UDACITY Data Analysis Nanodegree

Project:- Explore Weather Trends

Grant Patience, 19th June 2019

Table of Contents

1. Determine Objectives and Assess the Situation

The task is to extract data pertaining to Global and City temperatures and to subsequently analyse the data, setting out objectives, noting observations and determining our conclusions. Success is graded against the UDACITY rubric, noted below.

1.1 Outline of Steps

- We discuss what it is we wish to achieve, and decide which questions we want to ask of the data
- We will extract the data we need from UDACITY - Global average temperature, and Local City average temperature, in this case Edinburgh is used
- Import the data into Python for analysis
- Perform some rudimentary exploratory data analysis to help understand our data
- Create some visualisations, including a line chart
- Create our conclusions based on the data

1.2 What are the desired outputs of the project?

Submission Your submission should be a PDF that includes:

- An outline of steps taken to prepare the data to be visualized in the chart, such as:
- What tools did you use for each step? (Python, SQL, Excel, etc)
- How did you calculate the moving average?
- What were your key considerations when deciding how to visualize the trends?
- Line chart with local and global temperature trends
- At least four observations about the similarities and/or differences in the trends

Report Success Criteria as determined by UDACITY

Student is able to extract data from a database using SQL.

- The SQL query used to extract the data is included.
- The query runs without error and pulls the intended data. Student is able to manipulate data in a spreadsheet or similar tool.
- Moving averages are calculated to be used in the line chart.

Student is able to create a clear data visualization.

- A line chart is included in the submission.
- The chart and its axes have titles, and there's a clear legend (if applicable).
- Student is able to interpret a data visualization.

The student includes four observations about their provided data visualization.

- The four observations are accurate.

1.3 What Questions Are We Trying To Answer?

- Is your city hotter or cooler on average compared to the global average?
- Has the difference been consistent over time?
- How do the changes in your city’s temperatures over time compare to the changes in the global average?
- What does the overall trend look like?
- Is the world getting hotter or cooler?
- Has the trend been consistent over the last few hundred years?

