Explore weather trends.

Introduction.

This project is my implementation of the *"Explore weather trends"* project which is project 0 in the Udacity DataAnalyst Nanodegree program. Here is a link to the program. As I live in the Netherlands I will compare the global temperature data with that of a city in the Netherlands.

Step 1. Understanding the schema.

Before retrieving the data it is important to understand the schema of the database. It was given that the database consisted of three tables city_list, city_data, and global_data. The schema was explored in the following manner. Command:

```
SELECT * FROM city_list LIMIT 1;
```

Result:

city	country
Abidjan	Côte D'Ivoire

Command:

```
SELECT * FROM city_data LIMIT 1;
```

Result:

year	city	country	avg_temp
1849	Abidjan	Côte D'Ivoire	25.58

Command:

```
SELECT * FROM global_data LIMIT 1;
```

Result:

year	avg_temp
1750	8.72

Step 2. Retrieving the data.

Selecting a city.

Now it is time to select a city from the Netherlands. First I wanted to know which cities from the Netherlands were available in the database. Command:

```
SELECT * FROM city_list WHERE country = 'Netherlands';
```

Result:

city	country
Amsterdam	Netherlands

Hmm for obvious reasons I choose Amsterdam.

Getting the starting point.

To find out the first year from which global data is available I used the following command.

```
SELECT * FROM global_data ORDER BY year LIMIT 1;
```

Result:

year	avg_temp
1750	8.72

This is the same result as I got from investigating the global_data schema above, so apparently the data was already ordered by year.

Now for Amsterdam

```
SELECT * FROM city_data
WHERE city = 'Amsterdam'
ORDER BY year LIMIT 1;
```

Result:

year	city	country	avg_temp
1743	Amsterdam	Netherlands	7.43

The results above mean that I can compare Amsterdam with global data from the year 1750.

Using an inner join statement

The data that is needed to compare the temperature in Amsterdam with the global temperature is in two tables. Therefore an inner join is needed. In order to reduce the waiting time as wel as unnecessary load on the servers I use a LIMIT of 3 while developing the query. From city_data we want the year and the average temperature (avg_temp) for Amsterdam and only for 1750 and after.

This data must be sorted by year.

```
SELECT year, avg_temp
FROM city_data
WHERE city = 'Amsterdam' AND year >= 1750
ORDER BY year
LIMIT 3;
```

Result

year	avg_temp
1750	10.04
1751	9.63
1752	5.97

Great. Before adding the inner join it is important to realize that both the city_data as the global_data table have year and avg_temp colums and SQL commands need to be unambiguous. So the column names need to be preceded by the table name. Command:

```
SELECT city_data.year, city_data.avg_temp
FROM city_data
WHERE city = 'Amsterdam' AND city_data.year >= 1750
ORDER BY year
LIMIT 3;
```

The result is the same as the query above. Now we are ready to make the inner join.

```
SELECT city_data.year, city_data.avg_temp, global_data.avg_temp
FROM city_data
INNER JOIN global_data ON city_data.year=global_data.year
WHERE city = 'Amsterdam' AND city_data.year >= 1750
ORDER BY year LIMIT 3;
```

Result:

year	avg_temp
1750	8.72
1751	7.98
1752	5.78

Getting there. To seperate the avg_temp data from city_data and global_data both columns will be renamed with the keyword AS. Also the "LIMIT 3" can now be removed. Command:

```
SELECT city_data.year,
city_data.avg_temp AS amsterdam_avg_temp,
global_data.avg_temp AS global_avg_temp
```

```
FROM city_data
INNER JOIN global_data ON city_data.year=global_data.year
WHERE city = 'Amsterdam' AND city_data.year >= 1750
ORDER BY year;
```

The result is exported in a csv file named raw_data.csv, which is included in the project.

