

Project 1: Exploring Weather Trends

This is the first project of the Data Analyst Udacity Nanodegree program. In this project, we will analyze local and global temperature data and compare the temperature trends where you live to overall global temperature trends.

Project Instructions

Your goal will be to create a visualization and prepare a write up describing the similarities and differences between global temperature trends and temperature trends in the closest big city to where you live. To do this, you'll follow the steps below:

- Extract the data from the database. There's a workspace in the previous section that is connected to a database. You'll need to export the temperature data for the world as well as for the closest big city to where you live. You can find a list of cities and countries in the `city_list` table. To interact with the database, you'll need to write a SQL query.
- Write a SQL query to extract the city level data. Export to CSV. Write a SQL query to extract the global data. Export to CSV. Open up the CSV in whatever tool you feel most comfortable using. We suggest using Excel or Google sheets, but you are welcome to use another tool, such as Python or R.
- Create a line chart that compares your city's temperatures with the global temperatures. Make sure to plot the moving average rather than the yearly averages in order to smooth out the lines, making trends more observable (the last concept in the previous lesson goes over how to do this in a spreadsheet).
- Make observations about the similarities and differences between the world averages and your city's averages, as well as overall trends. Here are some questions to get you started.
 - 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?

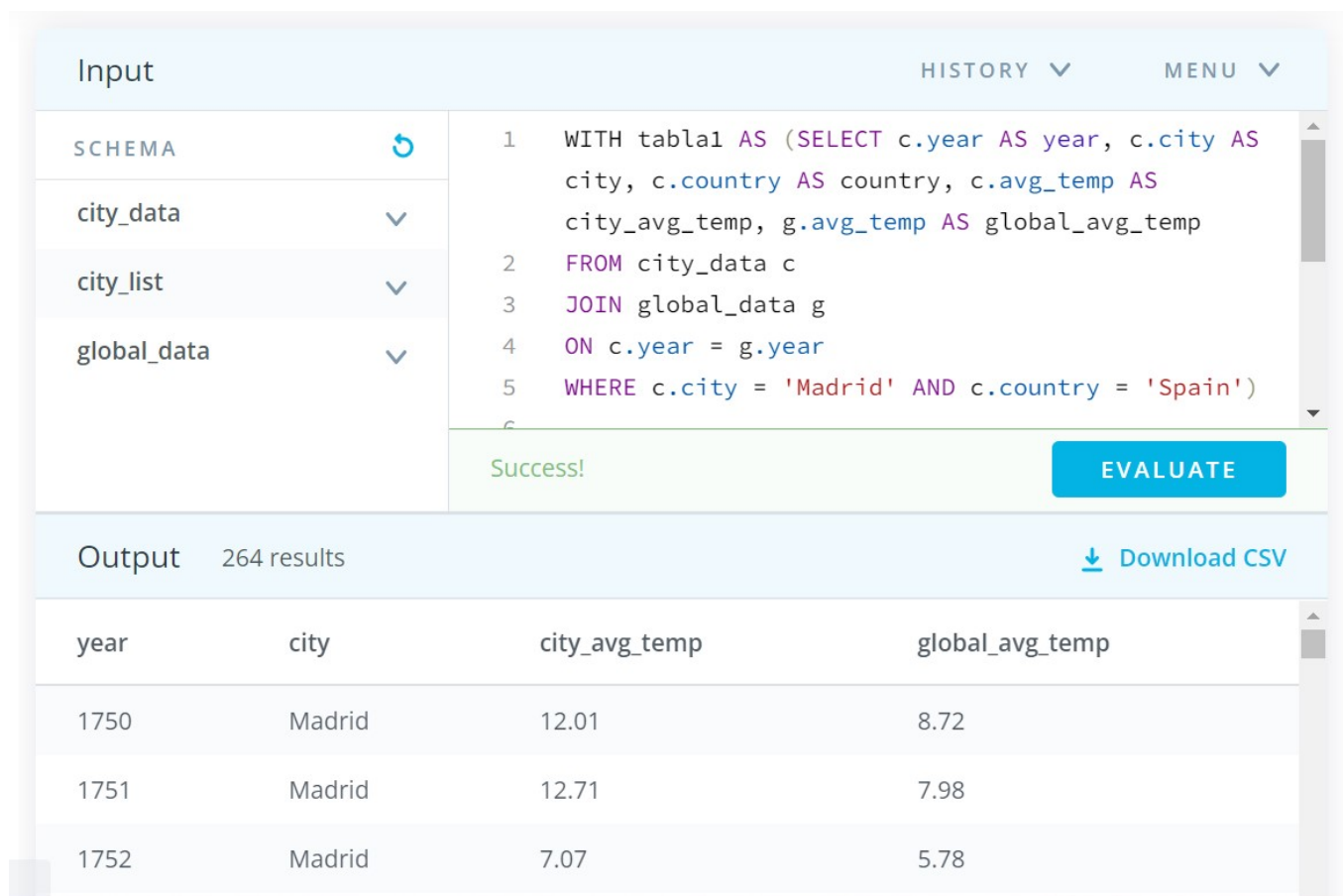
Extract the data from the database

The SQL query to get the data from the database was:

```
WITH tabl1 AS (SELECT c.year AS year, c.city AS city,
c.country AS country, c.avg_temp AS city_avg_temp,
g.avg_temp AS global_avg_temp
FROM city_data c
JOIN global_data g
ON c.year = g.year
WHERE c.city = 'Madrid' AND c.country = 'Spain')
SELECT year, city, city_avg_temp, global_avg_temp
FROM tabl1
ORDER BY 1
```

Although I live in Vigo, Spain, the closest city to Vigo in the database is Madrid, so I used Madrid as the city in Spain to do my analysis and compare it to the global average temperature.

Just below you can see a screenshot of the query on the Udacity platform.



