Exploring Weather Trends – Project

Name: Meshael Almusairii

Step 1: Tools used

The tools that I have used while solving this project are:

- SQL query.
- Jupyter Notebook.
- Matplotlib library to draw line chart.

Step 2: calculating the moving average

The moving average is calculated by taking the average of the last 10 years.

Step 3: key considerations when visualizing the trends

1- I filled the empty values with the median average temperature of each city individually (because each city has its own average temperature).
2- Then I create new dataFrame to calculate moving average for each 10 years for each city individually.
3- After that, I removed the first 9 rows for each city that does not have average temperature for 10 years (because the first 9 rows of each city do not have average temperature for 10 years).
4- Next, I got the data for the nearest city "Riyadh".
5- After that, I used the (global_data) to calculate the moving average for each 10 years.

But before plotting the line chart for nearest city and global, I did some edits:

1- First, I printed the min and max years for both data.
2- After that, I noticed the following information:

	Riyadh	**Global**
Min	1852	1759
Max	2013	2015

Table 1 min and max years for Riyadh and global data before

3- Then, I decided to get the data with same year in both datasets. So, the min and max now are:

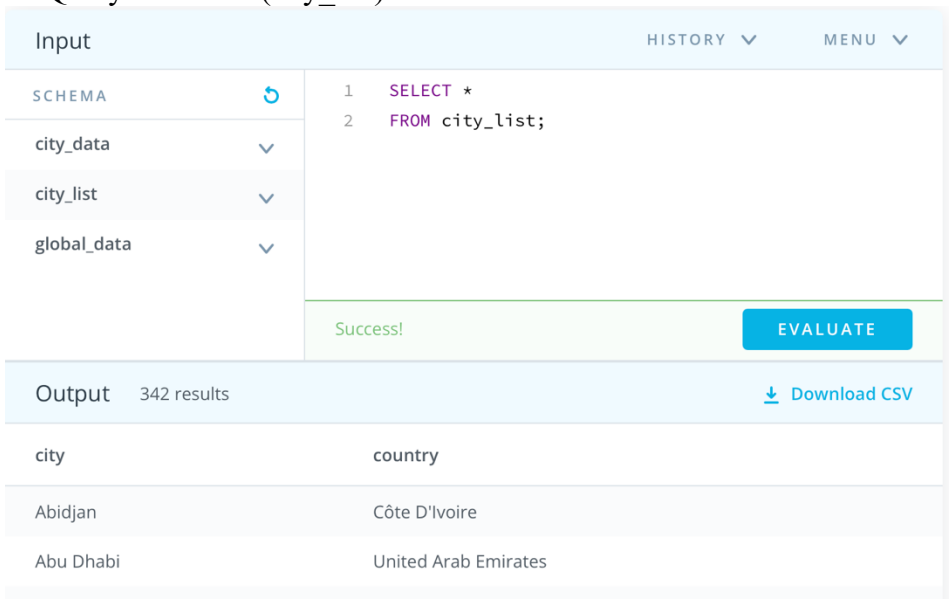
	Riyadh	**Global**
Min	1852	1852
Max	2013	2013

Table 2 min and max years for Riyadh and global data after

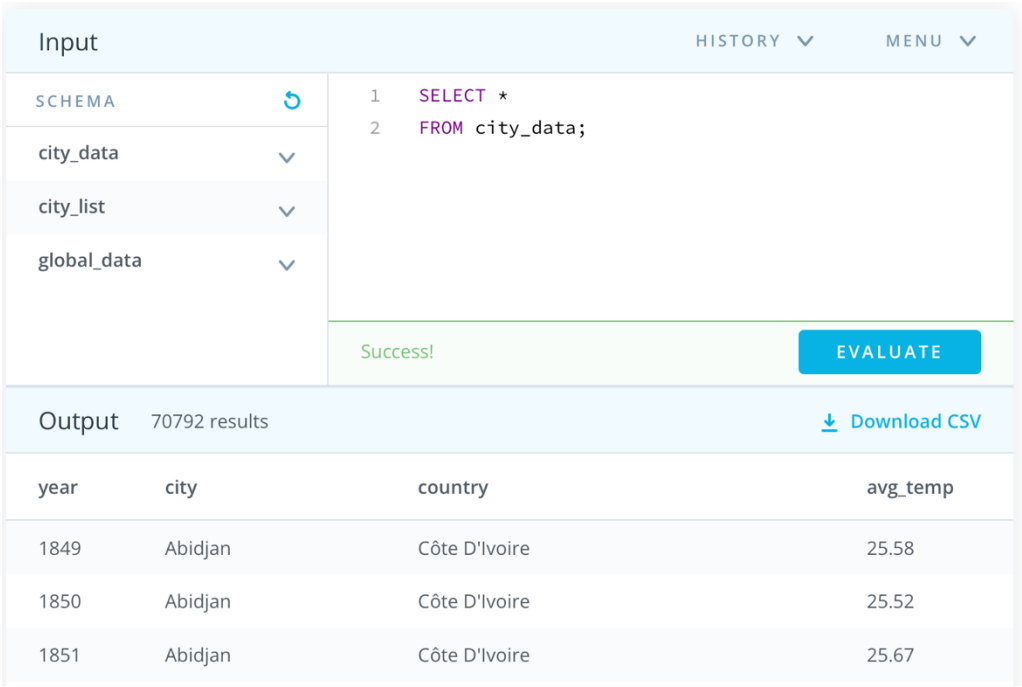
4- Finally, I plotted both charts for Riyadh and global data to make observations.