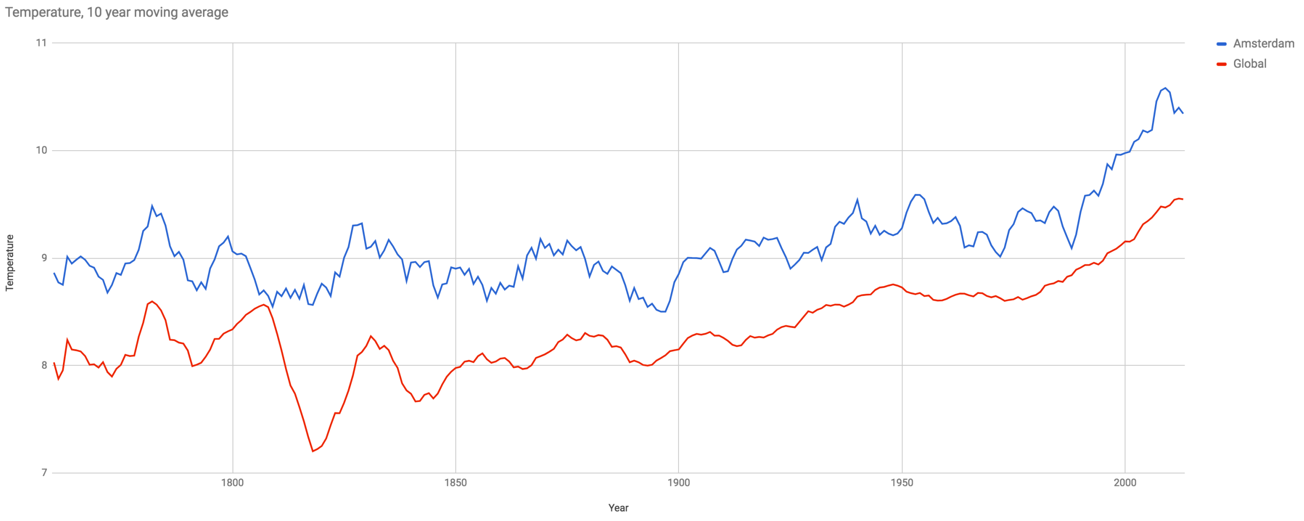
Step 3. Visualize the data

Moving average.

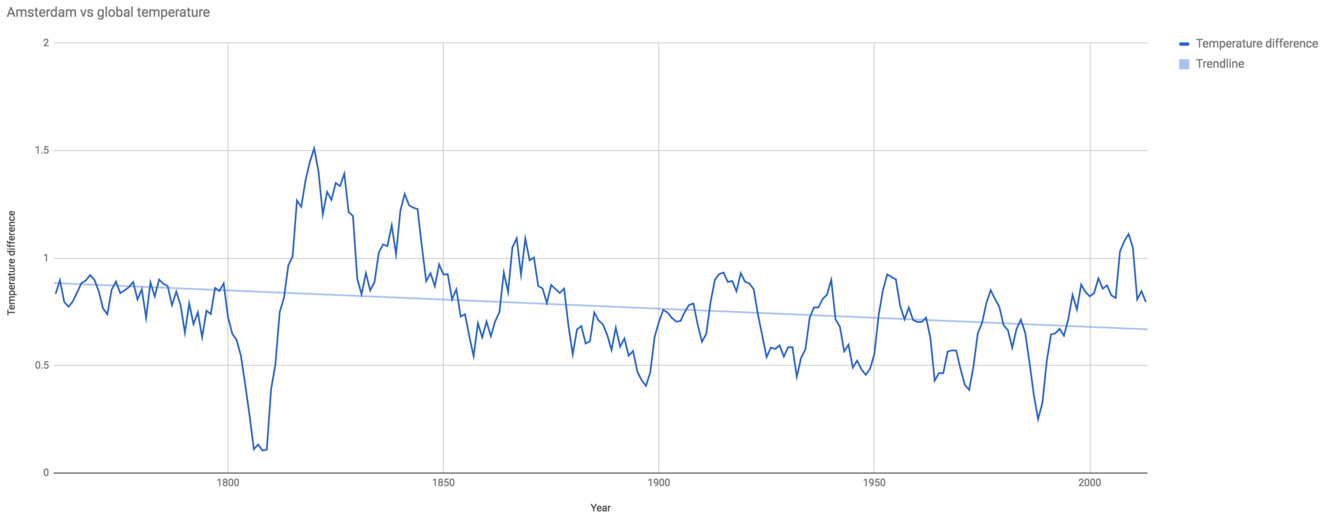
The raw_data.csv file has been imported into a google sheet. The link is sheet. I also exported the file to excel. This excel file is included in the project. The year is in column A, the temperature of Amsterdam is in column B, and the global temperature is in column C. In columns D and E I calculated the moving averages of columns B and C respectively. Simply by inserting the formula =AVERAGE(B2:B11) into field D11 and then copy downard. In column F I substracted column E from column D to get the temperature difference of Amsterdam - Global.

Plot.

Two plots were made. The first plot shows the 10 year moving average of the global temperatures as wel as the temperatures in Amsterdam, from 1760 to 2013.



The second plot show the temperature difference between Amsterdam and the rest of the world.



Step 4. Observations.

- The signal of Amsterdam is noicier than the global line. This is no surprice as the Amsterdam is just one place and the global line is the average of the whole world.
- The temperature in Amsterdam is consitently higher than the global temperature.
- The difference between the the temperature in Amsterdam and the global temperatures is decreasing (Figure 2)
- The global temperature have been rising from early 20th century.
- After a small plateau in the seventies the global temperature has been increasing at an even higher rate.
- The temperature in Amsterdam follow the global trends and is rising also.
- The last 20 years the difference between Amsterdam and global temperatures ranged from 0.64 to 1.11 with an average of 0.87. That is to big an error to reliably get the temperature in Amsterdam from the global temperature.