The screenshot shows the Udacity SQL interface. On the left, under the 'Input' tab, there is a 'SCHEMA' section with a refresh icon and a list of tables: 'city_data', 'city_list', and 'global_data', each with a dropdown arrow. The main area displays a SQL query with line numbers 1 through 6. Below the query, a green 'Success!' message is shown next to a blue 'EVALUATE' button. The 'Output' section at the bottom shows '264 results' and a 'Download CSV' link. A table of results is displayed with four columns: 'year', 'city', 'city_avg_temp', and 'global_avg_temp'. The first three rows of data are visible.

year	city	city_avg_temp	global_avg_temp
1750	Madrid	12.01	8.72
1751	Madrid	12.71	7.98
1752	Madrid	7.07	5.78

```
In [1]: # Import the package
from google.colab import files
import pandas as pd

# To plot the data, we are going to use plotnine package to use ggplot
insted of matplotlib.
# The ggplot package was created in R language.
from plotnine import ggplot, aes, geom_line, geom_point, ggtitle, labs,
stat_smooth, theme, geom_hline
import warnings

warnings.filterwarnings('ignore')
```

```
In [2]: # Because I used Jupyter in google colab I had to upload the csv file f
rom my computer
uploaded = files.upload()
```

Examinar... No se han seleccionado archivos.

Upload widget is only available when the cell has been executed in the current browser session.
Please rerun this cell to enable.

Saving madrid_and_global_temp.csv to madrid_and_global_temp.csv

Read the csv file

The SQL query was saved as madrid_and_global_temp.csv so we will open the file with read_csv pandas method.

```
In [3]: data = pd.read_csv('madrid_and_global_temp.csv')
data.head()
```

Out[3]:

	year	city	city_avg_temp	global_avg_temp
0	1750	Madrid	12.01	8.72
1	1751	Madrid	12.71	7.98
2	1752	Madrid	7.07	5.78
3	1753	Madrid	11.47	8.39
4	1754	Madrid	11.49	8.47

Explore the data

First, we are going to explore the data to see if it is necessary to clean it and see some basic characteristics of the data.

```
In [5]: # Let's look at the dimension of the data frame
data.shape
```

```
Out[5]: (264, 4)
```

We have 264 rows and 6 columns in the data frame

Let's check if there are NA in the data to clean the data

```
In [6]: data.isna().sum()
```

```
Out[6]: year                0
city                0
city_avg_temp        0
global_avg_temp      0
dtype: int64
```

There are no NAs in the data frame.

