Weather Exploration Project - Udacity

Diego Merino Muñoz

First of all, I used the following SQL queries to extract the data:

- For my local City:

```
SELECT
    year,
    avg_temp as Average_Temperature
FROM
    city_data
WHERE
    city = 'Madrid'
ORDER BY
    year
```

- For the Global Data:

```
SELECT
    year,
    avg_temp as average_temperature
FROM
    global_data
ORDER BY
    year
```

Now, we import the necessary libraries to take on our project:

In [ ]:
```
# Importing needed libraries for the exploration and Setting the matplotlib f
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import csv

plt.rcParams["figure.figsize"] = (16,5)
```

Now it's time to load our data, selecting our index col and parsing the dates in case we need to manage them.

In [ ]:
```
# Loading the CSVs as Pandas dataframes, with the year as index and the dates

city_data = pd.read_csv('citywide-results.csv', index_col='year', parse_dates
global_data = pd.read_csv('global-data.csv', index_col = 'year', parse_dates=
```

We check for Null values that will cause trouble when calculating the moving average:

In [ ]:
```
# Checking for null Data on our dataframes
print('There are {} Null values in the City Dataframe'.format(city_data['aver
print('There are {} Null values in the City Dataframe'.format(global_data['av
```

```
There are 4 Null values in the City Dataframe
There are 0 Null values in the City Dataframe
```

As there are just 4 Null Values on the cities Dataframe, we'll choose to drop them

In [ ]:
```
# Droping the NAs on the dataframe
city_data.dropna(axis=0, inplace = True)
```

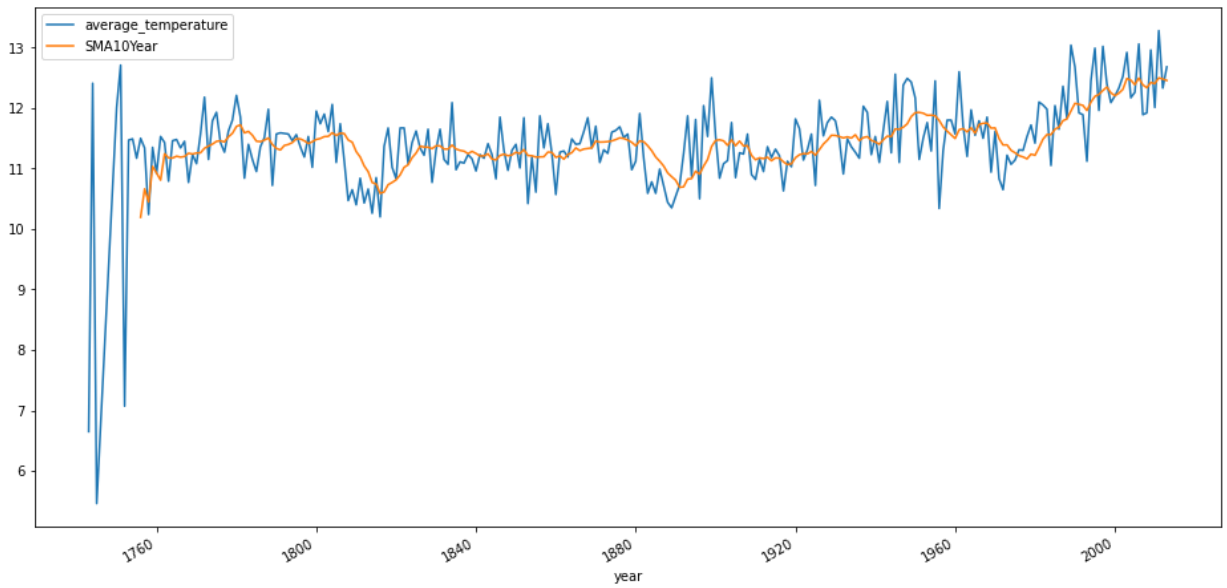
Now, Let's create our moving Average values. I've chosen a decade because it seemed like a logical value to consider (a decade) for temperature changes.

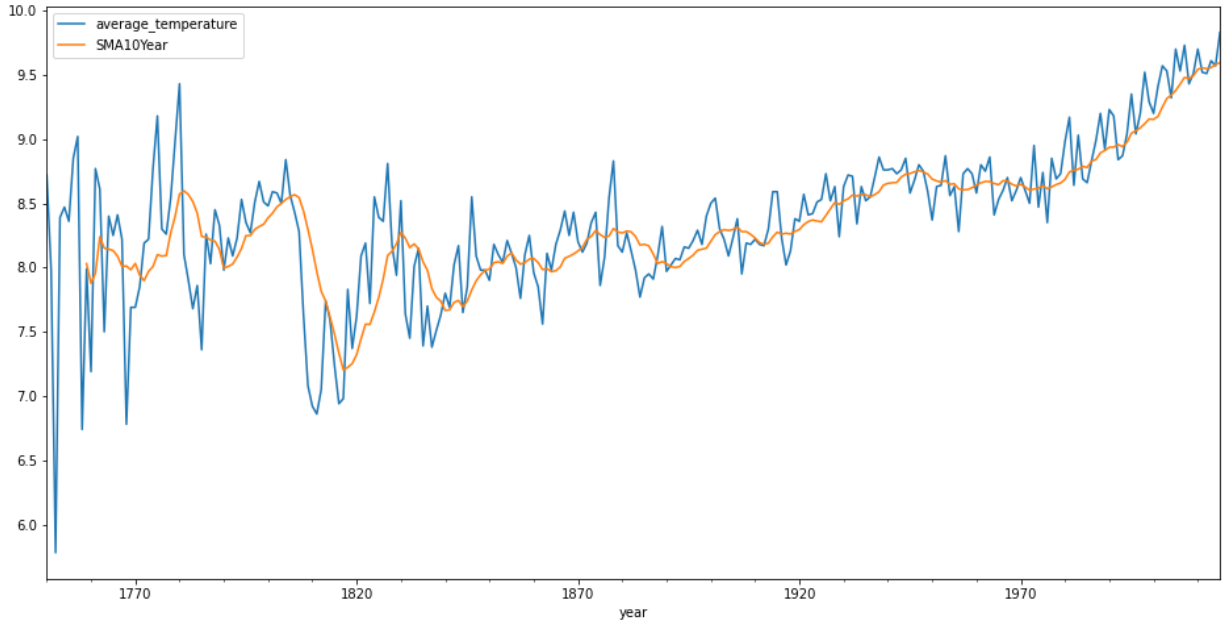
In [ ]:
```
# Let's do a 10 year moving average
city_data = city_data['average_temperature'].to_frame()
global_data = global_data['average_temperature'].to_frame()

city_data['SMA10Year'] = city_data['average_temperature'].rolling(10).mean()
global_data['SMA10Year'] = global_data['average_temperature'].rolling(10).mea
```

