Exploring Weather Trends

1. Extracting Data

- Extracting the weather data by using SQL software installed in the Udacity website
- Select local cities such as Bangkok and the neighbouring cities of Singapore and Kuala Lumpur to analyze the trend and see whether they are the same as Bangkok.

a) Use SQL to extract Bangkok average temperature data as well as Global average temperature data and join these two tables together by using JOIN on year data of both tables.

Input HISTORY MENU

SCHEMA

city_data

city_list

global_data

```
1  SELECT city_data.city AS city,
2  city_data.year AS year,
3  city_data.avg_temp AS avg_temp,
4  global_data.avg_temp AS global_avg_temp
5  FROM city_data
6  JOIN global_data ON global_data.year = city_data.year
7  WHERE city_data.city = 'Bangkok'
```

Success! EVALUATE

Output 198 results Download CSV

city	year	avg_temp	global_avg_temp
Bangkok	1816	25.96	6.94
Bangkok	1817	25.83	6.98
Bangkok	1818	26.48	7.83
Bangkok	1819	25.90	7.37
Bangkok	1820	26.42	7.62
Bangkok	1821	26.81	8.09

b) Use SQL to extract Singapore average temperature data as well as global average temperature data joining the two tables together by year data

Input HISTORY MENU

SCHEMA

city_data

city_list

global_data

```
1  SELECT city_data.city AS city,
2  city_data.year AS year,
3  city_data.avg_temp AS avg_temp,
4  global_data.avg_temp AS global_avg_temp
5  FROM city_data
6  JOIN global_data ON global_data.year = city_data.year
7  WHERE city_data.city = 'Singapore'
```

Success! EVALUATE

Output 189 results Download CSV

city	year	avg_temp	global_avg_temp
Kuala Lumpur	1825	26.46	8.39
Kuala Lumpur	1826		8.36
Kuala Lumpur	1827		8.81
Kuala Lumpur	1828		8.17
Kuala Lumpur	1829		7.94

c) Use SQL to extract Kuala Lumpur average temperature data as well as global average temperature data joining the two tables together by year data

Input HISTORY MENU

SCHEMA

city_data

city_list

global_data

```
SELECT city_data.city AS city,
city_data.year AS year,
city_data.avg_temp AS avg_temp,
global_data.avg_temp AS global_avg_temp
FROM city_data
JOIN global_data ON global_data.year = city_data.year
WHERE city_data.city = 'Kuala Lumpur'
```

Success! EVALUATE

Output 189 results Download CSV

city	year	avg_temp	global_avg_temp
Kuala Lumpur	1825	26.46	8.39
Kuala Lumpur	1826		8.36
Kuala Lumpur	1827		8.81
Kuala Lumpur	1828		8.17
Kuala Lumpur	1829		7.94
Kuala Lumpur	1830		8.52
Kuala Lumpur	1831		7.64
Kuala Lumpur	1832		7.45

2. Open the saved CSV of these 3 cities data and visualize the trend in Python

a) Import the Bangkok CSV file and explore its data

- Check the head and tail of the data of Bangkok and see that there is an increase of average temperature of Bangkok from 25.96 in 1816 to 28.98 in 2013. Global average temperature has also increased over the 197 years period from 6.94 to 9.61.

Import and Explore Data

In [307]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
BKK = pd.read_csv('/Users/atikarnpattamavichaiporn/Downloads/BKK.csv')
```

In [308]:
```
BKK.head()
```

Out[308]:

	city	year	avg_temp	global_avg_temp
0	Bangkok	1816	25.96	6.94
1	Bangkok	1817	25.83	6.98
2	Bangkok	1818	26.48	7.83
3	Bangkok	1819	25.90	7.37
4	Bangkok	1820	26.42	7.62

In [309]:
```
BKK.tail()
```

Out[309]:

	city	year	avg_temp	global_avg_temp
193	Bangkok	2009	27.79	9.51
194	Bangkok	2010	28.54	9.70
195	Bangkok	2011	27.56	9.52
196	Bangkok	2012	28.48	9.51
197	Bangkok	2013	28.98	9.61

In [310]:
```
BKK.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 198 entries, 0 to 197
Data columns (total 4 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   city             198 non-null    object
 1   year             198 non-null    int64
 2   avg_temp         190 non-null    float64
 3   global_avg_temp  198 non-null    float64
dtypes: float64(2), int64(1), object(1)
memory usage: 6.3+ KB
```

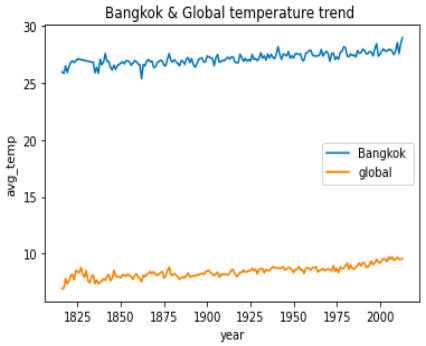
b) Visualize the trend of the average temperature in Bangkok compared to the world average temperature.

- Slight upward trend can be identified from the graph of both global and Bangkok average temperature.
- Bangkok temperatures were approximately 20 degrees hotter than the world average

Visualize the trend of temperatures

