Udacity First Project – Explore Weather Trends

My goal is to create a visualization and prepare a write up describing the similarities and differences between global temperature trends and temperature trends in the closest big city to where you live. To do this, I follow the steps below:

- **Extract the data** from the database. There's a workspace in the next section that is connected to a database. I export the temperature data for the world as well as for the closest big city to where you live. You can find a list of cities and countries in the `city_list` table. To interact with the database, I write a SQL queries as shown below.

```
SELECT * FROM city_list where city = 'Ankara' and
 country = 'Turkey';

SELECT * FROM city_data WHERE city = 'Ankara';

SELECT * FROM global_data;
```

- **Download csv file for each query**. All these files are listed below.
  - city_data_ankara.csv
  - city_list_ankara_turkey.csv
  - global_data.csv
- **Open Jupyter Notebook** to handle with this process.
- **Create** P1_Explore_Weather_Trends.ipynb as **Python 3**.
- **Import** all relevant and essential libraries used in this process.

```
# Importing essential packages used in this Project
import pandas as pd # import pandas for analyzing data
import numpy as np # import numpy for facilitating the process of calculation
import seaborn as sns # import seaborn for beautifing well-defined graph
import matplotlib.pyplot as plt # import matplotlib.pyplot for devising graph
from sklearn.linear_model import LinearRegression # import LinearRegression for drawing the line into the graph
```

- **Load all these csv files** and **assign** them to each relevant **dataframe**.

```
# Loading all datasets downloaded from platform
df_ankara_data = pd.read_csv("city_data_ankara.csv")
df_ankara_data_information = pd.read_csv("city_list_ankara_turkey.csv")
df_global_data = pd.read_csv("global_data.csv")
```

- **Determine** whether **each data frame** has value or not.

```
# Exloring df_ankara_data dataset
df_ankara_data.head(15)
```

	year	city	country	avg_temp
0	1755	Ankara	Turkey	9.62
1	1756	Ankara	Turkey	10.99
2	1757	Ankara	Turkey	10.82
3	1758	Ankara	Turkey	8.33
4	1759	Ankara	Turkey	9.93
5	1760	Ankara	Turkey	9.28
6	1761	Ankara	Turkey	NaN
7	1762	Ankara	Turkey	NaN
8	1763	Ankara	Turkey	NaN
9	1764	Ankara	Turkey	NaN
10	1765	Ankara	Turkey	NaN
11	1766	Ankara	Turkey	NaN
12	1767	Ankara	Turkey	NaN
13	1768	Ankara	Turkey	NaN
14	1769	Ankara	Turkey	NaN

```
# Exloring df_ankara_data_information dataset
df_ankara_data_information.head(1)
```

	city	country
0	Ankara	Turkey

```
# Exloring df_global_data dataset
df_global_data.head(15)
```

	year	avg_temp
0	1750	8.72
1	1751	7.98
2	1752	5.78
3	1753	8.39
4	1754	8.47
5	1755	8.36
6	1756	8.85
7	1757	9.02
8	1758	6.74
9	1759	7.99
10	1760	7.19
11	1761	8.77
12	1762	8.61
13	1763	7.50
14	1764	8.40

- **Eliminate** null values from **df_ankara_data** and check if null value is appeared or not.

```
#Elimainate cells with no data in df_ankara_data
df_ankara_data_complete = df_ankara_data.dropna()
```

```
# Exloring df_ankara_data_complete dataset
df_ankara_data_complete.head(15)
```

	year	city	country	avg_temp
0	1755	Ankara	Turkey	9.62
1	1756	Ankara	Turkey	10.99
2	1757	Ankara	Turkey	10.82
3	1758	Ankara	Turkey	8.33
4	1759	Ankara	Turkey	9.93
5	1760	Ankara	Turkey	9.28
19	1774	Ankara	Turkey	11.01
20	1775	Ankara	Turkey	11.22
21	1776	Ankara	Turkey	10.10
22	1777	Ankara	Turkey	10.08
23	1778	Ankara	Turkey	10.74
24	1779	Ankara	Turkey	10.82
25	1780	Ankara	Turkey	10.97
26	1781	Ankara	Turkey	10.47
27	1782	Ankara	Turkey	9.96

