Explore Weather Trends

This notebook includes the solution for the first project "Explore Weather Trends" in Data Analyst Nanodegree.

Tools

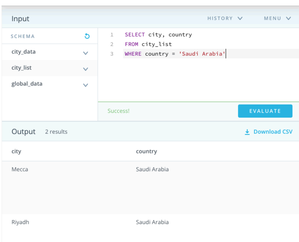
- SQL
- Python 3.8
  - Pandas
  - Matplotlib
  - Seaborn
- VS Code (Jupyter Notebook)

Haitaa Mohamad Alzahrani

1. Extract Data By SQL

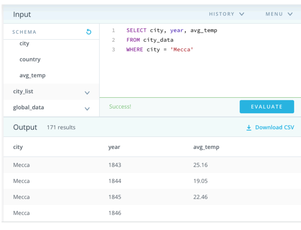
1.1 Finding The Nearest City

I live in Jeddah, so I retrived all records for Saudi Arabia to find out the nearest city to me as shown below.



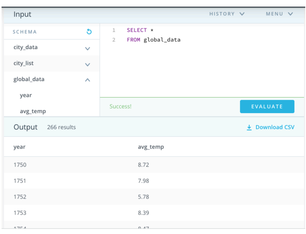
1.2 Finding Mecca Data

I retrived all records for Mecca and export them into a csv as shown below.



1.3 Export Global Data

I retrived all records in global_data table and export them into a csv as shown below.



2. Explore Data Using Python

Here, I used Pyhton, especially Pandas library to read and explore the data.

```
In [32]: # Import required library
         import pandas as pd
         import matplotlib.pyplot as plt
         import seaborn as sns

In [33]: # Reading data
         local_data = pd.read_csv("mecca.csv")

         len(local_data)

Out[33]: 171

In [34]: local_data.head()

Out[34]:
```

	city	year	avg_temp
0	Mecca	1843	25.16
1	Mecca	1844	19.05
2	Mecca	1845	22.46
3	Mecca	1846	NaN
4	Mecca	1847	NaN

```
In [35]: # Reading data
         global_data = pd.read_csv("global.csv")

         len(global_data)

Out[35]: 266

In [36]: global_data.head()

Out[36]:
```

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

3. Prepare Data

```
In [37]: # Remove unwanted column
         local_data = local_data.drop(['City'], axis=1)

In [38]: # As shown on the previous output, we have ,missing values,
         # I would like to know the number of NaN values in each table.
         # isnull() returns bool for each column, and I need the sum of True values
         local_data.isnull().sum()

Out[38]: year        0
         avg_temp    15
         dtype: int64

In [39]: global_data.isnull().sum()

Out[39]: year        0
         avg_temp    0
         dtype: int64

In [41]: # Luckly, the missing avg_temp data are associated with the a continues period [1846-1860], so I'll remove them.
         local_data[local_data['avg_temp'].isnull()]

Out[41]:
```

	year	avg_temp
3	1846	NaN
4	1847	NaN
5	1848	NaN
6	1849	NaN
7	1850	NaN
8	1851	NaN
9	1852	NaN
10	1853	NaN
11	1854	NaN
12	1855	NaN
13	1856	NaN
14	1857	NaN
15	1858	NaN
16	1859	NaN
17	1860	NaN

```
In [42]: # Remove NaN and reset index
         local_data=local_data.dropna().reset_index(drop=True)

In [43]: # Just tot check the index
         # You can see that year 1861 has the updated index
         local_data.head()

Out[43]:
```

	year	avg_temp
0	1843	25.16
1	1844	19.05
2	1845	22.46
3	1861	23.98
4	1862	24.13

```
In [44]: # Rename avg_temp column
         local_data = local_data.rename(columns={"avg_temp": "local_avg_temp"})

In [45]: global_data = global_data.rename(columns={"avg_temp": "global_avg_temp"})

In [46]: # Merge the 2 tables by using the intersection of year column
         all_data = pd.merge(local_data, global_data, how='inner')

In [47]: all_data.head()

Out[47]:
```

	year	local_avg_temp	global_avg_temp
0	1843	25.16	8.17
1	1844	19.05	7.65
2	1845	22.46	7.85
3	1861	23.98	7.85
4	1862	24.13	7.56

```
In [48]: all_data.shape

Out[48]: (156, 3)
```

4. Calculate Moving Average

The moving average is used to:

- Make it easier to observe long-term trends by smoothing out data.
- Eliminate daily fluctuations and prevent observations from being too volatile to interpret.

There are couple of methods to calculate the moving average in Pyhton and I used **rolling(window).mean()**.

```
In [50]: # 5 is selected to calculate the MA as the period is not too large
         all_data['local_ma'] = all_data['local_avg_temp'].rolling(window=10).mean()

In [51]: all_data['global_ma'] = all_data['global_avg_temp'].rolling(window=10).mean()

In [52]: all_data.head(10)

Out[52]:
```

	year	local_avg_temp	global_avg_temp	local_ma	global_ma
0	1843	25.16	8.17	NaN	NaN
1	1844	19.05	7.65	NaN	NaN
2	1845	22.46	7.85	NaN	NaN
3	1861	23.98	7.85	NaN	NaN
4	1862	24.13	7.56	NaN	NaN
5	1863	22.87	8.11	NaN	NaN
6	1864	25.43	7.98	NaN	NaN
7	1865	25.60	8.18	NaN	NaN
8	1866	25.42	8.29	NaN	NaN
9	1867	25.62	8.44	23.972	8.008

5. Plot The

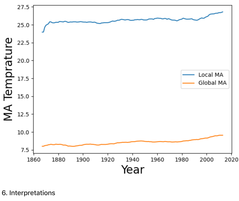
Before plotting using the Moving Average, I'll plot the original average to see the difference

```
In [53]: p = sns.lineplot()

         # Set labels and legend
         p.set_xlabel("Year", fontsize = 20)
         p.set_ylabel("MA Temprature", fontsize = 20)
         # plt.legend(loc='center right')

         sns.lineplot(x='year', y='local_ma', data=all_data, label= 'Local MA')
         sns.lineplot(x='year', y='global_ma', data=all_data, label= 'Global MA')

         plt.show()
```



6. Interpretations

- Globally and locally, temperatures are rising as the world becomes hotter.
- Obviously, Mecca has consistently experienced higher temperatures over the years compared to the global average.
- The change in the Mecca's temperature over time was higher than what has been seen in the global average.
- The change in the global's temperature seems more stable compared to Mecca.
- According to the chart, there is a noticeable increase in the average temperature in Mecca in the period between 1860 to 1870.