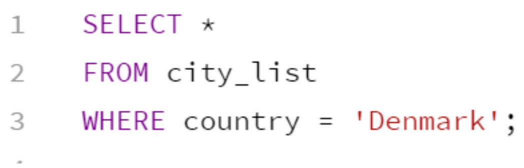
Project 1: Exploring Weather Trends

Outline

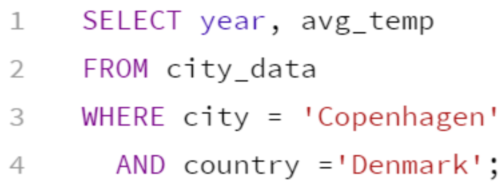
Use SQL to extract the data in different datasets.

1. Check the city_list dataset to find the close city in my region by using a condition clause in SQL. The output data is Copenhagen city in Denmark.



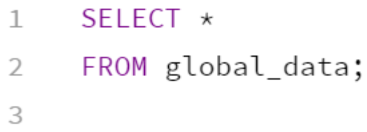
```sql
SELECT *
FROM city_list
WHERE country = 'Denmark';
```

2. Using SQL to query all the year and avg_temperature data of Copenhagen in city_data dataset and then download it as the csv file.



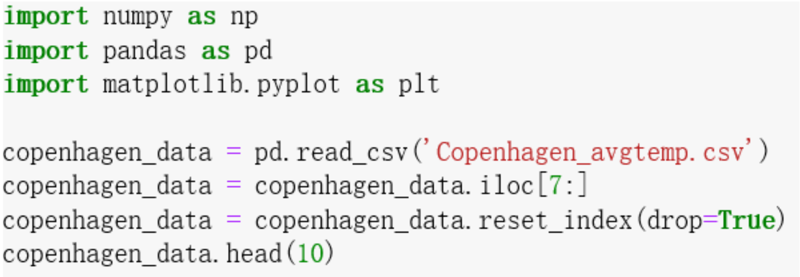
```sql
SELECT year, avg_temp
FROM city_data
WHERE city = 'Copenhagen'
  AND country ='Denmark';
```

3. Using SQL to query all the year and avg_temperature data in globe_data dataset and then download it as the other csv file.

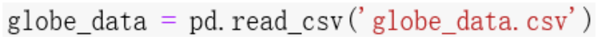


```sql
SELECT *
FROM global_data;

```

Use Python to read the above two csv files (Copenhagen_avgtemp and globe_data) as new DataFrames.

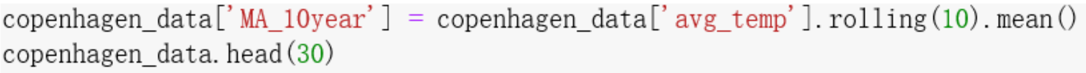


```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

copenhagen_data = pd.read_csv(’Copenhagen_avgtemp.csv’)
copenhagen_data = copenhagen_data.iloc[7:]
copenhagen_data = copenhagen_data.reset_index(drop=True)
copenhagen_data.head(10)
```

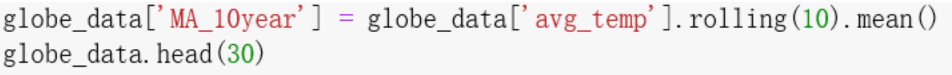


```python
globe_data = pd.read_csv(’globe_data.csv’)
```

Calculate the moving average temperature of Copenhagen and globe DataFrames by using .rolling( ).mean( ) functions and add the result as the new column in each DataFrames separately.

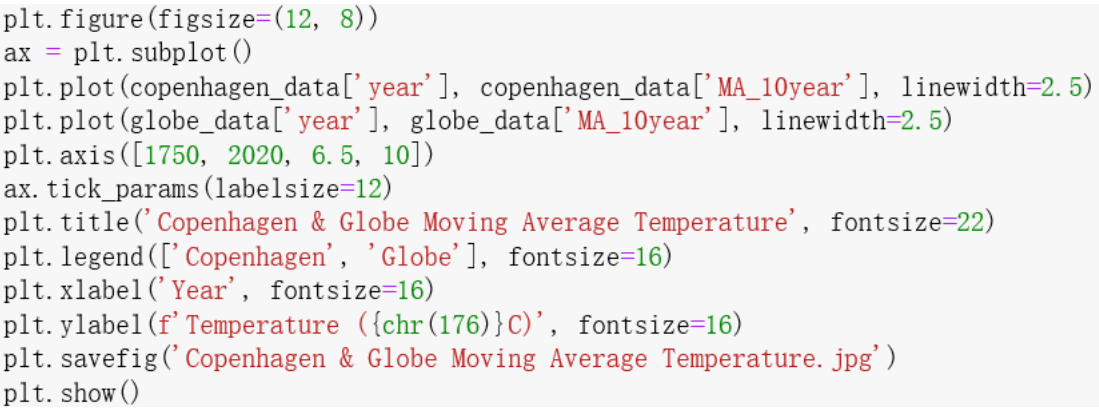


```
copenhagen_data['MA_10year'] = copenhagen_data['avg_temp'].rolling(10).mean()
copenhagen_data.head(30)
```



```
globe_data['MA_10year'] = globe_data['avg_temp'].rolling(10).mean()
globe_data.head(30)
```