```
In [311]: sns.lineplot(data=BKK, y='avg_temp', x='year')
          sns.lineplot(data=BKK, y='global_avg_temp', x='year')
          plt.legend(['Bangkok ', 'global'])
          plt.title('Bangkok & Global temperature trend ')

Out[311]: Text(0.5, 1.0, 'Bangkok & Global temperature trend ')
```



3. Calculate the Moving Average of Bangkok and Global Average Temperature over the period of 10 years, 15 years, and 20 years

a) Calculate the Moving Average

Calculate Moving Average of 10, 15, and 20 years period

```
In [312]: # calculate Moving Average of BKK average temperature over 10 years
          BKK['BKK_MA_10'] = BKK.avg_temp.rolling(10, min_periods=1).mean()
          # calculate Moving Average of global average temperature over 10 years
          BKK['global_MA_10'] = BKK.global_avg_temp.rolling(10, min_periods=1).mean()
          # calculate Moving Average of BKK average temperature over 15 years
          BKK['BKK_MA_15'] = BKK.avg_temp.rolling(15, min_periods=1).mean()
          # calculate Moving Average of global average temperature over 15 years
          BKK['global_MA_15'] = BKK.global_avg_temp.rolling(15, min_periods=1).mean()
          # calculate Moving Average of BKK average temperature over 20 years
          BKK['BKK_MA_20'] = BKK.avg_temp.rolling(20, min_periods=1).mean()
          # calculate Moving Average of global average temperature over 20 years
          BKK['global_MA_20'] = BKK.global_avg_temp.rolling(20, min_periods=1).mean()
```

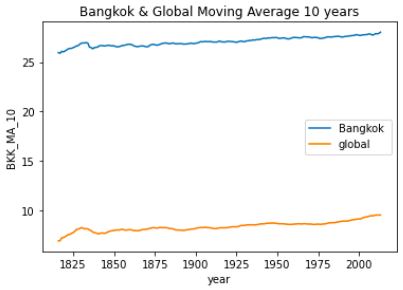
b) Visualize the moving average of 10, 15 and 20 years for Bangkok average temperature with Global average temperature

- From the charts, the moving average over the 20 years period smoothen out the noise of the data the most, making the chart the most smooth when comparing the 10 years moving average and 15 years moving average graph. The chart of moving average over 20 years help see the trend clearer.

Visualize the BKK Moving Average with Global trend

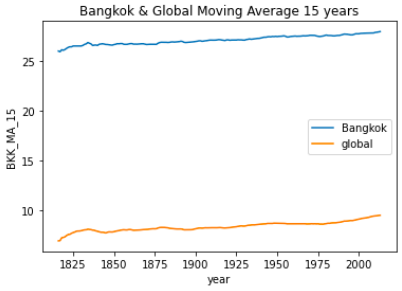
In [313]:
```
sns.lineplot(data=BKK, y='BKK_MA_10', x='year')
sns.lineplot(data=BKK, y='global_MA_10', x='year')
plt.legend(['Bangkok ', 'global'])
plt.title('Bangkok & Global Moving Average 10 years')
```

Out[313]:
```
Text(0.5, 1.0, 'Bangkok & Global Moving Average 10 years')
```



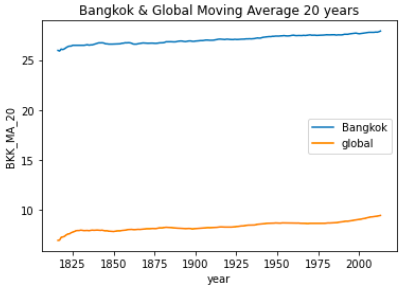
In [314]:
```
sns.lineplot(data=BKK, y='BKK_MA_15', x='year')
sns.lineplot(data=BKK, y='global_MA_15', x='year')
plt.legend(['Bangkok', 'global'])
plt.title('Bangkok & Global Moving Average 15 years')
```

Out[314]:
```
Text(0.5, 1.0, 'Bangkok & Global Moving Average 15 years')
```



In [315]:
```
sns.lineplot(data=BKK, y='BKK_MA_20', x='year')
sns.lineplot(data=BKK, y='global_MA_20', x='year')