Step 4: the following are the steps that I have make to get to the final solution

- **1: Extracting the data from the database using SQL query.**
  - Query to extract (city_list) data.



  - Query to extract (city_data) data.



  - Query to extract (global_data) data.

Input HISTORY MENU

SCHEMA

city_data

city_list

global_data

```
1  SELECT *
2  FROM global_data;
```

Success! EVALUATE

Output 266 results Download CSV

year	avg_temp
1750	8.72
1751	7.98
1752	5.78

- **2: Code (in details with explanation)**

import libraries

In [1]:
```
import csv
import pandas as pd
import matplotlib.pyplot as plt
```

load data

In [2]:
```
city_data=pd.read_csv("city_data.csv")
city_list=pd.read_csv("city_list.csv")
global_data=pd.read_csv("global_data.csv")
```

city_data work

In [3]:
```
city_data["10_Y_MA"]=0
city_data
```

Out[3]:

	year	city	country	avg_temp	10_Y_MA
0	1849	Abidjan	Côte D'Ivoire	25.58	0
1	1850	Abidjan	Côte D'Ivoire	25.52	0
2	1851	Abidjan	Côte D'Ivoire	25.67	0
3	1852	Abidjan	Côte D'Ivoire	NaN	0
4	1853	Abidjan	Côte D'Ivoire	NaN	0
...	...	...	...	...	...
...	...	...	...	...	...
70787	2009	Zapopan	Mexico	21.76	0
70788	2010	Zapopan	Mexico	20.90	0
70789	2011	Zapopan	Mexico	21.55	0
70790	2012	Zapopan	Mexico	21.52	0
70791	2013	Zapopan	Mexico	22.19	0

70792 rows × 5 columns

check if there is empty values

In [4]:
```
city_data.isnull().sum()
```

Out[4]:
```
year           0
city           0
country        0
avg_temp    2547
10_Y_MA        0
dtype: int64
```

to get unique names of the cities

In [5]:
```
city_unique= city_data["city"].unique()
len(city_unique)
```

Out[5]: 329

create new dataFrame to calculate moving average for each 10 years for each city individualy

In [6]:
```
new_city_data= pd.DataFrame(columns=city_data.columns)
for i in city_unique:
    sub = city_data[city_data["city"] == i]
    sub["avg_temp"] = sub["avg_temp"].fillna(sub["avg_temp"].median())
    for j in range(9,len(sub)):
        count=0
        for k in range(j-9,j+1):
            count+=sub["avg_temp"].iloc[k]  # count for ten years
        sub["10_Y_MA"].iloc[j]=count/10.0   # get the average
    print(sub.head(12))
    new_city_data=new_city_data.append(sub, ignore_index=True)    # append the sub dataFrame to the new dataFrame
                                                                  # after calculate the moving average
```

```
10  1859  Abidjan  Côte D'Ivoire     25.92   25.897
11  1860  Abidjan  Côte D'Ivoire     25.46   25.891
     year       city               country  avg_temp  10_Y_MA
165  1843  Abu Dhabi  United Arab Emirates     26.04    0.000
166  1844  Abu Dhabi  United Arab Emirates     26.26    0.000
167  1845  Abu Dhabi  United Arab Emirates     26.44    0.000
168  1846  Abu Dhabi  United Arab Emirates     26.44    0.000
169  1847  Abu Dhabi  United Arab Emirates     26.44    0.000
170  1848  Abu Dhabi  United Arab Emirates     25.83    0.000
171  1849  Abu Dhabi  United Arab Emirates     26.01    0.000
172  1850  Abu Dhabi  United Arab Emirates     25.69    0.000
173  1851  Abu Dhabi  United Arab Emirates     26.25    0.000
174  1852  Abu Dhabi  United Arab Emirates     26.44   26.184
175  1853  Abu Dhabi  United Arab Emirates     26.44   26.224
176  1854  Abu Dhabi  United Arab Emirates     26.44   26.242
     year   city  country  avg_temp  10_Y_MA
336  1856  Abuja  Nigeria     26.93    0.000
337  1857  Abuja  Nigeria     24.67    0.000
338  1858  Abuja  Nigeria     25.87    0.000
339  1859  Abuja  Nigeria     25.31    0.000
```

