# Explore Weather Trends

## Summary

In this project, I analyze the Berlin and global temperature data and compare the temperature trends of Berlin, Germany to overall global temperature trends.

## Tools Used

| | |
|---|---|
| **SQL** | Collecting Data |
| **Pandas** | Data Processing |
| **Seaborn & Matplotlib** | Plotting and Visualization Charts |
| **Scikit Learn** | Linear Regression Model |

## Collecting Data

I used Udacity SQL Workspace to extract data from the temperatures database and then download the results to separate CSV files.

**The Database Schema:**

**city_list**

Contains a list of cities and countries in the database.

**city_data**

Contains the average temperatures for each city by year.

**global_data**

Contains the average global temperatures by year from 1750 - 2015.

The SQL queries I made to extract the data from the temperature database.

**Finding Nearest City**

```
// this query shows only one city Berlin
SELECT * FROM city_list WHERE city LIKE  'Berlin'
```

**Extract Berlin Average Temperature per Year**

```
// this query shows city data of Berlin only
SELECT * FROM city_data WHERE city LIKE 'Berlin'
```

**Extract Global Average Temperature per Year**

```
// this query shows all the data from the global_data table
SELECT * FROM global_data
```

## Data Processing

After downloading data from the database I read all data through pandas in Jupyter Notebook.

**berlin_df**

- Has 4 columns year, city, country and avg_temp
- Has 271 rows
- In avg_temp column 4 data is missing from year 1746 to 1749
- Starting year 1743
- End year 2013

**global_df**

- Has 2 columns year and avg_temp
- Has 266 rows
- Has no missing data
- Starting year 1750
- End year 2015

Matching length of both dataframe.

- Dropping rows which have years less than 1750 in berlin_df dataframe.
- Dropping the last two rows from global_df dataframe.

```
#  copy only those rows which have years less than 1750
berlin_df = berlin_df[(berlin_df['year'] >= 1750)].copy()

# dropping the last two rows
global_df = global_df[(global_df['year'] <= 2013)].copy()
```

## Calculating Moving Average

Moving Average is a method in which means are calculated from a set of consecutive time-series values over a certain time period.

In this project, the 10-year moving average seems reasonable enough for capturing long-term trends and reflecting short-term fluctuations.

```
#moving average for berlin_df
berlin_df["moving_avg"] = berlin_df["avg_temp"].rolling(window = 10).mean()
```

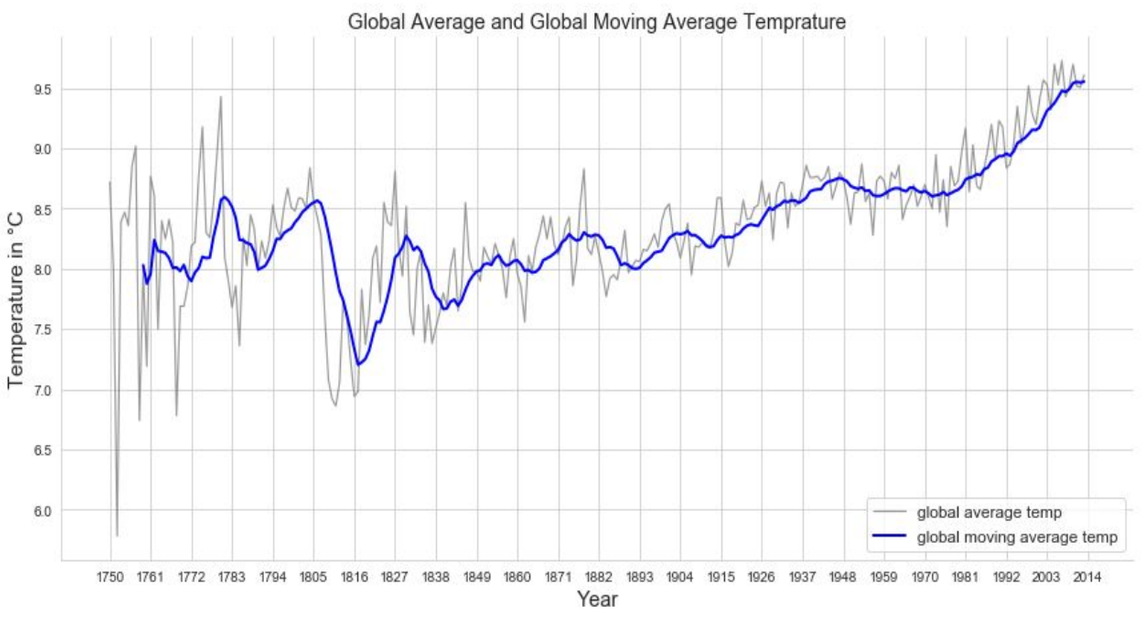
| | year | city | country | avg_temp | moving_avg |
|---|---|---|---|---|---|
| 0 | 1750 | Berlin | Germany | 9.83 | NaN |
| 1 | 1751 | Berlin | Germany | 9.75 | NaN |
| 2 | 1752 | Berlin | Germany | 4.84 | NaN |
| 3 | 1753 | Berlin | Germany | 8.72 | NaN |
| 4 | 1754 | Berlin | Germany | 8.49 | NaN |
| 5 | 1755 | Berlin | Germany | 8.26 | NaN |
| 6 | 1756 | Berlin | Germany | 9.62 | NaN |
| 7 | 1757 | Berlin | Germany | 9.15 | NaN |
| 8 | 1758 | Berlin | Germany | 8.25 | NaN |
| 9 | 1759 | Berlin | Germany | 9.04 | 8.595 |
| 10 | 1760 | Berlin | Germany | 8.99 | 8.511 |
| 11 | 1761 | Berlin | Germany | 9.47 | 8.483 |
| 12 | 1762 | Berlin | Germany | 8.53 | 8.852 |

```
#moving average for global_df
global_df["moving_avg"] = global_df["avg_temp"].rolling(window = 10).mean()
```

