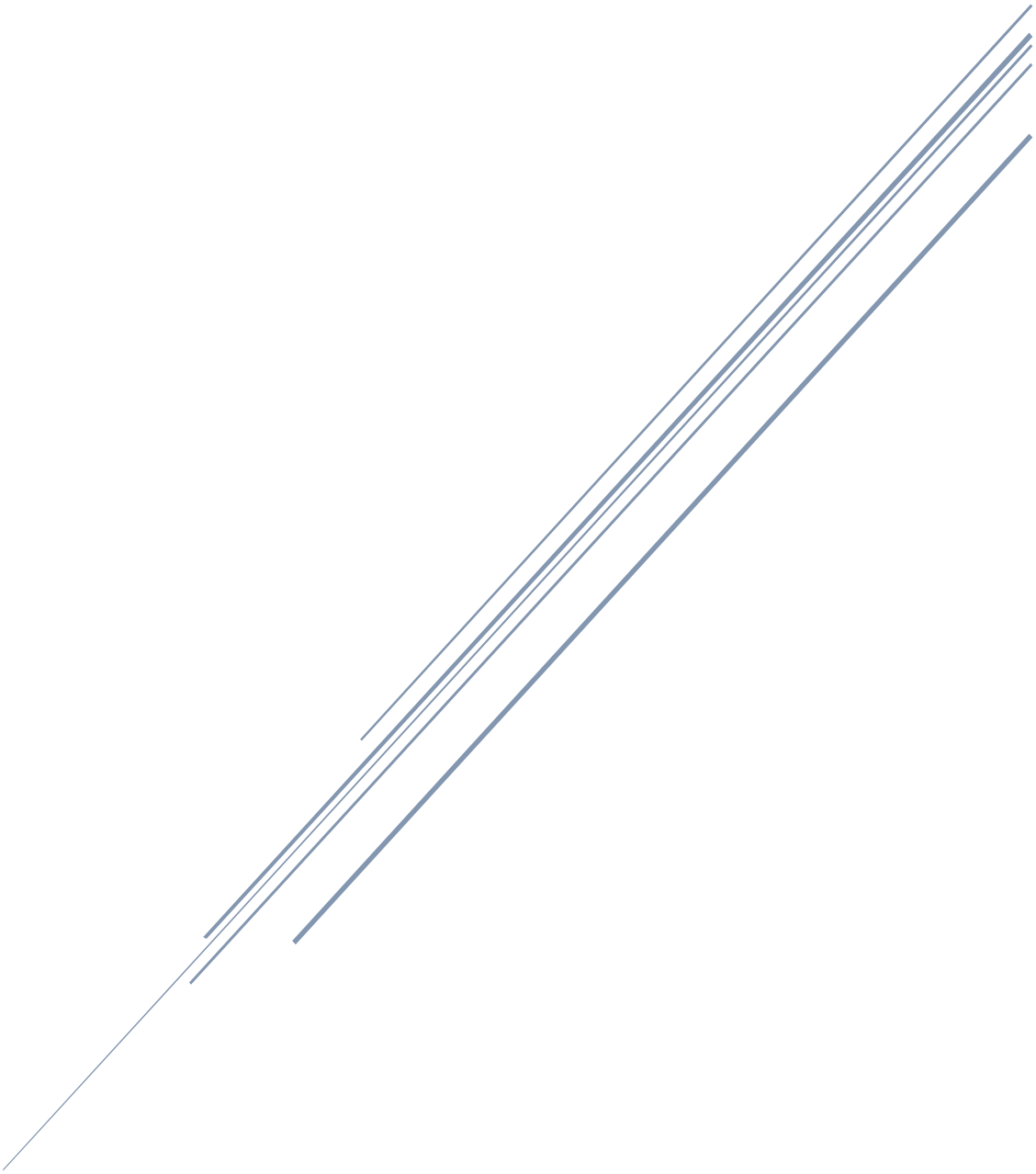
EXPLORING WEATHER TRENDS

Udacity - Nanodegree Data Analyst



Leonardo Simões

04/04/2020

Summary

1. Objective

This work is the first project of the Udacity Nanodegree Data Analyst program . The objective of this work is to carry out an analysis of global temperature data and in a large city close to where the author lives, obtaining the data from a database, organizing this data, processing the data, calculating moving averages, generating a visualization for these data and describe the similarities and differences between the trends of the two temperatures over a given period.