Let's Look at some basic statistics

```
In [7]: data.describe()
```

```
Out[7]:
```

	year	city_avg_temp	global_avg_temp
count	264.000000	264.000000	264.000000
mean	1881.500000	11.450682	8.359394
std	76.354437	0.630496	0.575184
min	1750.000000	7.070000	5.780000
25%	1815.750000	11.117500	8.077500
50%	1881.500000	11.415000	8.365000
75%	1947.250000	11.800000	8.700000
max	2013.000000	13.280000	9.730000

The mean of the average temperature of Madrid (11.45 °C) is higher than the global one (8.36 °C) as well as the standard deviation, so Madrid is warmer on average than the global one. The minimum value in the entire series of the average temperature of Madrid is 7.07 degrees centigrade; which represents a decrease of more than 38% with respect to the average. We can see a similar big difference between the average and the minimum temperature in global_avg_temp, this could be an abnormally cold year (outlier) for some climatic reason.

Let's see in what year these low temperatures occurred.

```
In [8]: data[data['city_avg_temp']== 7.070000]
```

```
Out[8]:
```

	year	city	city_avg_temp	global_avg_temp
2	1752	Madrid	7.07	5.78

We see that the minimum temperature in Madrid and in the global temperature took place in the same year (1752) so it could be an abnormally cold year.

```
In [9]: # Let's rename the columns in the data frame

data = data.rename(columns={'city_avg_temp': 'madrid_avg_temperature',
                             'global_avg_temp': 'global_avg_temperature'})
```

```
In [10]: data.head()
```

```
Out[10]:
```

	year	city	madrid_avg_temperature	global_avg_temperature
0	1750	Madrid	12.01	8.72
1	1751	Madrid	12.71	7.98
2	1752	Madrid	7.07	5.78
3	1753	Madrid	11.47	8.39
4	1754	Madrid	11.49	8.47

```
In [11]: # We are going to create two news columns to capture the differences
# among the actual year and the previous one.

data['diff_madrid_temp']= data.madrid_avg_temperature.diff()
data['diff_global_temp']= data.global_avg_temperature.diff()
data.head()
```

Out[11]:

	year	city	madrid_avg_temperature	global_avg_temperature	diff_madrid_temp	diff_global_temp
0	1750	Madrid	12.01	8.72	NaN	
1	1751	Madrid	12.71	7.98	0.70	
2	1752	Madrid	7.07	5.78	-5.64	
3	1753	Madrid	11.47	8.39	4.40	
4	1754	Madrid	11.49	8.47	0.02	