In [7]:
```
new_city_data
```

Out[7]:

	year	city	country	avg_temp	10_Y_MA
0	1849	Abidjan	Côte D'Ivoire	25.58	0.000
1	1850	Abidjan	Côte D'Ivoire	25.52	0.000
2	1851	Abidjan	Côte D'Ivoire	25.67	0.000
3	1852	Abidjan	Côte D'Ivoire	26.23	0.000
4	1853	Abidjan	Côte D'Ivoire	26.23	0.000
...	...	...	...	...	...
70787	2009	Zapopan	Mexico	21.76	21.365
70788	2010	Zapopan	Mexico	20.90	21.343
70789	2011	Zapopan	Mexico	21.55	21.375
70790	2012	Zapopan	Mexico	21.52	21.377
70791	2013	Zapopan	Mexico	22.19	21.445

70792 rows × 5 columns

to check again if there is no NaN values

In [8]:
```
new_city_data.isnull().sum()
```

Out[8]:
```
year        0
city        0
country     0
avg_temp    0
10_Y_MA     0
dtype: int64
```

to remove empty rows that does not have "10_Y_MA"

In [9]:
```
new_city_data = new_city_data[new_city_data["10_Y_MA"] != 0]
```

In [10]:
```
new_city_data
```

Out[10]:

	year	city	country	avg_temp	10_Y_MA
9	1858	Abidjan	Côte D'Ivoire	25.49	25.863
10	1859	Abidjan	Côte D'Ivoire	25.92	25.897
11	1860	Abidjan	Côte D'Ivoire	25.46	25.891
12	1861	Abidjan	Côte D'Ivoire	25.67	25.891
13	1862	Abidjan	Côte D'Ivoire	25.17	25.785
...	...	...	...	...	...
70787	2009	Zapopan	Mexico	21.76	21.365
70788	2010	Zapopan	Mexico	20.90	21.343
70789	2011	Zapopan	Mexico	21.55	21.375
70790	2012	Zapopan	Mexico	21.52	21.377
70791	2013	Zapopan	Mexico	22.19	21.445

67831 rows × 5 columns

reset the indexes

In [11]:
```
new_city_data.reset_index(drop=True)
```

Out[11]:

	year	city	country	avg_temp	10_Y_MA
0	1858	Abidjan	Côte D'Ivoire	25.49	25.863
1	1859	Abidjan	Côte D'Ivoire	25.92	25.897
2	1860	Abidjan	Côte D'Ivoire	25.46	25.891
3	1861	Abidjan	Côte D'Ivoire	25.67	25.891
4	1862	Abidjan	Côte D'Ivoire	25.17	25.785
...	...	...	...	...	...
67826	2009	Zapopan	Mexico	21.76	21.365
67827	2010	Zapopan	Mexico	20.90	21.343
67828	2011	Zapopan	Mexico	21.55	21.375
67829	2012	Zapopan	Mexico	21.52	21.377
67830	2013	Zapopan	Mexico	22.19	21.445

67831 rows × 5 columns

to get data for nearest city "Riyadh"

In [12]:
```
riyadh_data=new_city_data[new_city_data["city"]=="Riyadh"]
riyadh_data
```