2. Tools and steps

The steps of this project were data extraction, calculation of moving averages, line graph plot and observations.

The query of the data was made using SQL. The reading of the CSV, calculations of moving averages and plot of the line graph were made using the Python language in a single file.

The SQL and python scripts are attached.

3. Data extraction

To extract the desired data , SQL was used in the workspace provided in section "3. Acessing Data With SQL "on the course page. Three consultations were held:

- Search the nearest big city of Nova Friburgo, where I live. Then the query is made in the city_list table restricting the cities that are from Brazil . As expected, the closest city was Rio de Janeiro, as it is the only one listed in the same state as my city and is relatively close.

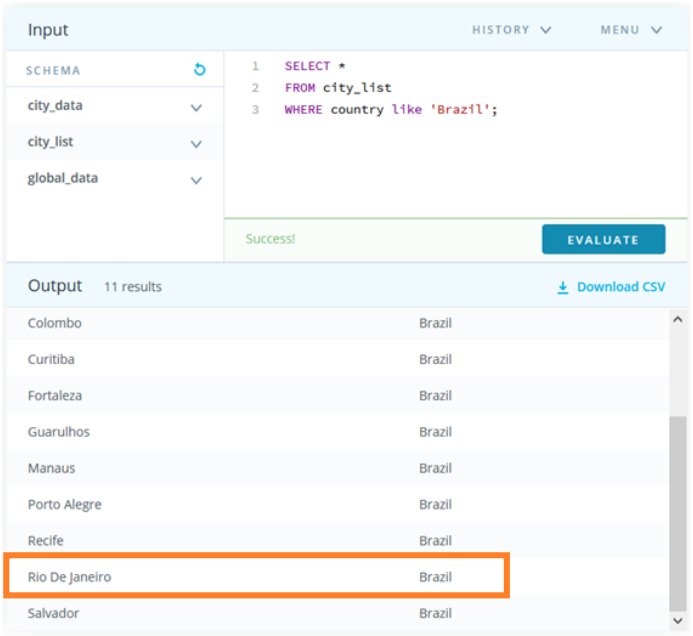


Figure 1 - Consultation of Brazilian cities

- Search the temperatures of the city of Rio de Janeiro in the last 110 years, as it will be used moving averages 10 years for the past 100 years. Then a query is made in the city_data table restricting the names of the city to be “Rio De Janeiro”, sorting by the most recent years, and limiting to the first 110 results.