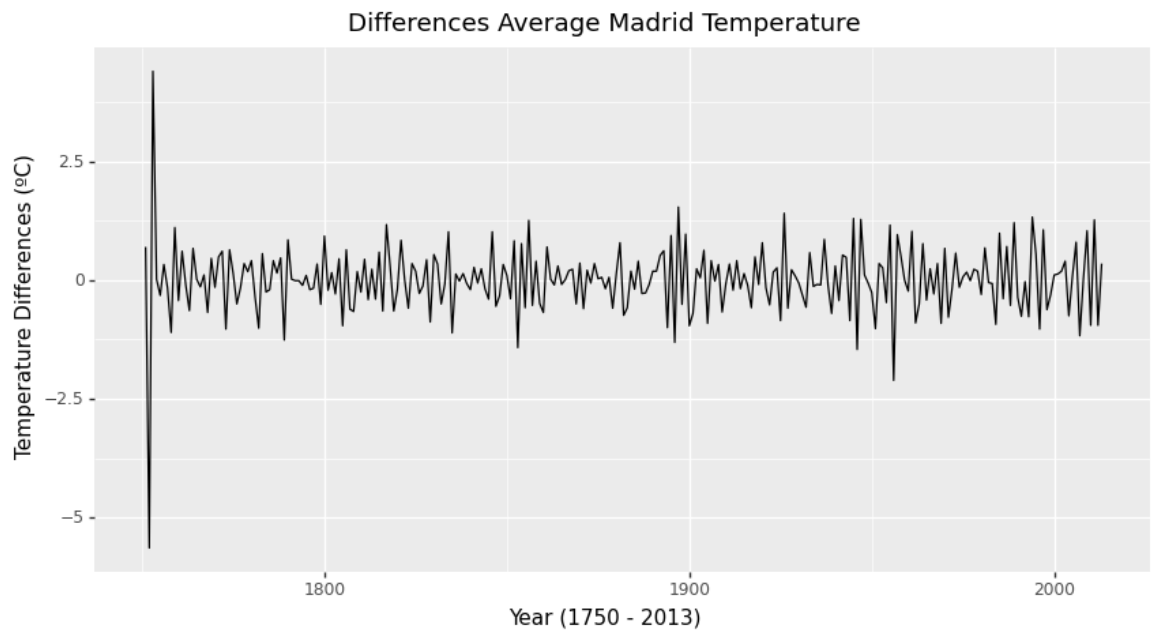
Make a plot

The idea is to see the differences in temperatures among years in Madrid and in global average temperature.

```
In [12]: # This function makes a line plot with the year in the x-axis
# and temperature or any variable in the y-axis.
def diff_plot(data, title, temp_variable, y= "Temperature Differences
(°C)"):
    '''
    This function make a plot using two arguments
    arg:
        data (The dataframe that you want to use
        title (principal title of the plot)
        Temp_variable (Chose one temperature variable in dataframe)
        y axis label, It should be a string
    '''
    print(ggplot(data, aes(x="year", y=temp_variable))
          + ggtitle(title)
          + labs (y= y, x = "Year (1750 - 2013)")
          + geom_line()
          + theme(figure_size=(10, 5)))
```

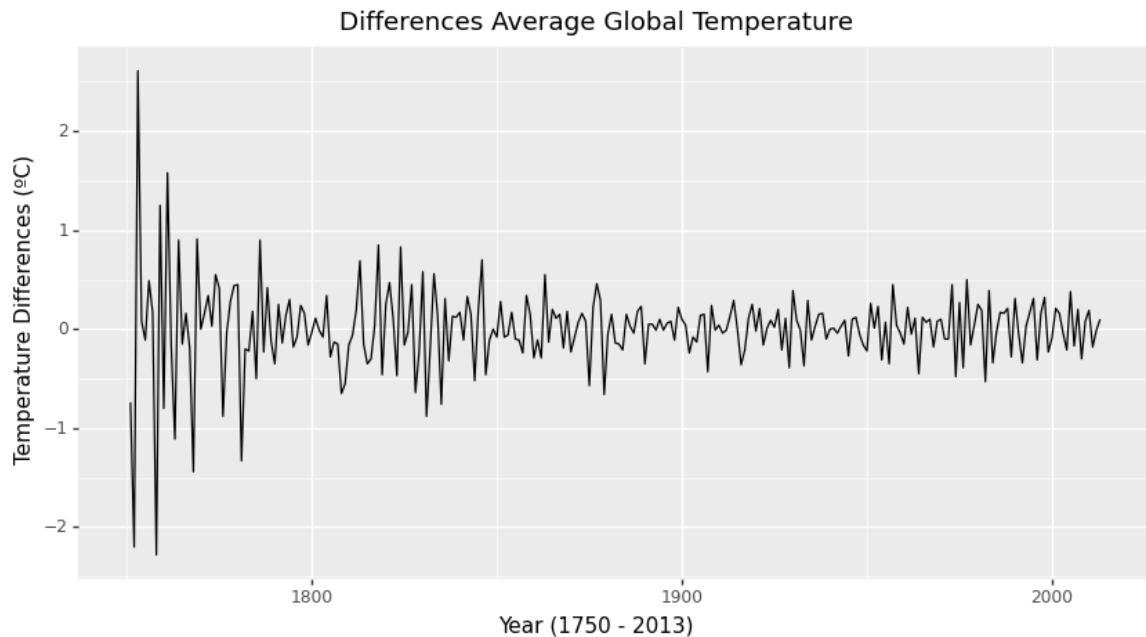
```
In [ ]: # Let's make the plot to see the differences in the average Madrid temperature
# among one year and the previous one.

diff_plot(data, 'Differences Average Madrid Temperature', 'diff_madrid_temperature')
```



<ggplot: (8744591894661)>

```
In [13]: # Let's make the plot to see the differences in the average global temperature
# among one year and the previous one.
diff_plot(data, 'Differences Average Global Temperature', 'diff_global_temp')
```



<ggplot: (8754949424689)>

Plotting the annual temperature differences, we see that the differences change from positive to negative around zero over the years. In the case of Madrid, volatility is more constant over time. An obvious curiosity is that more volatility is observed in global temperature differences in the first hundred years and then it has less volatility and remains similar. That's perhaps because the global data has increased in the number of countries over time, but we can't be sure about that. To try to see this, we are going to do another SQL query on the Udacity platform.

Extract more data from the dataBase

The idea is to make a query to get from the city_data how the numbers of countries increase or not their number over time.