- Because both global and Ankara data **csv files** have **avg_temp** column, I changed it as a **relevant name with respect to its relevant file** for calcualting **moving averange** having 10 window size. Moreover, as **all values** are the same as **avg_temp**, I defined **None** for assignment process.

```
# disable chained assignments
pd.options.mode.chained_assignment = None

# Calculate moving averages by using rolling()
df_ankara_data_complete["ankara_moving_avg"] = df_ankara_data_complete["avg_temp"].rolling(window=10).mean()
df_global_data["global_moving_avg"] = df_global_data["avg_temp"].rolling(window=10).mean()

```

- **Create a new data frame** having year, ankara_moving_avg and lastly global_moving_avg. **Eliminate null values** in this new data frame. **Show** it.

```
# Combining columns to create a new DataFrame
df = pd.DataFrame({'year': df_ankara_data_complete["year"],
                   'ankara_moving_avg': df_ankara_data_complete["ankara_moving_avg"],
                   'global_moving_avg': df_global_data["global_moving_avg"],
                  })
# Elimainate cells with no data in df
df = df.dropna()
# Show df
print(df)
```

```
      year  ankara_moving_avg  global_moving_avg
22  1777.0             10.138              7.898
23  1778.0             10.250              7.970
24  1779.0             10.233              8.007
25  1780.0             10.248              8.100
```

- **Reshape this data frame** by using **melt function** and **Determine** year **as id, ankara_moving_avg and global_moving_avg** as **variable** and lastly **their values** as shown **observation**.

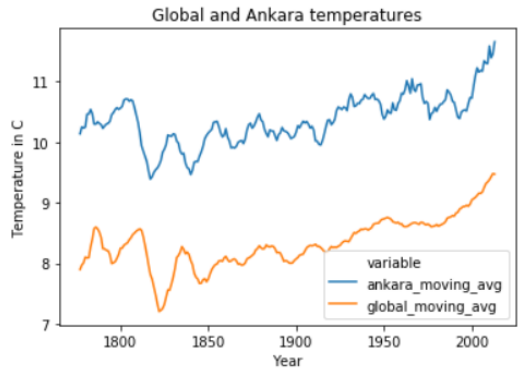
```
# Thanks to melt(), we can reshape the data into a more computer-friendly form using Pandas in Python
df_melt = pd.melt(df,
            id_vars=["year"],
            value_vars=["ankara_moving_avg", "global_moving_avg"],
            value_name="observation")

# Show df_melt
print(df_melt)
```

```
      year           variable  observation
0   1777.0  ankara_moving_avg       10.138
1   1778.0  ankara_moving_avg       10.250
2   1779.0  ankara_moving_avg       10.233
```

- **Show the graph** by using **this reshaped data frame**

```
p = sns.lineplot(data=df_melt.query('variable == "ankara_moving_avg" or variable == "global_moving_avg"')
                 ,x="year"
                 ,y="observation"
                 ,hue="variable")
p.set(xlabel='Year', ylabel='Temperature in C',title='Global and Ankara temperatures')
```

```
[Text(0, 0.5, 'Temperature in C'),
 Text(0.5, 0, 'Year'),
 Text(0.5, 1.0, 'Global and Ankara temperatures')]
```