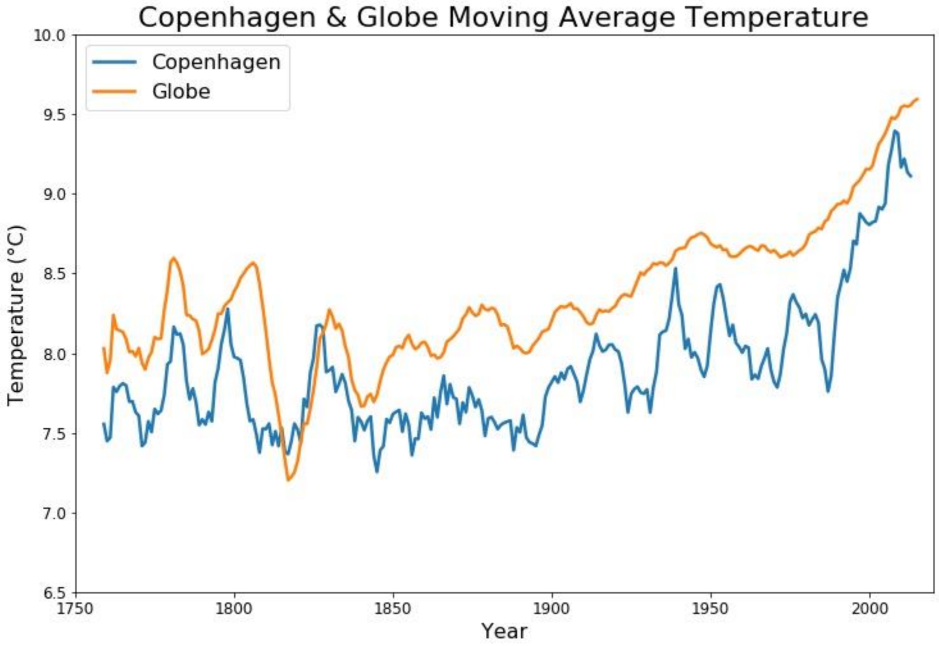
Use matplotlib in Python to plot the two temperature trends as the line chart and save it as .jpg file. The key consideration is the accuracy of the data, the readability of the plot, the impact of the daily or short term's fluctuation of the long term trends and the range of the axis for showing the data trends well.



```
plt.figure(figsize=(12, 8))
ax = plt.subplot()
plt.plot(copenhagen_data['year'], copenhagen_data['MA_10year'], linewidth=2.5)
plt.plot(globe_data['year'], globe_data['MA_10year'], linewidth=2.5)
plt.axis([1750, 2020, 6.5, 10])
ax.tick_params(labelsize=12)
plt.title('Copenhagen & Globe Moving Average Temperature', fontsize=22)
plt.legend(['Copenhagen', 'Globe'], fontsize=16)
plt.xlabel('Year', fontsize=16)
plt.ylabel(f'Temperature ({chr(176)}C)', fontsize=16)
plt.savefig('Copenhagen & Globe Moving Average Temperature.jpg')
plt.show()
```

- ✧ All the Python parts were done in Jupyter Notebook.
- ✧ The moving average methods was found in Stack overflow (https://stackoverflow.com/questions/43437657/rolling-mean-on-pandas-on-a-specific-column). The details in functions were found in https://pandas.pydata.org/pandas-docs/stable/user_guide/computation.html.

Line Chart



Observations

1. Both the average temperatures of Copenhagen city and the whole world showed the increasing trend after 1850.
2. Obviously, the average temperature of Copenhagen is always lower than that of the whole world, except for the period between 1810 and 1830.
3. Before 1900, both two average temperatures showed the similar trends. However, the average temperature of Copenhagen is more fluctuant than that of the whole world after 1900.
4. Both two temperature trends showed the sharp increase after 1990, but the Copenhagen one has larger increase than that of the world data.
5. Between 1850 and 1900, the average temperature is quite stable (around 7.5±2.5 ℃) when compared to that of the globe data.