The SQL query to get the data from the database was:

```
SELECT year,COUNT(DISTINCT country) AS number_countries
FROM city_data
GROUP BY year
ORDER BY year
```

And saved the query as number_countries.csv


```
In [14]: # Load the file
uploaded = files.upload()
```

Examinar... No se han seleccionado archivos.

Upload widget is only available when the cell has been executed in the current browser session.
Please rerun this cell to enable.

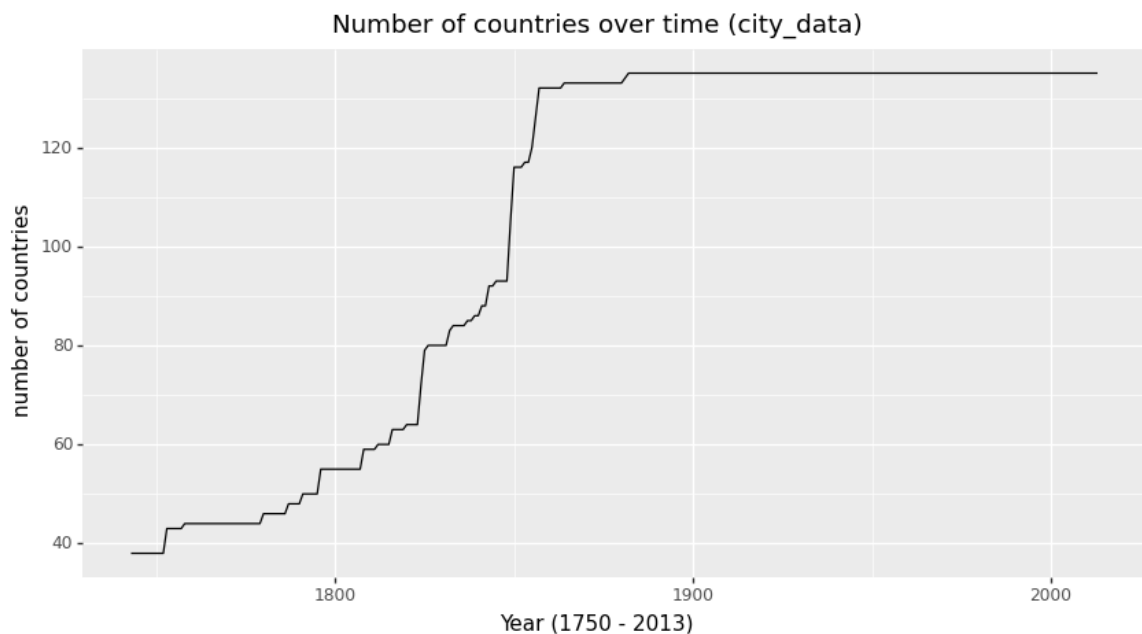
Saving number_countries.csv to number_countries.csv

```
In [15]: data_countries = pd.read_csv('number_countries.csv')
data_countries.head()
```

Out[15]:

	year	number_countries
0	1743	38
1	1744	38
2	1745	38
3	1746	38
4	1747	38

```
In [16]: # This plot shows the variation in the numbers of countries that the ci
ty_data has in the database.
diff_plot(data_countries,title='Number of countries over time (city_dat
a)', temp_variable = 'number_countries', y = 'number of countries')
```



<ggplot: (8754949415853)>

We see how the number of countries in city_data has been increasing in the first hundred years and that can cause great volatility in the differences in the global average temperature in that period, but we really do not know how many countries have been considered in the global_data (database). The number of countries in city_data remains constant from 1864 to 2013.

Check if there is a tendency in the differences between Madrid and Global temperature.