Out[12]:

	year	city	country	avg_temp	10_Y_MA
55247	1852	Riyadh	Saudi Arabia	24.85	23.489
55248	1853	Riyadh	Saudi Arabia	24.93	23.508
55249	1854	Riyadh	Saudi Arabia	24.72	24.435
55250	1855	Riyadh	Saudi Arabia	24.92	24.845
55251	1856	Riyadh	Saudi Arabia	24.57	24.787
...	...	...	...	...	...
55404	2009	Riyadh	Saudi Arabia	26.71	26.440
55405	2010	Riyadh	Saudi Arabia	27.37	26.522
55406	2011	Riyadh	Saudi Arabia	26.40	26.495
55407	2012	Riyadh	Saudi Arabia	26.83	26.534
55408	2013	Riyadh	Saudi Arabia	27.78	26.650

162 rows × 5 columns

global_data work

In [13]:
```
global_data["10_Y_MA"]=0
global_data
```

Out[13]:

	year	avg_temp	10_Y_MA
0	1750	8.72	0
1	1751	7.98	0
2	1752	5.78	0
3	1753	8.39	0
4	1754	8.47	0
...	...	...	...
261	2011	9.52	0
262	2012	9.51	0
263	2013	9.61	0
264	2014	9.57	0
265	2015	9.83	0

266 rows × 3 columns

to check if there are missing values

In [14]:
```
global_data.isnull().sum()
```

Out[14]:
```
year        0
avg_temp    0
10_Y_MA     0
dtype: int64
```

to calculate the moving average for ten years for global data

In [15]:
```
for i in range(9,len(global_data)):
    count=0
    for k in range(i-9,i+1):
        count+=global_data["avg_temp"].iloc[k]
    global_data["10_Y_MA"].iloc[i]=count/10.0
```

```
/Users/mashaelalmus/opt/anaconda3/lib/python3.7/site-packages/pandas/core/indexing.py:1732: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  self._setitem_single_block(indexer, value, name)
```

In [16]:
```
global_data
```

	year	avg_temp	10_Y_MA
0	1750	8.72	0.000
1	1751	7.98	0.000
2	1752	5.78	0.000
3	1753	8.39	0.000
4	1754	8.47	0.000
...	...	...	...
261	2011	9.52	9.554
262	2012	9.51	9.548
263	2013	9.61	9.556
264	2014	9.57	9.581
265	2015	9.83	9.594

to remove empty rows that does not have "10_Y_MA"¶

In [17]:
```
new_global_data = global_data[global_data["10_Y_MA"] != 0]
```

reset the index

In [18]:
```
new_global_data=new_global_data.reset_index(drop=True)
```

In [19]:
```
new_global_data
```

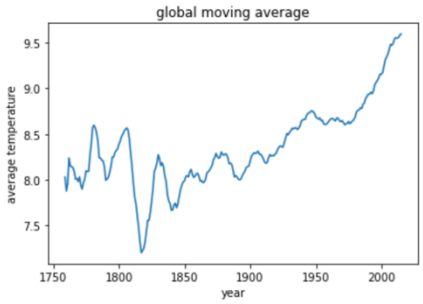
Out[19]:

	year	avg_temp	10_Y_MA
0	1759	7.99	8.030
1	1760	7.19	7.877
2	1761	8.77	7.956
3	1762	8.61	8.239
4	1763	7.50	8.150
...	...	...	...
252	2011	9.52	9.554
253	2012	9.51	9.548
254	2013	9.61	9.556
255	2014	9.57	9.581
256	2015	9.83	9.594

257 rows × 3 columns

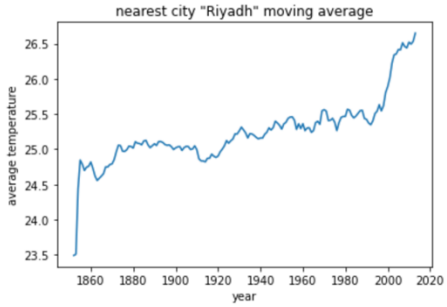
Plotting Global data

In [20]:
```
plt.plot(new_global_data["year"],new_global_data["10_Y_MA"])
plt.title('global moving average')
plt.xlabel('year')
plt.ylabel('average temperature')
plt.show()
```



Plotting nearest city data

In [21]:
```
plt.plot(riyadh_data["year"],riyadh_data["10_Y_MA"])
plt.title('nearest city "Riyadh" moving average')
plt.xlabel('year')
plt.ylabel('average temperature')
plt.show()
```



Compare between global and nearest city moving average temperature

A: Global 10 years moving average temperature

B: Nearest city "Riyadh" 10 years moving average temperature

First: the start and end years for A and B should be the same, so the data will be equal in years to compare between A and B

