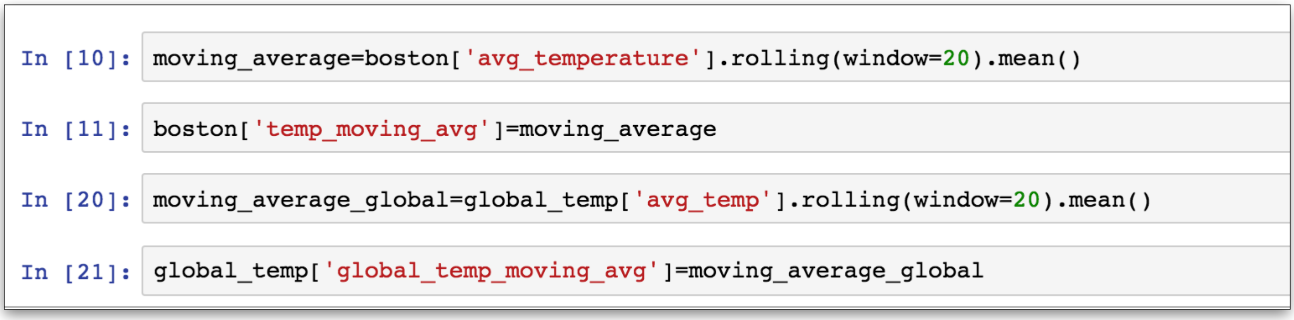
Out[9]:

	year	city	country	avg_temp	avg_temperature
0	1743	Boston	United States	1.19	1.190
1	1744	Boston	United States	9.63	9.630
2	1745	Boston	United States	-1.37	-1.370
3	1746	Boston	United States	NaN	7.355
4	1747	Boston	United States	NaN	7.355
5	1748	Boston	United States	NaN	7.355
6	1749	Boston	United States	NaN	7.355
7	1750	Boston	United States	7.88	7.880
8	1751	Boston	United States	8.60	8.600
9	1752	Boston	United States	0.36	0.360

Figure 6

Moving averages were calculated for the Boston temperature data and the global temperature data.
For the Boston moving averages, the recently added column with no ‘NaN’ values was used for this calculation.

Figure 7 shows the method used in Jupyter.
As seen from the figure, the moving averages were calculated after which they were added to the Boston and global_temp dataframes.



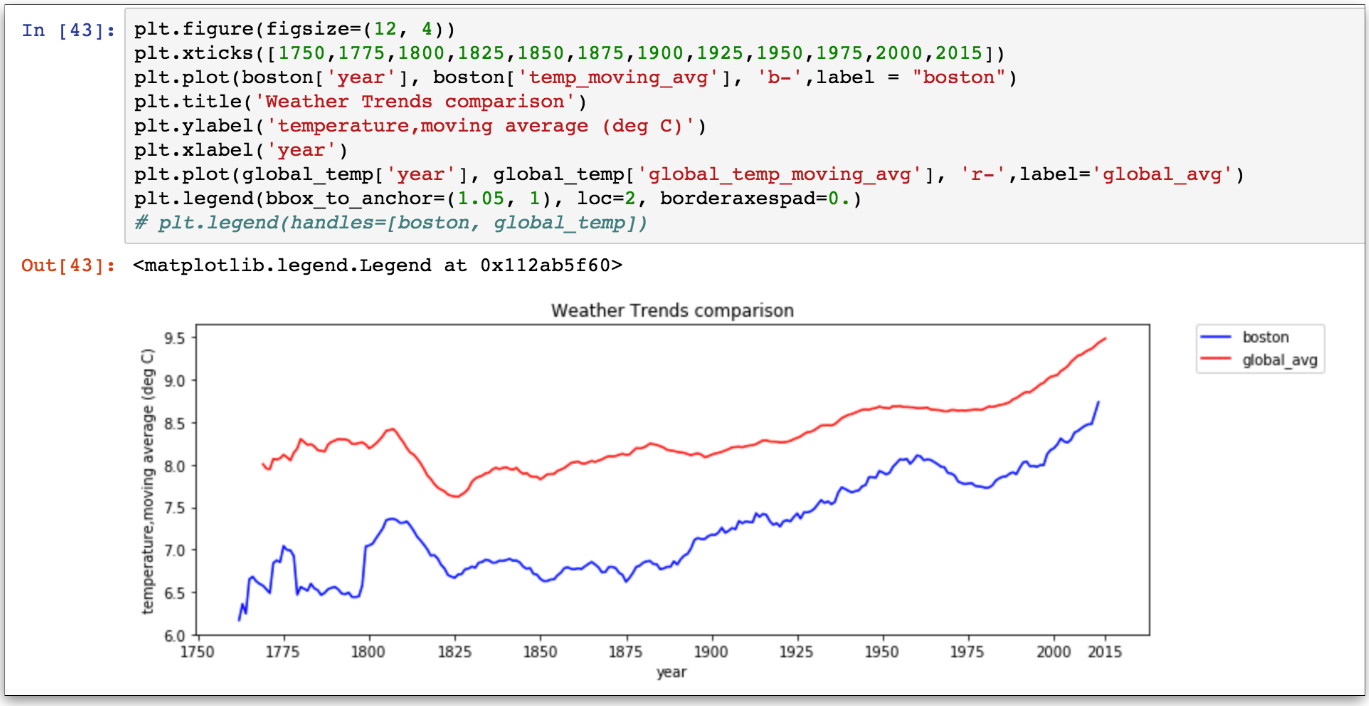
```
In [10]: moving_average=boston['avg_temperature'].rolling(window=20).mean()

In [11]: boston['temp_moving_avg']=moving_average

In [20]: moving_average_global=global_temp['avg_temp'].rolling(window=20).mean()

In [21]: global_temp['global_temp_moving_avg']=moving_average_global
```