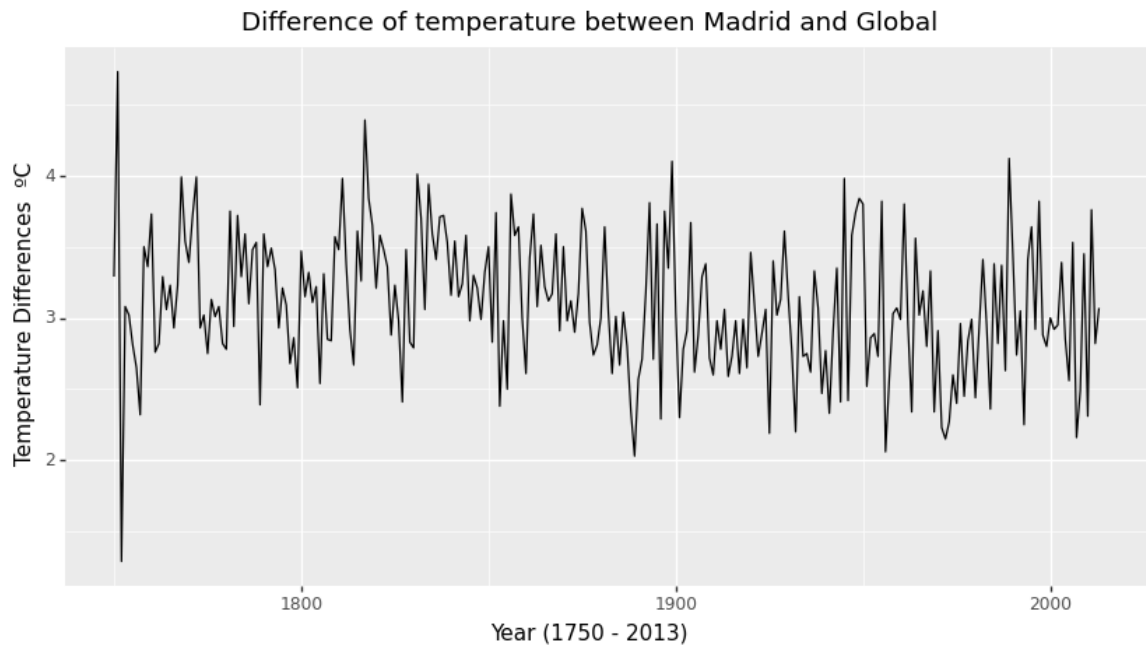
The difference between the Madrid mean temperature (11.45 °C) and the global mean (8.36 °C) in all data frame is equal to 3.09. That idea is to make a plot a see how the differences change over time.

```
In [17]: # Create a new column (diff_madrid_global) is the difference among madr
id_avg_temperature and global_avg temperature
data['diff_madrid_global'] = data['madrid_avg_temperature'] - data['glob
al_avg_temperature']
data.head()
```

Out[17]:

	year	city	madrid_avg_temperature	global_avg_temperature	diff_madrid_temp	diff_globe
0	1750	Madrid	12.01	8.72		NaN
1	1751	Madrid	12.71	7.98		0.70
2	1752	Madrid	7.07	5.78		-5.64
3	1753	Madrid	11.47	8.39		4.40
4	1754	Madrid	11.49	8.47		0.02

```
In [19]: # Plot diff_madrid_global
diff_plot(data, title='Difference of temperature between Madrid and Global', temp_variable = 'diff_madrid_global', y = 'Temperature Differences °C')
```



```
<ggplot: (8754949419153)>
```

The temperature differences between Madrid and the global temperature do not show a clear tendency to increase or decrease over time. We see how it increases and decreases around the mean of differences (3.09 °C).