- **Determine the observation** based on a graph shown above.
  - **First Observation** is determined whether Ankara hotter or colder than the global average and show how it has changed over time?
    - **Calculate the temperature difference** between the average temperature of both Ankara and Global. Then, Calculating moving

averange for temperature difference which has been done before. Add all two values into the data frame.

```
# To find the temperature changes between Ankara and Global ,
# firstly substract them and the result named for diff_temp to put in the df.
df['diff_temp'] = df_ankara_data_complete["avg_temp"]-df_global_data["avg_temp"]
```

```
# Calculating moving average in this column named for diff_temp
df['diff_temp_ma'] = df['diff_temp'].rolling(window=10).mean()
```

```
# show data set
df
```

	year	ankara_moving_avg	global_moving_avg	diff_temp	diff_temp_ma
22	1777.0	10.138	7.898	1.89	NaN
23	1778.0	10.250	7.970	2.52	NaN

- **Use describe()** function to get more detail information of this data frame.

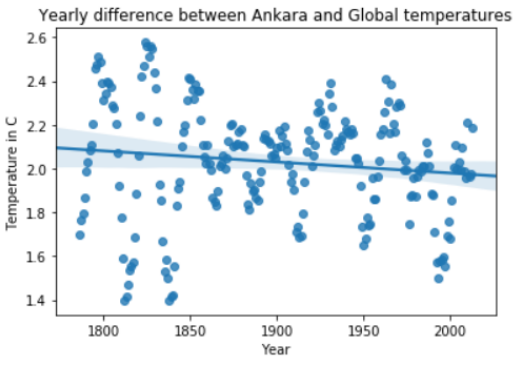
```
# use describe() to get more information about this data set
df.describe()
```

	year	ankara_moving_avg	global_moving_avg	diff_temp	diff_temp_ma
count	237.000000	237.000000	237.000000	237.000000	228.000000
mean	1895.000000	10.372068	8.335485	2.035274	2.030544
std	68.560193	0.411419	0.416336	0.661632	0.256548
min	1777.000000	9.389000	7.203000	-0.280000	1.395000
25%	1836.000000	10.118000	8.071000	1.650000	1.875500
50%	1895.000000	10.343000	8.279000	2.100000	2.057000
75%	1954.000000	10.642000	8.635000	2.460000	2.187250
max	2013.000000	11.656000	9.480000	3.840000	2.580000

- **Draw Graph**

```
p = sns.regplot(data=df,x="year",y="diff_temp_ma")
p.set(xlabel='Year', ylabel='Temperature in C',title='Yearly difference between Ankara and Global temperatures')
```

```
[Text(0, 0.5, 'Temperature in C'),
 Text(0.5, 0, 'Year'),
 Text(0.5, 1.0, 'Yearly difference between Ankara and Global temperatures')]
```

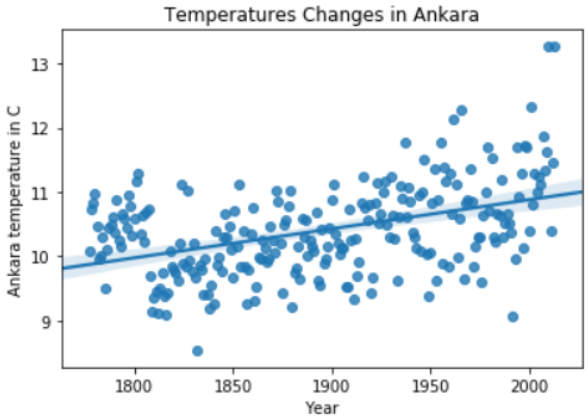


The result of Observation 1 : During 300 years, Ankara was hotter than the global averange approximately 2.03 coming from diff_temp column based on mean row. Moreover, its temperature ratio comparing to global averange was slightly getting lower year by year from starting point(nearly 2,1) to end point(nearly 1,9)