Figure 7



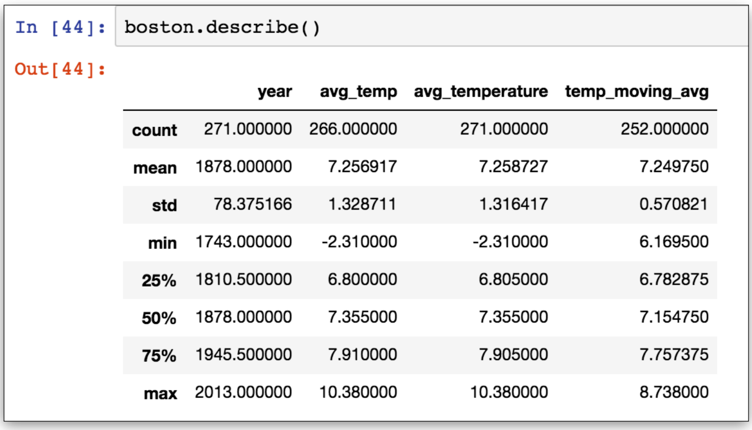
```
In [43]: plt.figure(figsize=(12, 4))
         plt.xticks([1750,1775,1800,1825,1850,1875,1900,1925,1950,1975,2000,2015])
         plt.plot(boston['year'], boston['temp_moving_avg'], 'b-',label = "boston")
         plt.title('Weather Trends comparison')
         plt.ylabel('temperature,moving average (deg C)')
         plt.xlabel('year')
         plt.plot(global_temp['year'], global_temp['global_temp_moving_avg'], 'r-',label='global_avg')
         plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
         # plt.legend(handles=[boston, global_temp])

Out[43]: <matplotlib.legend.Legend at 0x112ab5f60>
```