---

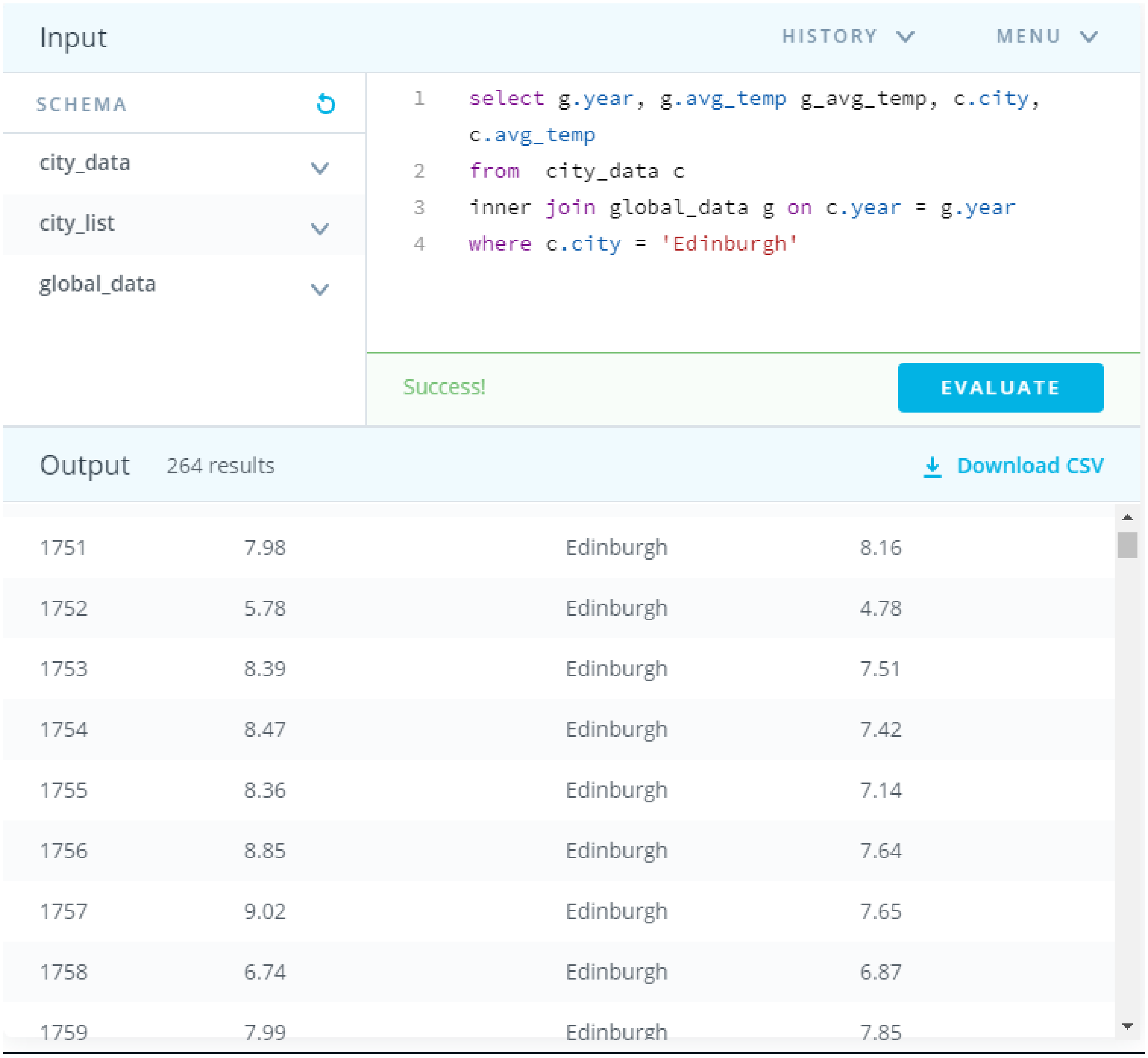
2. Data Understanding

The second stage of the process is where we acquire the data listed in the project resources. This initial collection includes extraction from UDACITY using SQL, and subsequently loaded into Python and analysed in Jupyter notebook.

2.1 SQL Extraction

The following SQL statement was written and subtmitted to the UDACITY virtual environment for extraction. The data was downloaded locally to AllData.csv

```
select g.year, g.avg_temp g_avg_temp, c.city, c.avg_temp
from  city_data c
inner join global_data g on c.year = g.year
where c.city = 'Edinburgh'
```



```
select g.year, g.avg_temp g_avg_temp, c.city,
c.avg_temp
from  city_data c
inner join global_data g on c.year = g.year
where c.city = 'Edinburgh'
```

1751	7.98	Edinburgh	8.16
1752	5.78	Edinburgh	4.78
1753	8.39	Edinburgh	7.51
1754	8.47	Edinburgh	7.42
1755	8.36	Edinburgh	7.14
1756	8.85	Edinburgh	7.64
1757	9.02	Edinburgh	7.65
1758	6.74	Edinburgh	6.87
1759	7.99	Edinburgh	7.85

2.2 Exploring the data

We import the data into Python and do any data preperation required, followed by some exploratory data analysis inside a Jupyter notebook

In [1]:

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

In [2]:

```
#Create variable to store path to file location
path = 'C:/Users/jammy/OneDrive/Documents/Developments/Data and Analysis/Udacity Data A
nalysis/Weather Project/AllData.csv'
```

In [3]:

```
# Read the file
df = pd.read_csv(path,sep=',')
```

2.2.1 Exploration to Understand the Data

In [4]:

```
df.shape
```

Out[4]:

```
(264, 4)
```

In [5]:

```
# Show columns
df.columns
```

Out[5]:

```
Index(['year', 'g_avg_temp', 'city', 'avg_temp'], dtype='object')
```

In [6]:

```
# Show the first 10 records
df.head(10)
```

Out[6]:

	year	g_avg_temp	city	avg_temp
0	1750	8.72	Edinburgh	8.41
1	1751	7.98	Edinburgh	8.16
2	1752	5.78	Edinburgh	4.78
3	1753	8.39	Edinburgh	7.51
4	1754	8.47	Edinburgh	7.42
5	1755	8.36	Edinburgh	7.14
6	1756	8.85	Edinburgh	7.64
7	1757	9.02	Edinburgh	7.65
8	1758	6.74	Edinburgh	6.87
9	1759	7.99	Edinburgh	7.85

In [7]:

```
df.isnull().sum()
```

Out[7]:

```
year         0
g_avg_temp   0
city         0
avg_temp     0
dtype: int64
```

In [8]:

```
df.describe()
```

Out[8]:

	year	g_avg_temp	avg_temp
count	264.000000	264.000000	264.000000
mean	1881.500000	8.359394	7.603030
std	76.354437	0.575184	0.618545
min	1750.000000	5.780000	4.780000
25%	1815.750000	8.077500	7.192500
50%	1881.500000	8.365000	7.595000
75%	1947.250000	8.700000	7.955000
max	2013.000000	9.730000	9.100000

2.2.2 Calculating a Rolling Average

As per instruction, we determine the rolling average over a 10 year period. We do this in Python by using the rolling function, stipulating the window as 10 years. The resulting averages are added back to our dataset as new columns