| | year | avg_temp | moving_avg |
|---|---|---|---|
| **0** | 1750 | 8.72 | NaN |
| **1** | 1751 | 7.98 | NaN |
| **2** | 1752 | 5.78 | NaN |
| **3** | 1753 | 8.39 | NaN |
| **4** | 1754 | 8.47 | NaN |
| **5** | 1755 | 8.36 | NaN |
| **6** | 1756 | 8.85 | NaN |
| **7** | 1757 | 9.02 | NaN |
| **8** | 1758 | 6.74 | NaN |
| **9** | 1759 | 7.99 | 8.030 |
| **10** | 1760 | 7.19 | 7.877 |
| **11** | 1761 | 8.77 | 7.956 |
| **12** | 1762 | 8.61 | 8.239 |

## Plotting Global Average and Global Moving Average Temperature

```
#plotting average and moving average of global temp
sns.lineplot(x = "year", y = "avg_temp", data = global_df, label = "global
average temp", alpha = 0.8, color = "grey", linewidth = 1.2)

sns.lineplot(x = "year", y = "moving_avg", data = global_df, label =
"global moving average temp", color = "blue", linewidth = 2)
```

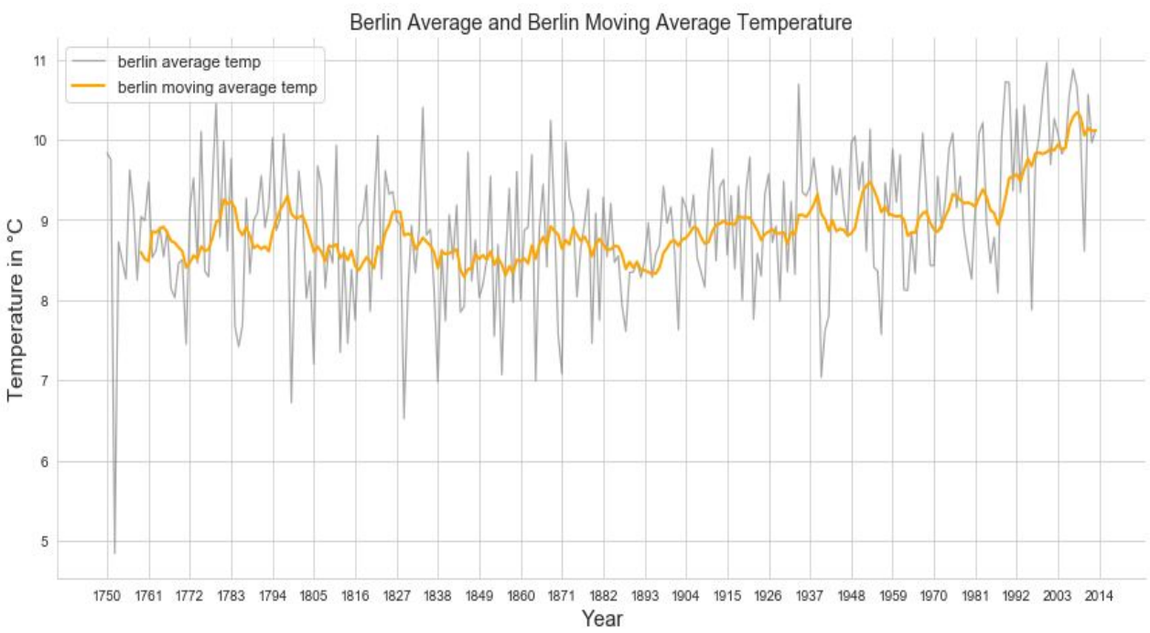


**Observations:**

- Global Average Temperature: 8.36 °C