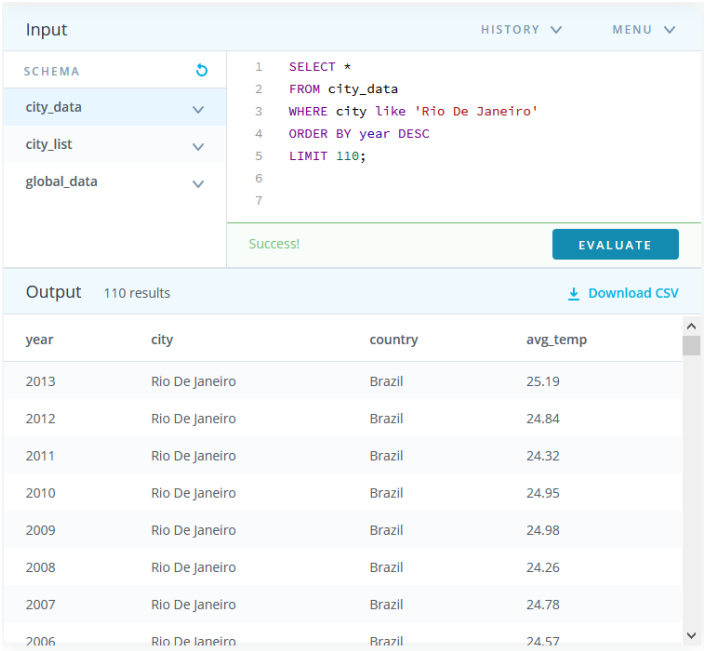


Figure 2 - Consultation of the last 100 temperatures in Rio de Janeiro

- Search for global temperatures over the past 110 years, as 10-year moving averages for the past 100 years will be used. Then a query is made in the global_data table, sorting by the most recent years, and limiting to the first 110 results.

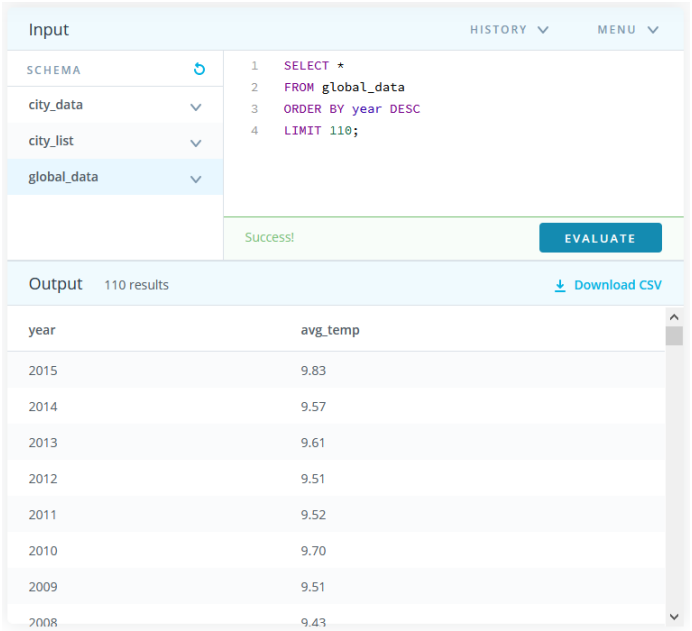


Figure 3 - Consultation of the last 100 global temperatures

4. Calculation of moving averages

The calculation of moving averages was done in Python code implemented in the function called “ calculateeMovingAverage10 ”. This function receives an array list , and iterates over it at indexes 0 to the size of array-10 (inclusive), and for each iteration it adds the average of the index values of that position up to 9 positions in front of another list array . In this case, size 10 (years) was chosen for the moving averages.

5. Line chart

The graphic plot was made in the same Python file implemented in the function named “ plotTemperaturesPerYears ” , which receives arrays list of years and moving averages of global and Rio de Janeiro temperatures . The graph is a line graph, whose x-axis is the years and the y-axis is the temperature.