Now, let's compare the effect that our Moving Average had in our data

In [ ]:
```
# Comparing Moving Average to Granular data
city_data[['average_temperature', 'SMA10Year']].plot(label = 'Temperature', f
global_data[['average_temperature', 'SMA10Year']].plot(label = 'Temperature',
```

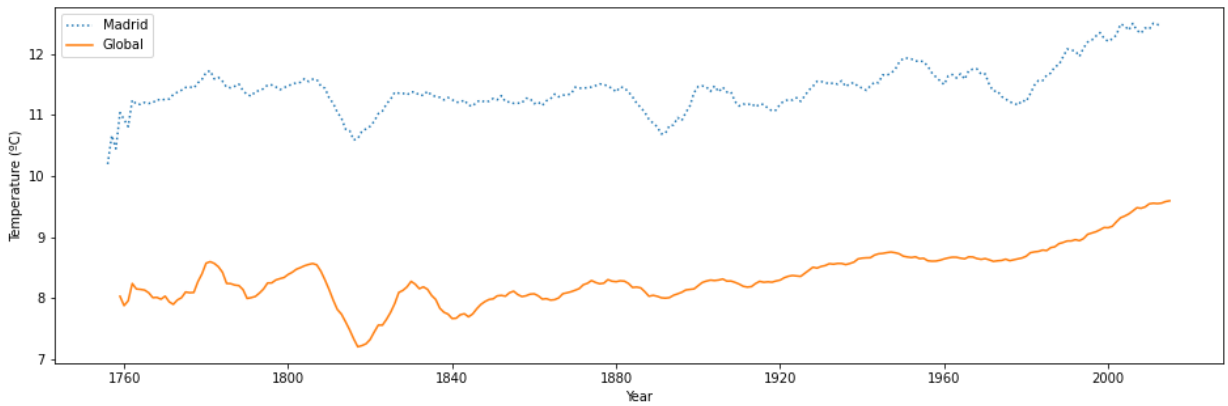
Out[ ]:
```
<AxesSubplot:xlabel='year'>
```





Finally, let's plot our Moving Average data for Madrid and for the Global Temperature.

In [ ]:
```
plt.plot(city_data['SMA10Year'], label = 'Madrid', linestyle = ':')
plt.plot(global_data['SMA10Year'], label = 'Global', linestyle = '-')
plt.legend()
plt.xlabel('Year')
plt.ylabel('Temperature (ºC)')
plt.show()
```



Conclusions:

- Madrid is by mean, 4 degrees hotter than the global average, which makes sense since Madrid is located relatively near to the Equator.
- We can see the same trends on both graphs, for example the drop in temperatures around 1820.
- We can observe that in the last years (2000s forward), both Madrid and Global Temperatures have suffered a constant hike. Madrid mean temperature has risen more sharply than the global average.
- We can see that Madrid suffered two drops in average temperature aroung 1890 and 1980 that seem to be milder in the global average. This could be due to local phenomena such as "El niño".

EXTRAS

Let's calculate the Correlation Coefficient between the two series

1. First of all, let's add the columns into a single dataframe:

In [ ]:
```
# Setup of the new dataframe
total_dataframe = city_data['SMA10Year'].to_frame()
total_dataframe.rename(columns={'SMA10Year':'Madrid'}, inplace= True)
total_dataframe['Global'] = global_data['SMA10Year']

# We drop the Na values created so that it's not a problem when we calculate

total_dataframe.dropna(inplace=True)

# Now We calculate the correlation coefficient

total_dataframe.corr(method= 'pearson')
```

Out[ ]:

	Madrid	Global
Madrid	1.000000	0.879192
Global	0.879192	1.000000

We observe that we have a really high correlation coefficient: 0.879

Estimation of Madrid's temperature Based on the Global Temperature

Madrid's mean temperature:

In [ ]:
```
print('The Average temperature in Madrid is:  {}ºC'.format(total_dataframe['M
```

```
The Average temperature in Madrid is:  11.44ºC
```

Global Mean Temperature:

In [ ]:
```
print('The Average Global temperature is:  {}ºC'.format(total_dataframe['Glob
```

```
The Average Global temperature is:  8.34ºC
```

In the first Part I said that Madrid was around '4ºC' Hotter in average, but it's actually 3.1ºC hotter (Not a bad approximation)