In [9]:

```
df['g_rolling_avg'] = df['g_avg_temp'].rolling(window=10).mean()
df['c_rolling_avg'] = df['avg_temp'].rolling(window=10).mean()
```

In [10]:

```
df.head(15)
```

Out[10]:

	year	g_avg_temp	city	avg_temp	g_rolling_avg	c_rolling_avg
0	1750	8.72	Edinburgh	8.41	NaN	NaN
1	1751	7.98	Edinburgh	8.16	NaN	NaN
2	1752	5.78	Edinburgh	4.78	NaN	NaN
3	1753	8.39	Edinburgh	7.51	NaN	NaN
4	1754	8.47	Edinburgh	7.42	NaN	NaN
5	1755	8.36	Edinburgh	7.14	NaN	NaN
6	1756	8.85	Edinburgh	7.64	NaN	NaN
7	1757	9.02	Edinburgh	7.65	NaN	NaN
8	1758	6.74	Edinburgh	6.87	NaN	NaN
9	1759	7.99	Edinburgh	7.85	8.030	7.343
10	1760	7.19	Edinburgh	7.13	7.877	7.215
11	1761	8.77	Edinburgh	7.93	7.956	7.192
12	1762	8.61	Edinburgh	7.13	8.239	7.427
13	1763	7.50	Edinburgh	6.92	8.150	7.368
14	1764	8.40	Edinburgh	7.13	8.143	7.339

2.2.3 Visualising the Data

Key considerations to take into account when visualising this dataset are

- data is time series, ideal for plotting along an x-axis in a line chart.
- Both global and local temperatures use the same measure, so we can plot both in the same chart if desired, using the same y-axis

2.2.4 Line Chart

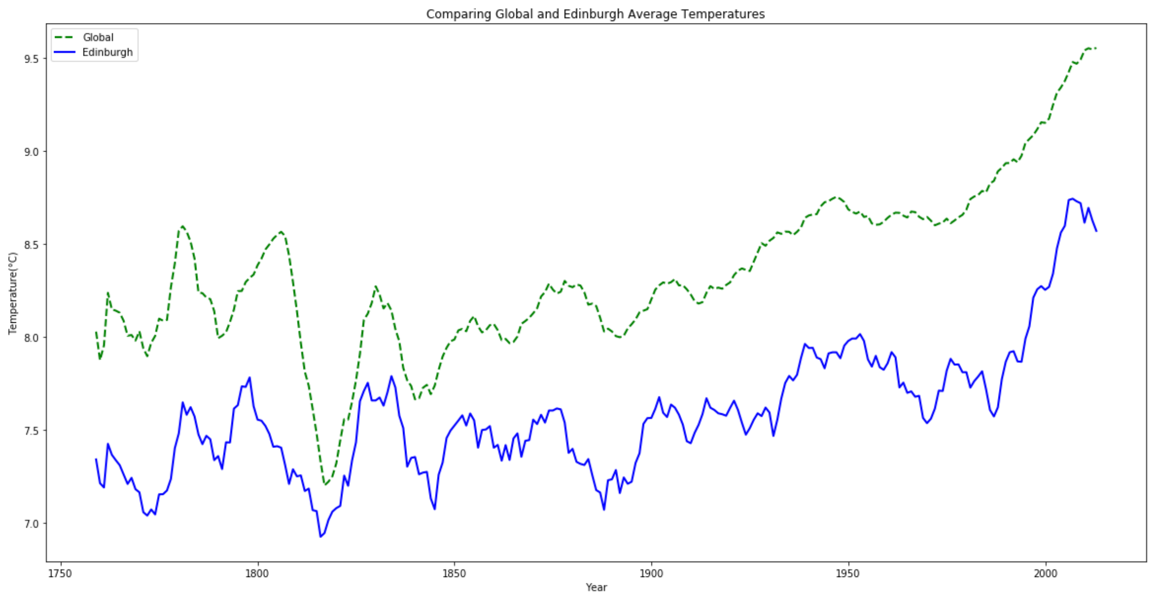
In [11]:

```
# Create our chart area
fig = plt.figure()
plt.figure(figsize=(20,10))
ax = plt.axes()

#Get the data for our axis
x = df["year"]
y1 = df["g_rolling_avg"]
y2 = df["c_rolling_avg"]

# Plot items to chart
plt.title("Comparing Global and Edinburgh Average Temperatures")
plt.xlabel("Year")
plt.ylabel("Temperature(°C)");
plt.plot( x, y1, marker='', color='green', linewidth=2, linestyle='dashed', label="Glob
al")
plt.plot( x, y2, marker='', color='blue', linewidth=2, label="Edinburgh")
plt.legend()
```