- Maximum Global Average Temperature: 9.73 °C
- Minimum Global Average Temperature: 5.78 °C
- Huge temperature drop between 1805 and 1816.
- From 1893 average temperature has been increasing over time.
- Rapid increase in temperature from 1970.

## Plotting Berlin Average and Berlin Moving Average Temperature

```
#plotting average and moving average of Berlin temp
sns.lineplot(x = "year", y = "avg_temp", data = berlin_df, label = "berlin
average temp", alpha = 0.7, color = "grey", linewidth = 1.2)

sns.lineplot(x = "year", y = "moving_avg", data = berlin_df, label =
"berlin moving average temp", color = "orange", linewidth = 2)
```

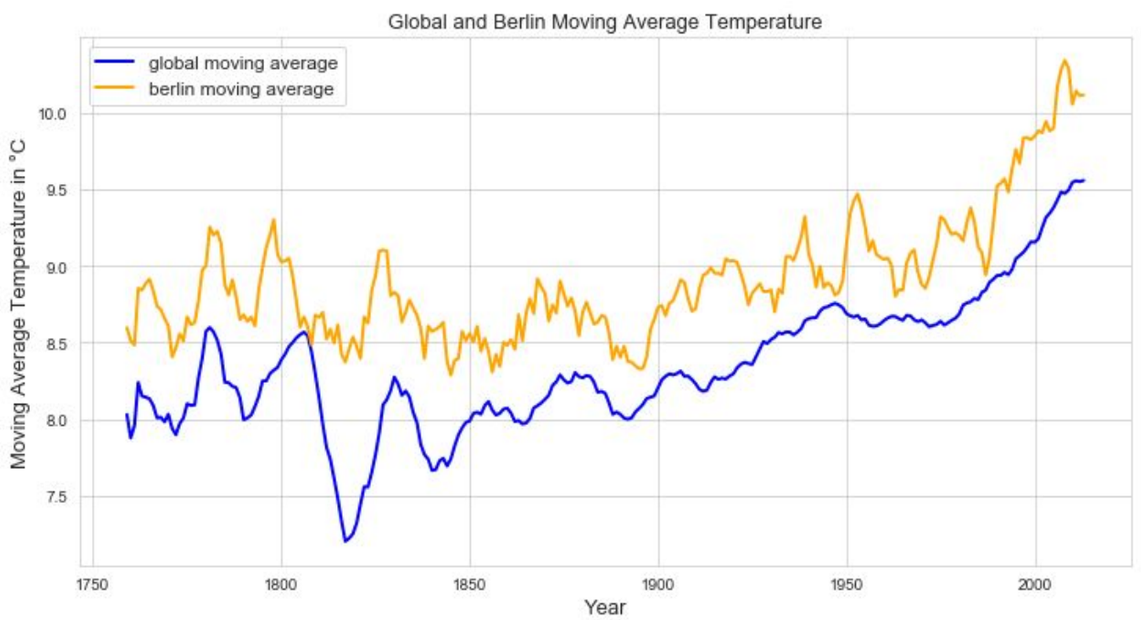



**Observations:**

- Average Temperature of Berlin: 8.92 °C
- Maximum Average Temperature of Berlin: 10.96 °C
- Minimum Average Temperature of Berlin: 4.84 °C
- From 1894 the average temperature of Berlin increased over time.
- Rapid increase in temperature from 1985.