plt.legend(['Bangkok', 'global'])
plt.title('Bangkok & Global Moving Average 20 years')
```

Out[315]:
```
Text(0.5, 1.0, 'Bangkok & Global Moving Average 20 years')
```



4. Import Singapore and Kuala Lumpur CSV data and calculate the moving average of 20 years period

- Since moving average of 20 years period smoothen out the trend the most

Import Singapore Data & Calculate Moving Average

In [316]:
```
SG = pd.read_csv('/Users/atikarnpattamavichaiporn/Downloads/SG.csv')
```

In [317]:
```
SG
```

Out[317]:

	city	year	avg_temp	global_avg_temp
0	Singapore	1825	26.43	8.39
1	Singapore	1826	NaN	8.36
2	Singapore	1827	NaN	8.81
3	Singapore	1828	NaN	8.17
4	Singapore	1829	NaN	7.94
...	...	...	...	...
184	Singapore	2009	27.47	9.51
185	Singapore	2010	27.60	9.70
186	Singapore	2011	27.28	9.52
187	Singapore	2012	27.30	9.51
188	Singapore	2013	27.60	9.61

189 rows × 4 columns

In [318]:
```
# calculate Moving Average of SG average temperature over 20 years
SG['SG_MA_20'] = SG.avg_temp.rolling(20, min_periods=1).mean()
```

Import Kuala Lumpur Data and Calculate Moving Average

In [319]:
```
KL = pd.read_csv('/Users/atikarnpattamavichaiporn/Downloads/KL.csv')
```

In [320]:
```
KL
```

Out[320]:

	city	year	avg_temp	global_avg_temp
0	Kuala Lumpur	1825	26.46	8.39
1	Kuala Lumpur	1826	NaN	8.36
2	Kuala Lumpur	1827	NaN	8.81
3	Kuala Lumpur	1828	NaN	8.17
4	Kuala Lumpur	1829	NaN	7.94
...	...	...	...	...
184	Kuala Lumpur	2009	27.47	9.51
185	Kuala Lumpur	2010	27.69	9.70
186	Kuala Lumpur	2011	27.27	9.52
187	Kuala Lumpur	2012	27.36	9.51
188	Kuala Lumpur	2013	27.80	9.61

189 rows × 4 columns

In [321]:
```
# calculate Moving Average of KL average temperature over 20 years
KL['KL_MA_20'] = KL.avg_temp.rolling(20, min_periods=1).mean()
```

5. Visualize and compare the average temperature trend of each city vs. global average temperature

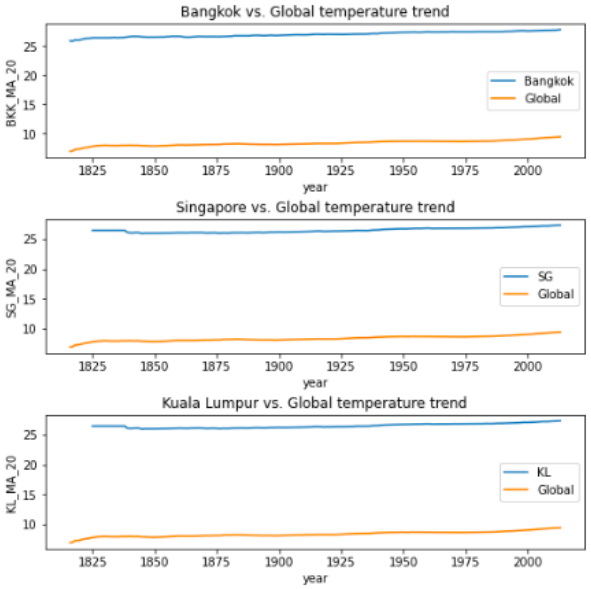
- Singapore and Kuala Lumpur have similar trends like Bangkok, foreseeing some upward trend of temperature from 1816 to 2013. Both cities also have the same gap of temperature differences between the city temperature and global temperature, which is around 20 degrees.

Compare BKK, SG, and KL trend with Global trend

In [322]:
```
fig, axs = plt.subplots(3, constrained_layout=True, figsize=(7,7))
ax1 =sns.lineplot(data=SG, y='SG_MA_20', x='year', ax=axs[1])
ax1 = sns.lineplot(data=BKK, y='global_MA_20', x='year', ax=axs[1])
axs[1].legend(['SG', 'Global'])
ax1.title.set_text('Singapore vs. Global temperature trend')

ax0 = sns.lineplot(data=BKK, y='BKK_MA_20', x='year', ax=axs[0])
ax0 = sns.lineplot(data=BKK, y='global_MA_20', x='year', ax=axs[0])
axs[0].legend(['Bangkok', 'Global'])
ax0.title.set_text('Bangkok vs. Global temperature trend')