We are going to reshape the data

The idea is to have a column called legend where we have two factors (for example, Madrid avg temperature, global avg temperature) to capture the two together in one plot.

```
In [20]: # This function reshape the data frame data changing the columns name to factors
# in a new column called Leyend to later plot in one graph.

def reshape_data(variable1, variable2):
    """
    arg:
        variable1 and variable2 ( change columns name for a factors in
        Leyend colums)
    """

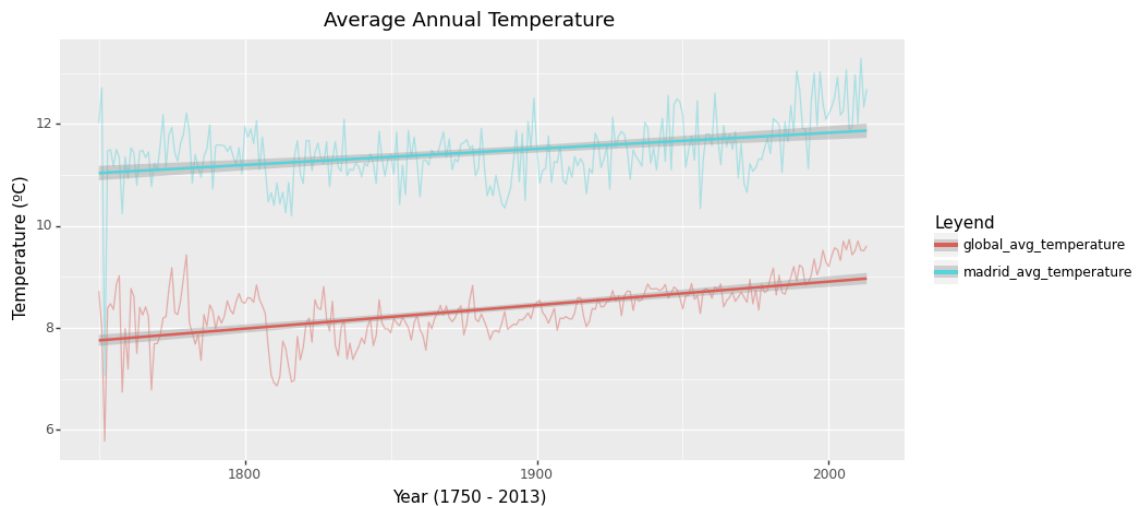
    data_reshape = pd.melt(data, id_vars='year',
                           value_vars=[variable1, variable2],
                           var_name='Leyend',
                           value_name='temperature')
    return data_reshape

data_reshape = reshape_data('madrid_avg_temperature', 'global_avg_temperature')
print(data_reshape.head())
print(data_reshape.tail())
```

	year	Leyend	temperature
0	1750	madrid_avg_temperature	12.01
1	1751	madrid_avg_temperature	12.71
2	1752	madrid_avg_temperature	7.07
3	1753	madrid_avg_temperature	11.47
4	1754	madrid_avg_temperature	11.49
	year	Leyend	temperature
523	2009	global_avg_temperature	9.51
524	2010	global_avg_temperature	9.70
525	2011	global_avg_temperature	9.52
526	2012	global_avg_temperature	9.51
527	2013	global_avg_temperature	9.61

We are going to plot all the points of the annual average temperature of Madrid and the global world, together in the same graph with a regression line.

```
In [21]: (ggplot(data_reshape, aes(x="year", y="temperature", colour = 'Legend'))
+ ggtitle('Average Annual Temperature')
+ labs (y="Temperature (°C)", x = "Year (1750 - 2013)")
+ geom_line(alpha = 0.45)+ stat_smooth(method = "lm")
+ theme (figure_size=(10, 5)))
```



```
Out[21]: <ggplot: (8754946361093)>
```

We can see an increase in the average temperature trend over time, both in the city of Madrid as well as in the global average temperature.

Moving average

- Using the moving average smoothing the data and makes it easier to see the trend in the data.
- We will calculate the moving average with a window of 10 years.

```
In [22]: data['madrid_moving_avg'] =data.madrid_avg_temperature.rolling(10).mean()
data['global_moving_avg'] =data.global_avg_temperature.rolling(10).mean()
data.head()
```

```
Out[22]:
```

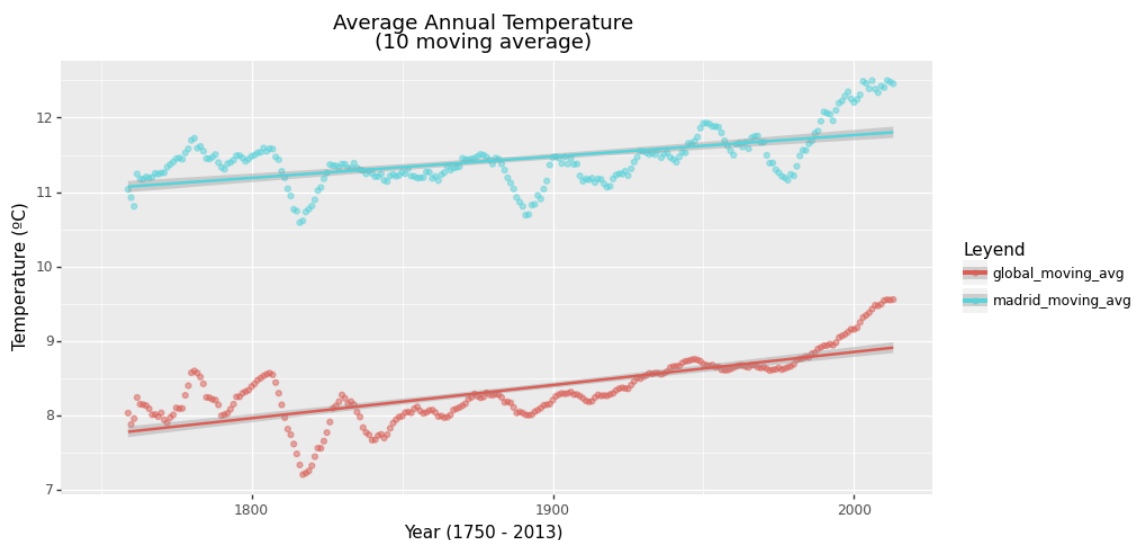
	year	city	madrid_avg_temperature	global_avg_temperature	diff_madrid_temp	diff_global
0	1750	Madrid	12.01	8.72	NaN	
1	1751	Madrid	12.71	7.98	0.70	
2	1752	Madrid	7.07	5.78	-5.64	
3	1753	Madrid	11.47	8.39	4.40	
4	1754	Madrid	11.49	8.47	0.02	

```
In [23]: # We are using the moving average values to make a new plot and before
          # that, we reshape the data changing the moving average
          # variable as a factor in a Legend column.
          data_reshape = reshape_data('madrid_moving_avg', 'global_moving_avg')
          print(data_reshape.head())
          print(data_reshape.tail())
```