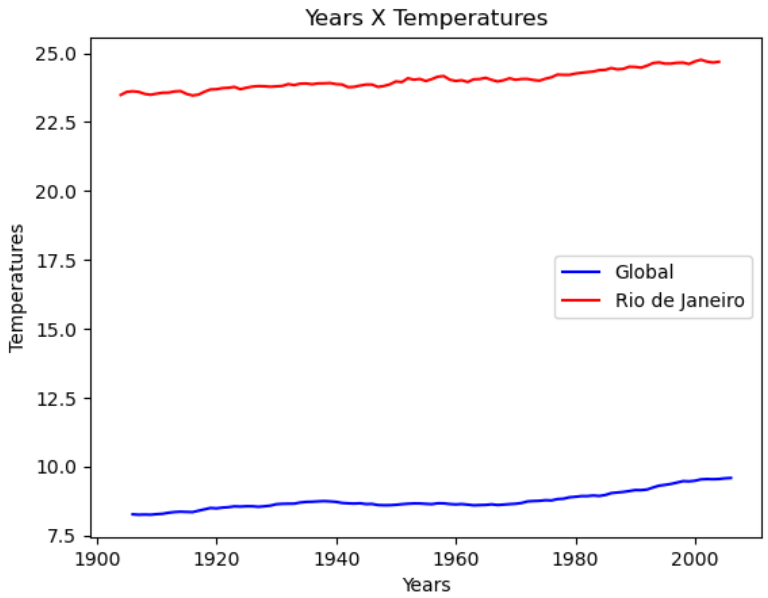


Figure 4 - Graph of global and Rio de Janeiro temperatures

6. Conclusions

From the graph plotted in the previous section, some conclusions about global and city temperatures in Rio de Janeiro over the past 100 years can be made:

- The general trend for both temperatures is to increase gradually over the years, and in the last 20 years they have increased at a higher rate.
- The city of Rio de Janeiro has much higher temperatures than the global ones during the whole time analyzed.
- The difference between the temperatures of the city of Rio de Janeiro and the global remains consistent until 20 years ago, when it decreased a little even with the increase of both.
- The trend is that global temperatures will not reach the temperature of the city of Rio de Janeiro even in the coming centuries, except that there is some extreme climatic phenomenon of global magnitude.

REFERENCES

[1] Exploring Weather Trends - Project Instructions, UDACITY . Available in <https://classroom.udacity.com/nanodegrees/nd002-ent/parts/978b95a2-2fa2-4263-a2a7-b0e0015ddb5e/modules/23fd1baf-2377-4fa8-87f1-052bd78922d6/lessons/d551938c-d004-4801-a269-4b8dd784cc3b/concepts/7792fbaa-7f3d-46f1-b6fb-e5f65ce35796>. Acessed in 04/04/2020

ATTACHMENT A

```sql
SELECT *
FROM city_list
WHERE country like 'Brazil';
```

ATTACHMENT B

```sql
SELECT *
FROM city_data
WHERE city like 'Rio De Janeiro'
ORDER BY year DESC
LIMIT 110;
```

ATTACHMENT C

```sql
SELECT *
FROM global_data
ORDER BY year DESC
LIMIT 110;
```

ATTACHMENT D

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

YEARS_PERIOD = 10

def plotTemperaturesPerYears(years_global_temp, global_temp, years_rj_temp,
rj_temp):
    fig = plt.figure()
    plt.title("Years X Temperatures (10-year Moving Averages)")
    plt.plot(years_global_temp, global_temp, "blue", label="Global")
    plt.plot(years_rj_temp, rj_temp, "red", label="Rio de Janeiro")
    plt.ylabel("Temperatures (°C)")
    plt.xlabel("Years")
    plt.legend(loc="center right")
    fig.savefig('temperaturesPerYears.png')
    plt.show()

def calculeMovingAverage10(temp):
    mov_avg = []

    for i in range(0, len(temp) - YEARS_PERIOD + 1):
        sum_avg = sum(temp[i:i + YEARS_PERIOD]) / YEARS_PERIOD
        mov_avg.append(sum_avg)

    return mov_avg

def main():
    # Opening csv files as pandas data frames
    df_global_temp = pd.read_csv("Temperature_Global.csv", sep=",",
encoding="UTF8")
    df_rj_temp = pd.read_csv("Temperature_Rio_de_Janeiro.csv", sep=",",
encoding="UTF8")
```

```
    # Filling in possible null or NaN values
    df_global_temp = df_global_temp.fillna(0)
    df_rj_temp = df_rj_temp.fillna(0)

    # Separating data from years and temperatures, and sorting in an
increasing way
    years_global_temp = df_global_temp['year'].values[::-1]
    years_rj_temp = df_rj_temp['year'].values[::-1]
    global_temp = df_global_temp['avg_temp'].values[::-1]
    rj_temp = df_rj_temp['avg_temp'].values[::-1]

    plotTemperaturesPerYears(years_global_temp[0:101],
                             calculeMovingAverage10(global_temp),
                             years_rj_temp[0:101],
                             calculeMovingAverage10(rj_temp))

if __name__ == '__main__':
    main()
```