ax2 = sns.lineplot(data=KL, y='KL_MA_20', x='year', ax=axs[2])
ax2 = sns.lineplot(data=BKK, y='global_MA_20', x='year', ax=axs[2])
axs[2].legend(['KL', 'Global'])
ax2.title.set_text('Kuala Lumpur vs. Global temperature trend')
```



6. Find and make the heatmap chart of correlation of Bangkok data

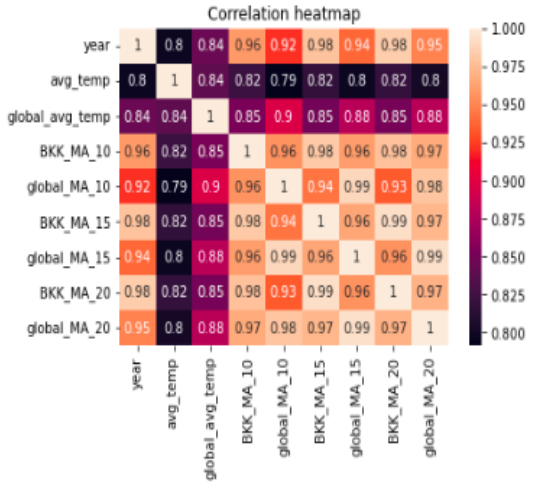
- From the chart, the closer the correlation coefficient is to 1, the higher the correlation between those 2 columns.
- From the correlation coefficient in the chart, all columns are highly correlated to other columns, all coefficients are higher than or equal to 0.8

Correlation of BKK dataset

In [323]:
```
BKK_corr = BKK.corr()
```

In [324]:
```
sns.heatmap(BKK_corr, annot=True)
plt.title('Correlation heatmap')
```

Out[324]:
```
Text(0.5, 1.0, 'Correlation heatmap')
```



7. Predict Bangkok temperature from Global temperature using Linear Regression model

Predict BKK temp based on Global temp

In [325]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import numpy as np
```

In [347]:
```
x = np.array(BKK['global_avg_temp'])
y = np.array(BKK['avg_temp'])
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

In [343]:
```
BKK=BKK.dropna(how='any')
```

In [344]:
```
BKK.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 190 entries, 0 to 197
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   city             190 non-null    object 
 1   year             190 non-null    int64  
 2   avg_temp         190 non-null    float64
 3   global_avg_temp  190 non-null    float64
 4   BKK_MA_10        190 non-null    float64
 5   global_MA_10     190 non-null    float64
 6   BKK_MA_15        190 non-null    float64
 7   global_MA_15     190 non-null    float64
 8   BKK_MA_20        190 non-null    float64
 9   global_MA_20     190 non-null    float64
dtypes: float64(8), int64(1), object(1)
memory usage: 16.3+ KB
```

In [348]:
```
model = LinearRegression()
model.fit(x_train.reshape(-1,1), y_train)
```

Out[348]:
```
LinearRegression()
```

In [365]:
```
bkk_temp = np.array([12]).reshape(-1,1)
print(model.predict(bkk_temp))
```

```
[30.27194125]
```

In [366]:
```
y_pred = model.predict(x_test.reshape(-1,1))
```

In [367]:
```
BKK_temp_pred = pd.DataFrame({'Global Temp': x_test, 'Bangkok Temp': y_pred})
```

a) Importing the relevant libraries need for implementing Linear Regression prediction
b) Specity global_avg_temp as x and Bangkok avg_temp as y, then proceed to split the data to x_train, x_test, y_train, y_test
c) Drop NA values presented since NAN values are presented in the data making the model unable to train the x_train data
d) Predict the temp:
    i) Specify the bkk_temp and plug into model.predict
    ii) Plug x_test into model.predict to make prediction stored the y_value as y_pred, then create data frame of x_test (Global Temp) and y_pred (Bangkok Temp)

In [364]:
```
BKK_temp_pred
```

Out[364]:

	Global Temp	Bangkok Temp
0	9.29	27.885559
1	9.32	27.911977
2	8.52	27.207510
3	8.32	27.031393
4	8.53	27.216316
5	7.63	26.423790
6	8.01	26.758412
7	7.80	26.573490
8	8.53	27.216316
9	8.75	27.410044
10	8.34	27.049005
11	8.38	27.084228
12	8.02	26.767218
13	9.20	27.806307
14	8.18	26.908112
15	8.85	27.498103
16	8.63	27.304374
17	7.97	26.723189
18	8.12	26.855276
19	8.29	27.004976
20	8.21	26.934529
21	7.98	26.731995
22	8.73	27.392433
23	8.57	27.251539
24	8.07	26.811247
25	8.63	27.304374
26	8.80	27.454073
27	8.17	26.899306
28	8.43	27.128257