	year	Legend	temperature
0	1750	madrid_moving_avg	NaN
1	1751	madrid_moving_avg	NaN
2	1752	madrid_moving_avg	NaN
3	1753	madrid_moving_avg	NaN
4	1754	madrid_moving_avg	NaN

	year	Legend	temperature
523	2009	global_moving_avg	9.493
524	2010	global_moving_avg	9.543
525	2011	global_moving_avg	9.554
526	2012	global_moving_avg	9.548
527	2013	global_moving_avg	9.556

```
In [24]: (ggplot(data_reshape, aes(x="year", y="temperature", colour = 'Legend'
          '))
          + ggtitle('Average Annual Temperature\n(10 moving average)')
          + labs (y="Temperature (°C)", x = "Year (1750 - 2013)")
          + geom_point(alpha = 0.50)+ stat_smooth(method = "lm")
          + theme(figure_size=(10, 5)))
```



```
Out[24]: <ggplot: (8754948347245)>
```

Now the plot shows a more clear increase in the temperature average tendency in Madrid as well in the global world. The slope of the regression line is higher in the global world than in Madrid.

```
In [25]: # Correlation between Madrid and global temperatures
data[['madrid_avg_temperature', 'global_avg_temperature']].corr(method=
'pearson')
```

Out[25]:

	madrid_avg_temperature	global_avg_temperature
madrid_avg_temperature	1.000000	0.684226
global_avg_temperature	0.684226	1.000000

The correlation coefficient is 0.68 (moderately strong), maybe is not possible to estimate with enough precision Madrid temperature based on global average temperature.

Conclusion

- The average temperature of Madrid is 3.09 degrees Celsius higher than the global average temperature.
- In 1752 the lowest temperatures were recorded both in Madrid (7.07 °C) and in the global temperature (5.78) with a difference of 38% about the mean of all the data.
- Between 2010 and 2013 the highest temperatures of the series were recorded both in Madrid (12.49 °C) and in the global temperature (9.56).
- The temperature differences between Madrid and the global temperature vary above and below the mean difference (3.09 °C) over time, but in a somewhat random way.
- There is greater volatility in the global temperature in the period between 1750 and 1850, while the volatility of the temperature of Madrid remains more constant throughout the entire series. There is no conclusive explanation for this, perhaps it is because in that period the global temperature database included fewer countries, as the "Number of Countries Over Time" graph shows, or perhaps the Mediterranean climate of Madrid was seen less affected by those fluctuations.
- charts made with a moving average are very similar. There is a period between 1810 and 1820 that the average temperatures suffer a decrease both in Madrid and worldwide, that period coincides with the climatic period in which Europe and Asia cooled down.
- We see a peak around 1830 and this coincides with the industrial revolution, closely linked to the extraction and use of coal.
- There is an overall uptrend in the global temperature and Madrid temperature, indicating global warming.
- Since 1900 the global temperature has been increasing and we see how since the 1970s the slope of the temperature curve both in Madrid and in the global temperature has increased markedly, which indicates that global warming is accelerating.

In []: