Explore Weather Trends – Data Analyst Nanodegree Project

High-level steps taken

1. Extracting the data from the database using SQL
2. Importing the data into Spyder (Python)
3. Cleaning up and understanding the data better (key considerations tackled here)
4. Calculating moving averages for the global and local temperature (Python)
5. Plotting a line chart and making observations (Python)

Step 1 – SQL used

Since I live in NYC now, I chose NYC as the local city

*SELECT * FROM global_data*

*SELECT * FROM city_data WHERE city = 'New York'*

Step 2 – Importing the data into Spyder (Python)

Once I had the CSV files on my laptop, I used the panda's *read_csv* function to import the files into the Python environment. While I could have done this much easily in Excel, I really wanted to explore the files using Python

Step 3 – Understanding the data better and handling key considerations

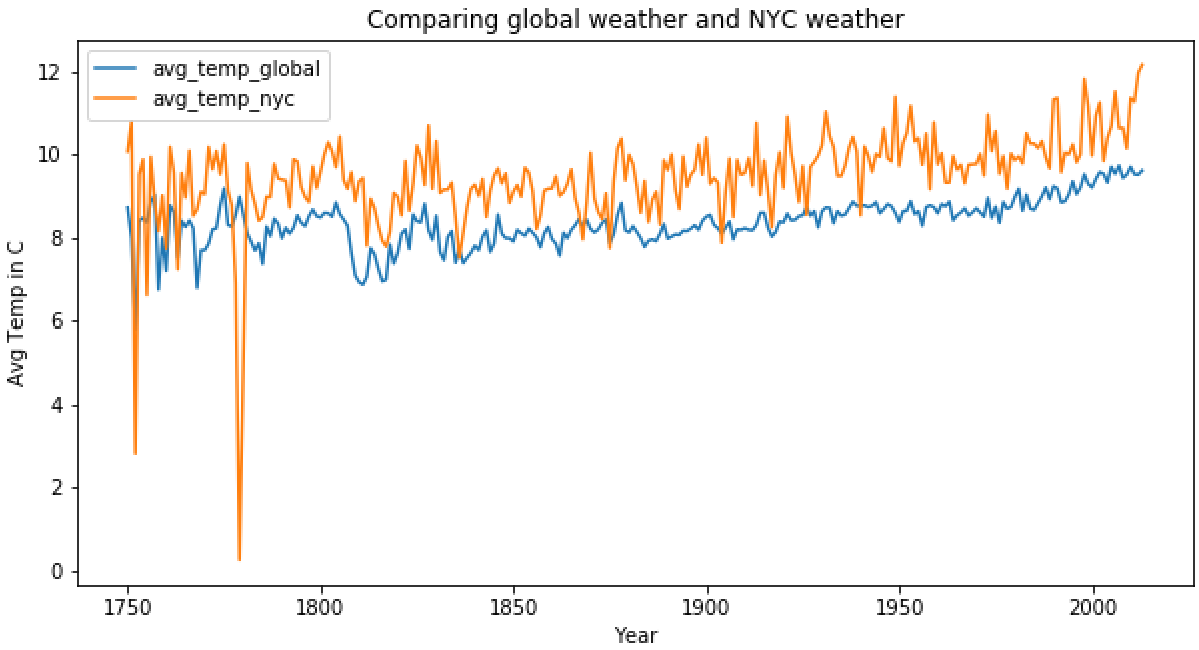
The first thing I did after getting the data into Python was comparing how the global data relates to my local NYC data. My **main considerations** were

- Are there any **duplicates** per year in each of the datasets?
  - It is important to make sure we do not have duplicates at the year level and I wanted to confirm that
- Are both the datasets covering the **same/similar time range**?
  - If the time range is super different, plotting the data points side by side would have revealed a wrong comparison
- Are there lots of **missing values** in the datasets?
  - Ignoring missing values will lead to cleaner results provided there are not too many and there were only few missing values in these data sets
- Do we have a good percentage of **common years between the datasets**?
  - It is important to compare the weather trends for the same years. So I did an INNER JOIN and retained only the years that overlapped which did not lose any value in the data

In the end, the data frame I created had 263 years that were common for both global/NYC weather

Step 4 – Calculating moving averages

Before calculating moving averages, I plotted the **raw temperatures** just to see how they look. As you can see below, the NYC data had two major dips in the 1700's. Otherwise, the lines more or less follow a pattern with NYC at a higher temperature in general.

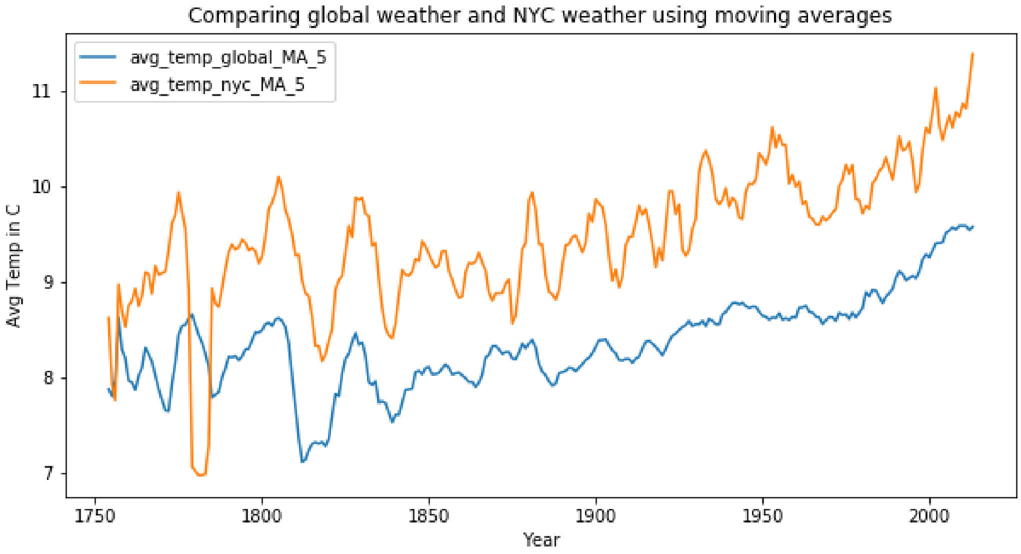


In order to eliminate the dip and get a better read at the line chart, I used the panda's inbuilt *rolling* function with a window size of five. Sample code below.

global_local_data['avg_temp_global_MA_5'] = global_local_data['avg_temp_global'].rolling(window=5).mean()

Step 5 – Line chart and observations

Using the newly calculated moving averages, I plotted the chart using panda's inbuilt *plot* function and then labeled the axes appropriately.



Observations

1. NYC data is almost always higher than the global data except for a period around ~1770 marked by the sharp dip seen in the line chart above
    a. This is because of two years (1752 and 1779) that have a much lower temperature than other years
2. Even though NYC is averaging at a higher temperature than global data, the vicissitudes are generally in the same direction between the data sets. This is not always true but it is a general observation
3. Over the course of the years (from 1800 to 2000), the average temperature for NYC has gone up (~9 to ~11.5) at a much higher rate than the global data (~8.5 to ~9.5) indicating that NYC is experiencing global warming issues at a bigger scale
4. The global data has one or two spikes but otherwise it is more or less centered around the mean whereas NYC data has major spikes across the years indicating a more volatile environment