Figure 8

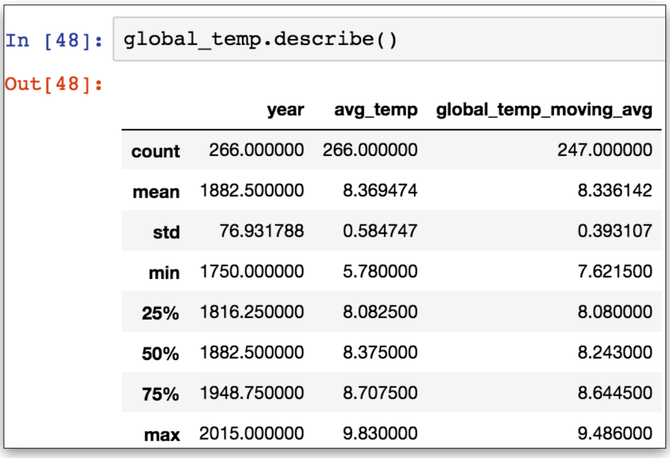
Figure 8 shows the procedure followed and python code used to produce a line chart showing the temperature trend lines for the 'boston' dataset and the 'global_avg' dataset. Referring back to Figure 7, a window of 20 was used for the moving average calculation. This helped to smooth out the lines and allow for more observable trends.
The legend in the right hand corner of the chart provides easy identification of the Boston and global_temp trend lines. As seen on the y-axis, temperature is given in degrees Celsius. The number of ticks on the x-axis were increased, providing higher resolution to gain better insights into the changes over time. A title was also added to the chart to clearly indicate the purpose of the chart.

From the chart it is evident that the average global temperature (global_avg) is consistently higher than the temperature in Boston. In other words, Boston is colder than the global average temperature throughout the years in the data used in this investigation. Figures 9 & 10 below show the mean temperature of Boston over this time period at 7.26 °C and for global average temperature at 8.40 °C.



```
In [44]: boston.describe()
Out[44]:
```

	year	avg_temp	avg_temperature	temp_moving_avg
count	271.000000	266.000000	271.000000	252.000000
mean	1878.000000	7.256917	7.258727	7.249750
std	78.375166	1.328711	1.316417	0.570821
min	1743.000000	-2.310000	-2.310000	6.169500
25%	1810.500000	6.800000	6.805000	6.782875
50%	1878.000000	7.355000	7.355000	7.154750
75%	1945.500000	7.910000	7.905000	7.757375
max	2013.000000	10.380000	10.380000	8.738000



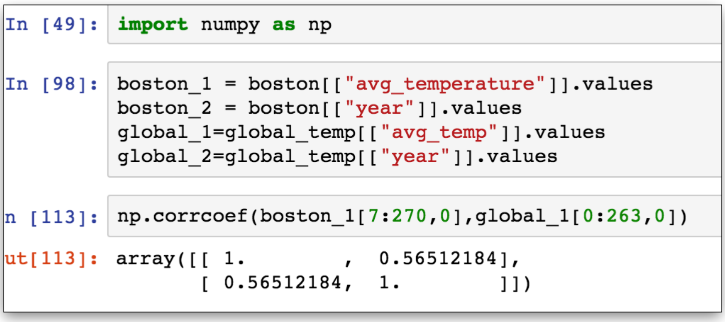
```
In [48]: global_temp.describe()
Out[48]:
```

	year	avg_temp	global_temp_moving_avg
count	266.000000	266.000000	247.000000
mean	1882.500000	8.369474	8.336142
std	76.931788	0.584747	0.393107
min	1750.000000	5.780000	7.621500
25%	1816.250000	8.082500	8.080000
50%	1882.500000	8.375000	8.243000
75%	1948.750000	8.707500	8.644500
max	2015.000000	9.830000	9.486000

Figures 9 & 10

There also seems to be an overall increase in the global_avg and boston trend lines, despite occasional dips along the way. This is particularly noticeable from around 1850 to the end of the lines. This illustrates the gradual warming related to the increase in green house gas emissions that lead to global warming.

Many trends in the global_avg trend line is seen reflected in the boston trend line. For example, the increase in temperature followed by a decrease in temperature at around 1800 can clearly be seen in both lines. The increase here is greater in the boston line than in the global_temp line however. At around 1825 temperatures start to increase again in Boston as well as globally followed by a drop in temperatures until 1850. Once again both lines show this increase followed by a decrease at this point in time. This serves as a visual representation of the correlation between temperatures in Boston and globally. To better understand the correlation between global temperatures and Boston temperatures in Boston the correlation coefficient was calculated as seen below.



```
In [49]: import numpy as np

In [98]: boston_1 = boston[["avg_temperature"]].values
         boston_2 = boston[["year"]].values
         global_1=global_temp[["avg_temp"]].values
         global_2=global_temp[["year"]].values

n [113]: np.corrcoef(boston_1[7:270,0],global_1[0:263,0])

ut[113]: array([[ 1.        ,  0.56512184],
                [ 0.56512184,  1.        ]])
```

Figure 11

The method used to calculate the correlation coefficient requires numpy arrays as inputs, hence numpy was imported and these arrays were created first. The correlation coefficient was then calculated as shown in Figure 11. The boston and global temperature data was used in the correlation coefficient method.
A correlation coefficient of 0.565 was returned, indicating a moderate positive relationship between global temperature and boston temperature. This result confirms that a correlation exists between these temperatures , however this correlation is not very strong.
Furthermore, from Figure 8 it appears that the distance between the two trend lines, hence the temperature delta between boston and global data remains largely unchanged over the years.
To get a rough idea of the difference between boston and average global temperature at any point at time the following procedure can be followed:

1. choose points in time over the given time range, spread equally over
2. Determine the temperature in Boston and the global average temperature at each point
3. Calculate die difference in temperatures at each point
4. Calculate the average of these five temperatures deltas.
5. Use this average delta to predict temperatures at other points in time where there is missing data for either global_avg or boston temperature

For example:
1. Times = (1800,1850,1900,2000)
2. boston temps = (7.59, 6.72, 7.50, 7.85, 8.00)
   global_avg temps= (8.48, 7.90, 8.50, 8.37, 9.70)
3. ∆Temp = (0.89, 1.18, 1.00, 0.52, 1.70)
4. Avg ∆Temp = 1.058 °C

Alternatively, to get a more accurate result, one could find the ∆Temp for all observations in both data sets and get a weighted average ∆Temp.

The section at the start of the Boston trend line, before year 1800 , is very sporadic. This is most likely due to the missing values that had to be substituted with median values. It is likely that the actual temperatures at this point would have been lower than the median value of 7.355 °C that was used.

The graph below includes trend lines from temperature data for Johannesburg. The Johannesburg trend line is included in the legend and is shown in green on the chart.
From the graph you can see that there is less temperature data available for Johannesburg than for Boston and global average temperature. Johannesburg's average yearly temperature is much higher than the other graphs. This is because Johannesburg is closer to the equator than Boston and clearly most other cities used to calculate the global average temperature. The Johannesburg trend line also shows an increase in temperature over time, this agrees with the other two trend lines displayed in the chart.

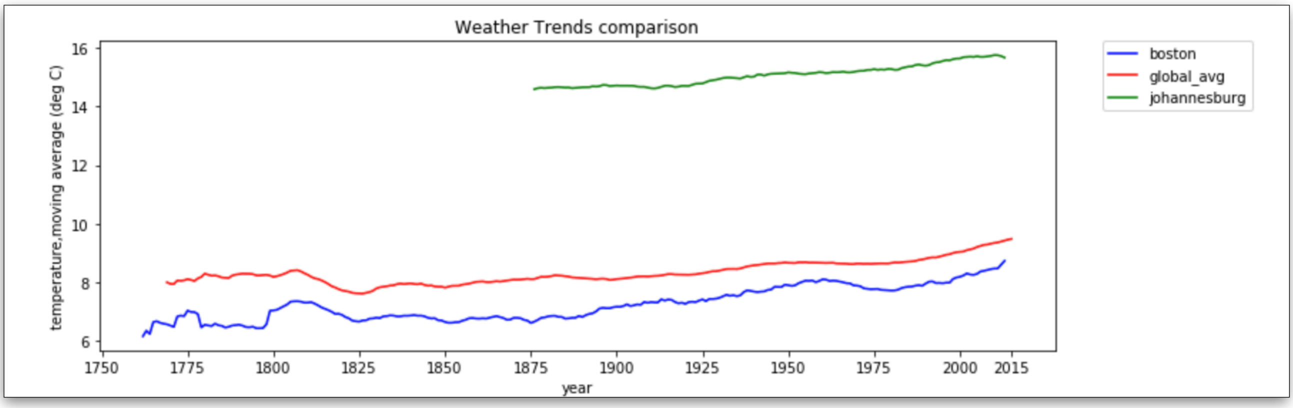


Figure 12