- **Second Observation and Third Observation** are shown what the general trend looks like and determine both the temperature of Ankara and Global are getting hotter or not?
  - In order to look through Ankara Graph firstly, first add **averange temperature of Ankara** into the dataframe and **draw the graph** be using year and this temperature.

```
# add averange temperature of Ankara into df
df["ankara_temp"] = df_ankara_data_complete["avg_temp"]

# Draw a graph to show Ankara temperature trend
p = sns.regplot(data=df,x="year",y="ankara_temp")
p.set(xlabel='Year', ylabel='Ankara temperature in C',title='Temperatures Changes in Ankara')
```

```
[Text(0, 0.5, 'Ankara temperature in C'),
 Text(0.5, 0, 'Year'),
 Text(0.5, 1.0, 'Temperatures Changes in Ankara')]
```



  - In order to find slope of the line, first **reshape of year** part of this graph.

```
# Reshaping year part for LinearRegression to calculting slope
year_reshape = df['year']
year_reshape = year_reshape.values.reshape(-1, 1)
year_reshape
```

```
array([[1777.],
       [1778.],
       [1779.],
       [1780.],
       [1781.],
```

Determine y-axis as averange temperature of Ankara and determine a Linear Regression by using year as x-axis and y-axis for averange temperature of Ankara.

```
# finding y part as y_ankara_temp
y_ankara_temp = df['ankara_temp']

model = LinearRegression()
model.fit(year_reshape, y_ankara_temp)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

- **Show slope**

```
# find slope -> https://intellipaat.com/blog/what-is-linear-regression/
print("Ankara Slope Temperature {}".format(model.coef_))
```

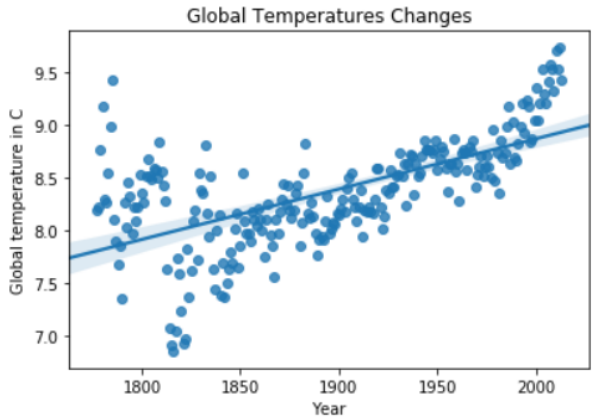
```
Ankara Slope Temperature [0.00451744]
```

The result of Observation 2 : During 300 years, Ankara has been getting hotter by approximately 0.0045 ratio per pear. Its trend is going up.

- In order to look through Global Temperature lastly, The same process has been done for it by using the averange temperature of Global.

```
# add averange temperature of Global into df
df["global_temp"] = df_global_data["avg_temp"]

# Draw a graph to show Global temperature trend
p = sns.regplot(data=df,x="year",y="global_temp")
p.set(xlabel='Year', ylabel='Global temperature in C',title='Global Temperatures Changes')
```

```
[Text(0, 0.5, 'Global temperature in C'),
 Text(0.5, 0, 'Year'),
 Text(0.5, 1.0, 'Global Temperatures Changes')]
```



```
# finding y part as y_global_temp
y_global_temp = df['global_temp']

model = LinearRegression()
model.fit(year_reshape, y_global_temp)

# find slope -> https://intellipaat.com/blog/what-is-linear-regression/
print("Global Slope Temperature {}".format(model.coef_))
```