In [22]:
```
min_global=new_global_data["year"].min()
max_global=new_global_data["year"].max()
min_nearest=riyadh_data["year"].min()
max_nearest=riyadh_data["year"].max()
print("global : ",min_global , max_global)
print("riyadh : ",min_nearest , max_nearest)
```

```
global :  1759 2015
riyadh :  1852 2013
```

In [23]:
```
final_global_data = new_global_data[new_global_data["year"]>=1852]
final_global_data = final_global_data[new_global_data["year"]<=2013]
```

```
/Users/mashaelalmus/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:2: UserWarning: Boolean Series
key will be reindexed to match DataFrame index.
```

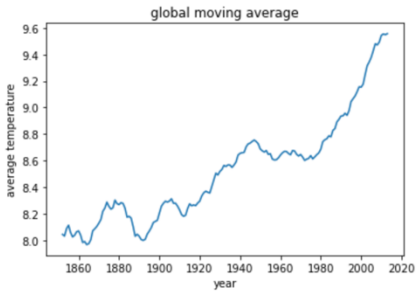
now the data are equal so we can compare

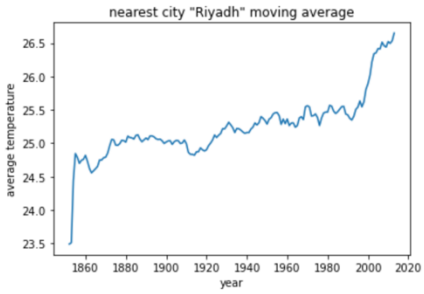
In [24]:
```
min_global=final_global_data["year"].min()
max_global=final_global_data["year"].max()
min_nearest=riyadh_data["year"].min()
max_nearest=riyadh_data["year"].max()
print("global : ",min_global , max_global)
print("riyadh : ",min_nearest , max_nearest)
```

```
global :  1852 2013
riyadh :  1852 2013
```

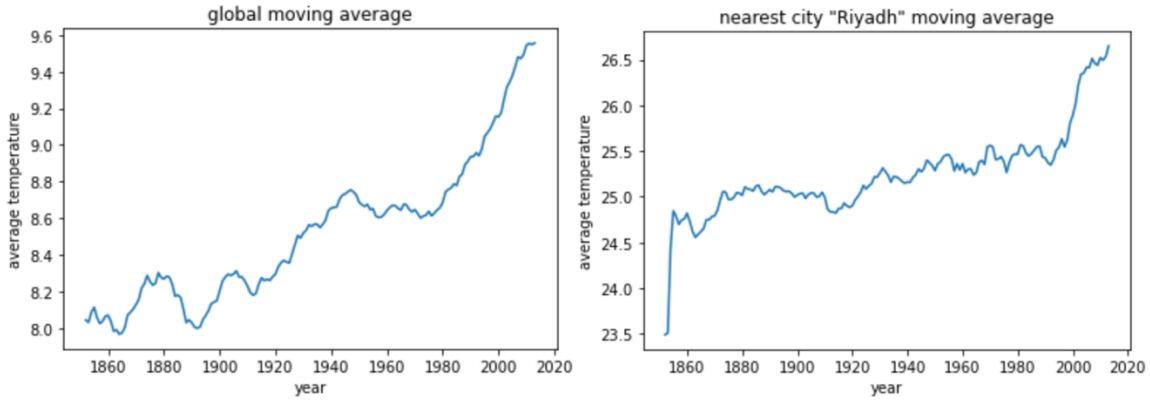
In [25]:
```
plt.plot(final_global_data["year"],final_global_data["10_Y_MA"])
plt.title('global moving average')
plt.xlabel('year')
plt.ylabel('average temperature')
plt.show()

plt.plot(riyadh_data["year"],riyadh_data["10_Y_MA"])
plt.title('nearest city "Riyadh" moving average')
plt.xlabel('year')
plt.ylabel('average temperature')
plt.show()
```





- **3: Observations:**



The overall trends showed that both global and nearest city "Riyadh" temperatures are increasing over the years, and the world is getting hotter over time.

Differences:

- The city where I live "Riyadh" is hotter that global temperature, and it has been consistent over time.
- From 1852 to 1855 the temperature in "Riyadh" has increased by 1.356, while the global temperature only increased by only 0.069.
- From 1880 to 1900 the temperature in "Riyadh" has approximately the same, while the global temperature goes up and down.

Similarities:

- Both the average temperature of global and nearest city "Riyadh" is increasing over the years.
- The world is getting hotter over time.