Out[11]:

```
<matplotlib.legend.Legend at 0x1ddd9227630>
```

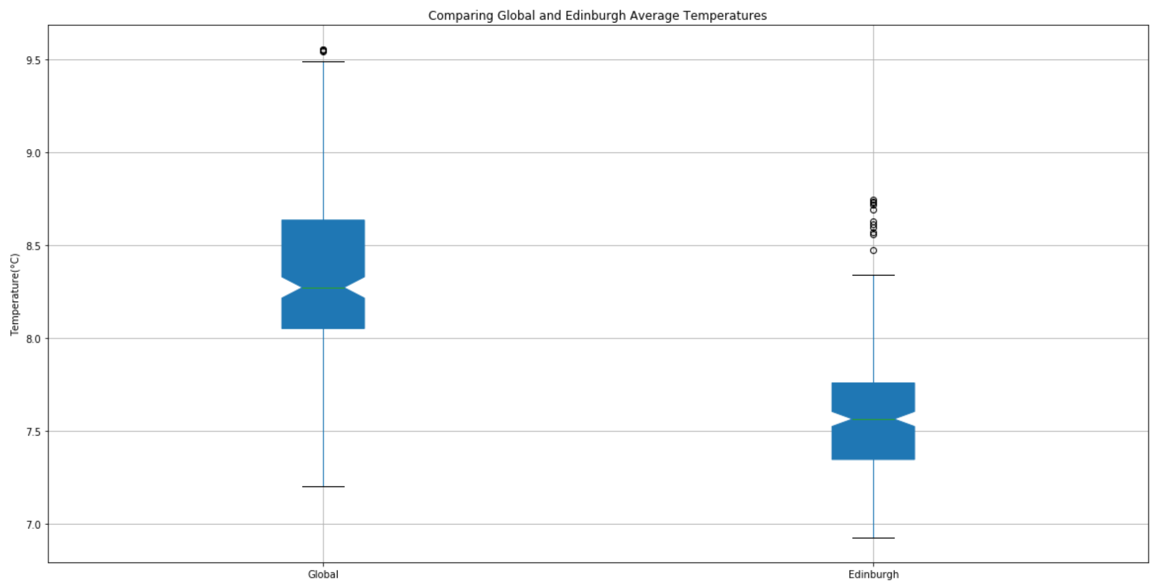
```
<Figure size 432x288 with 0 Axes>
```



2.2.5 Box Plot

In [12]:

```
plt.figure(figsize=(20,10))
boxplot = df.boxplot(column=['g_rolling_avg', 'c_rolling_avg'],  notch=True,
                         vert=True,
                         patch_artist=True,
                    )

plt.title('Comparing Global and Edinburgh Average Temperatures')
plt.xticks([1, 2], ['Global', 'Edinburgh'])
plt.ylabel("Temperature(°C)");


plt.show()
```



3. Observations and Conclusion

- **Q - Is your city hotter or cooler on average compared to the global average? -**
  - A - The city of Edinburgh is slightly cooler than the global average, by around 1 degree
- **Q - Has the difference been consistent over time? -**
  - A - On the whole, Yes, the difference appears to be consistent over time. However, during the period of 1800 to 1840 the margin is much closer.
- **Q - How do the changes in your city's temperatures over time compare to the changes in the global average? -**
  - A - On the whole, Yes, the trends align over time. However, one exception is after the year 2000, Edinburghs average temperature drops, while the global temperature continues to rise.
- **Q - What does the overall trend look like? -**
  - A - Both are trending upwards in temperature, indicating a similarity between the average temperature of the city and the global average.
- **Q - Is the world getting hotter or cooler? -**
  - A - The world is growing warmer over time.
- **Q - Has the trend been consistent over the last few hundred years? -**
  - A - This trend has been consistent rising over the last 150 years. The start of the rise in temperatures appears to co-incide with the start of the Industrial Revolution.

Conclusion

The data suggests that as global temperature changes, the local city temperate changes in line which would indicate a correlation. With regards to the trend upwards over the last 150 years, this would support the claim that global warming is linked to the increase in industrialisation around the globe since the Industrial Revolution in the mid 19th centuary.

References

- https://chrisalbon.com/python/data_wrangling/pandas_moving_average/ (https://chrisalbon.com/python/data_wrangling/pandas_moving_average/)
- https://matplotlib.org/3.1.0/gallery/statistics/boxplot_demo.html7 (https://matplotlib.org/3.1.0/gallery/statistics/boxplot_demo.html7)
- http://blog.bharatbhole.com/creating-boxplots-with-matplotlib/ (http://blog.bharatbhole.com/creating-boxplots-with-matplotlib/)
- https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.boxplot.html (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.boxplot.html)