## Plotting Global and Berlin Moving Average Temperature

```
#plotting moving average of global temp
sns.lineplot(x = "year", y = "moving_avg", data = global_df, label =
"global moving average", color = "blue", linewidth = 2)

#plotting moving average of berlin temp
sns.lineplot(x = "year", y = "moving_avg", data = berlin_df, label =
"berlin moving average", color= 'orange', linewidth = 2)
```



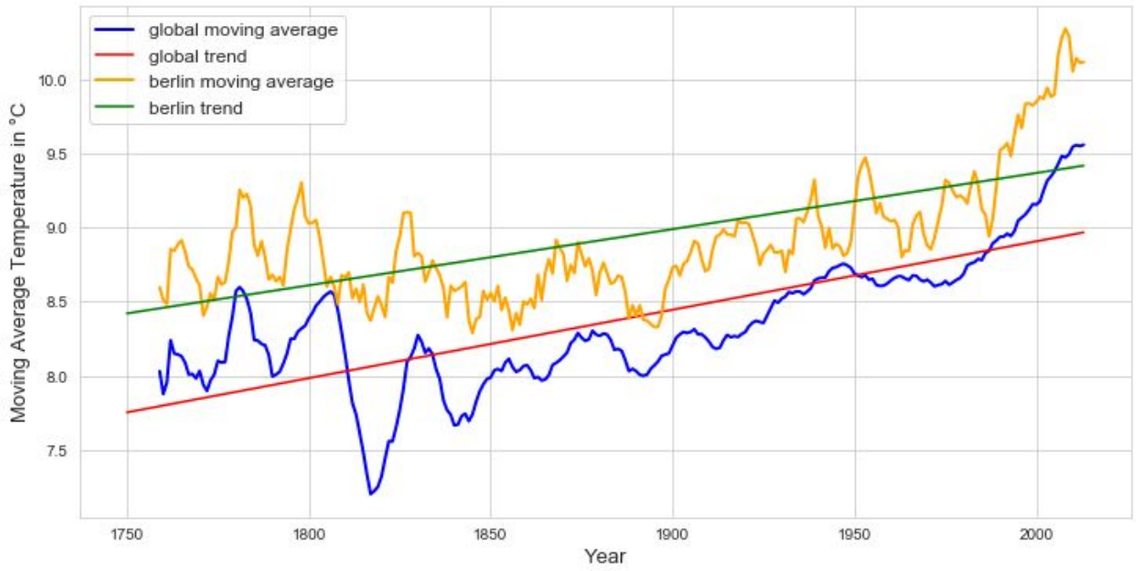

### Observations:

- Average Temperature of Berlin: 8.92 °C
- Global Average Temperature: 8.36 °C
- Average Temperature Difference: 0.558 °C
- Berlin temperature also dropped when there was a huge drop in global temperature around 1805.
- Average temperature of global and Berlin is increasing over time.
- Average temperature difference between Berlin and global is 0.558 °C.

## Comparing Trends

```
global_trend = LinearRegression()
X = global_df[["year"]]
y = global_df[["avg_temp"]]
global_trend.fit(X, y)
global_df["trend"] = global_trend.predict(X)

local_trend = LinearRegression()
X = berlin_df[["year"]]
y = berlin_df[["avg_temp"]]
local_trend.fit(X, y)
berlin_df["trend"] = local_trend.predict(X)
```

```
sns.lineplot(x = "year", y = "moving_avg", data = global_df, label =
"global moving average", color = "blue", linewidth = 2)

sns.lineplot(x = "year", y = "trend", data = global_df, label = "global
trend", color = "red")

sns.lineplot(x = "year", y = "moving_avg", data = berlin_df, label =
"berlin moving average", color = 'orange', linewidth = 2)

sns.lineplot(x = "year", y = "trend", data = berlin_df, label = "berlin
trend", color = 'green')
```




**Observations:**

- Global Slope: 0.00461
- Berlin Slope: 0.00378
- Slope of Berlin is smaller than the global slope, which means that the global average temperature is increasing faster.
- Both global and Berlin average temperature is increasing over time.
- Average temperature of Berlin is slightly higher than the global average temperature.

## Correlation Coefficient

```
#correlation between berlin average temp and global average temp
berlin_df[["avg_temp"]].corrwith(global_df["avg_temp"])
```

With 0.515 correlation coefficient Berlin data are moderately correlated with global data.

## Conclusion

- Average temperature of global and Berlin is rising over time.
- From 1980 temperature is continuously rising.
- Berlin is getting hotter over time.
- Average temperature change between global and Berlin is very small.
- Temperature of Berlin is increasing because large natural landscapes are replaced with buildings and asphalt streets that absorb more heat which makes cities warmer.