```
Global Slope Temperature [0.0047598]
```

The result of Observation 3 : During 300 years, Global Temperature has been getting hotter by approximately 0.0047 ratio per pear. Its trend is going up.

- **Fourth Observation and Fifth Observation** are shown how the trend has changed in the last 10 years?
  - **Using Filter** to shown only the last 10 years for year's column and then show new data frame.

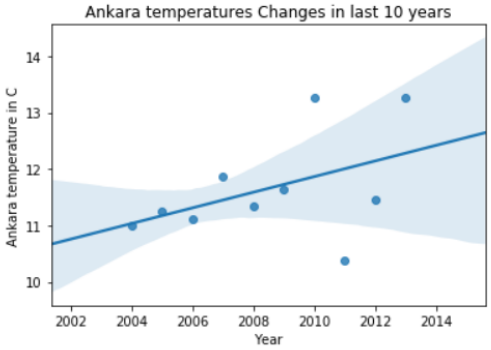
```
# Using filter to get years greater than 2003
df_10 = df[df.year > 2003]
df_10
```

	year	ankara_moving_avg	global_moving_avg	diff_temp	diff_temp_ma	ankara_temp	global_temp
249	2004.0	11.155	9.156	1.70	1.999	10.99	9.29
250	2005.0	11.186	9.153	2.04	2.033	11.24	9.20
251	2006.0	11.169	9.176	1.71	1.993	11.12	9.41
252	2007.0	11.343	9.249	2.30	2.094	11.87	9.57
253	2008.0	11.304	9.315	1.80	1.989	11.33	9.53
254	2009.0	11.298	9.343	2.31	1.955	11.63	9.32
255	2010.0	11.586	9.378	3.57	2.208	13.27	9.70
256	2011.0	11.393	9.427	0.86	1.966	10.39	9.53
257	2012.0	11.457	9.480	1.72	1.977	11.45	9.73
258	2013.0	11.656	9.471	3.84	2.185	13.27	9.43

  - Draw a graph for his data frame as shown above for Ankara.

```
# Ankara Graph
p = sns.regplot(data=df_10,x="year",y="ankara_temp")
p.set(xlabel='Year', ylabel='Ankara temperature in C',title='Ankara temperatures Changes in last 10 years')
```

```
[Text(0, 0.5, 'Ankara temperature in C'),
 Text(0.5, 0, 'Year'),
 Text(0.5, 1.0, 'Ankara temperatures Changes in last 10 years')]
```



  - Find Slope fort his graph

```python
# Calculating slope of Ankara Temperature
x_year = df_10['year']
x_year = x_year.values.reshape(-1, 1)
y_ankara_temp = df_10['ankara_temp']
model = LinearRegression().fit(x_year, y_ankara_temp)
print("Ankara Slope Temperature {}".format(model.coef_))
```

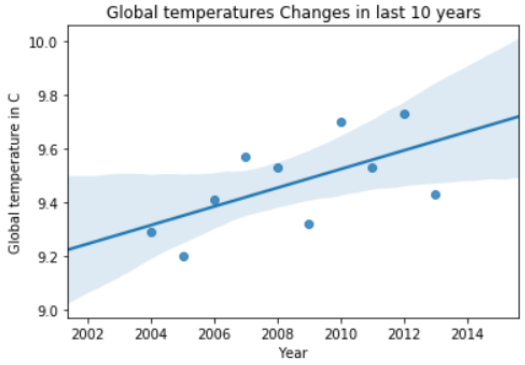
```
Ankara Slope Temperature [0.13842424]
```

The result of Observation 4 : During 10 years, Ankara has been getting hotter by approximately 0.13 ratio per pear. Its trend is going up.

- The same process has done for Global Temperature

```python
# Global Temperature Graph
p = sns.regplot(data=df_10,x="year",y="global_temp")
p.set(xlabel='Year', ylabel='Global temperature in C',title='Global temperatures Changes in last 10 years')
```

```
[Text(0, 0.5, 'Global temperature in C'),
 Text(0.5, 0, 'Year'),
 Text(0.5, 1.0, 'Global temperatures Changes in last 10 years')]
```




```python
# Calculating slope of Global Temperature
x_year = df_10['year']
x_year = x_year.values.reshape(-1, 1)
y_ankara_temp = df_10['global_temp']
model = LinearRegression().fit(x_year, y_ankara_temp)
print("Global Slope Temperature {}".format(model.coef_))
```

```
Global Slope Temperature [0.03484848]
```

The result of Observation 4 : During 10 years, Ankara has been getting hotter by approximately 0.034 ratio per pear. Its trend is going up.

The resource Link : [Project Link]