Udacity Data Analyst Nanodegree

Project1: Explore Weather Trends

By

Rajesh Dulam

Introduction:

Analyzing and comparing global temperature trends to Seattle, USA. In this project we will explore the data, describe similarities and differences between the city and global temperature trends.

Extracting data from database:

Below are the following SQL commands used to extract city and global data into a .csv format. A combined table of city and global data is also extracted for analysis purposes.

Select global data	`select *`
	`from global_data`
	`order by year`

Select my city data and order by year	`select *`
	`from city_data`
	`where country = 'United States' and city = 'Seattle'`
	`order by year`

Extract combined data by joining global and city tables. Removing null values from city data to have complete data points.	`select c.year,c.city,c.country,c.avg_temp as city_avg_temp,`
	`g.avg_temp as global_avg_temp`
	`from city_data as c`
	`join global_data as g`
	`on c.year = g.year`
	`where c.city = 'Seattle' and c.country = 'United States' and c.avg_temp is not null`
	`order by c.year`

Exploring weather trends:

The data analysis will be conducted using python pandas on the data extracted from joining two tables above.

- Combined data will contain both city and global temperatures.

```
temp_data = pd.read_csv('combined_data.csv')
temp_data.head()
```

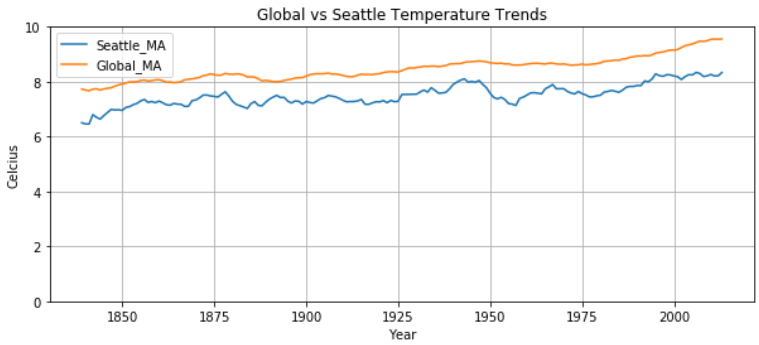
	year	city	country	city_avg_temp	global_avg_temp
0	1828	Seattle	United States	7.13	8.17
1	1829	Seattle	United States	6.80	7.94
2	1832	Seattle	United States	3.52	7.45
3	1833	Seattle	United States	7.48	8.01
4	1834	Seattle	United States	7.10	8.15

- Line chart: Comparing city data with global data. To get smooth lines while plotting temperature data, moving averages is applied to both city and global temperature and appended to data frame.

```
temp_data['Seattle_MA'] = temp_data['city_avg_temp'].rolling(window=10).mean()

temp_data['Global_MA'] = temp_data['global_avg_temp'].rolling(window=10).mean()
temp_data.tail()
```

	year	city	country	city_avg_temp	global_avg_temp	Seattle_MA	Global_MA
178	2009	Seattle	United States	8.02	9.51	8.212	9.493
179	2010	Seattle	United States	8.25	9.70	8.265	9.543
180	2011	Seattle	United States	7.35	9.52	8.210	9.554
181	2012	Seattle	United States	8.08	9.51	8.215	9.548
182	2013	Seattle	United States	9.95	9.61	8.336	9.556

```
#Create figure and axes object
fig,ax = plt.subplots(nrows=1,ncols=1,figsize=(10,4))

#Plot temperature data
temp_data.plot(kind='line', x='year',y=['Seattle_MA','Global_MA'], ax=ax)

ax.set(title='Global vs Seattle Temperature Trends',xlabel='Year',ylabel='Celcius',ylim=(0,10))
ax.grid(True)
ax.legend(loc='upper left')
```



- Exploring data, below are some interesting findings:

1.) From line graph, Seattle is consistently cooler than global temperature.

2.) The standard deviation, mean, min and max of the temperature data.

```
temp_data[['city_avg_temp','global_avg_temp']].describe(percentiles=[])
```

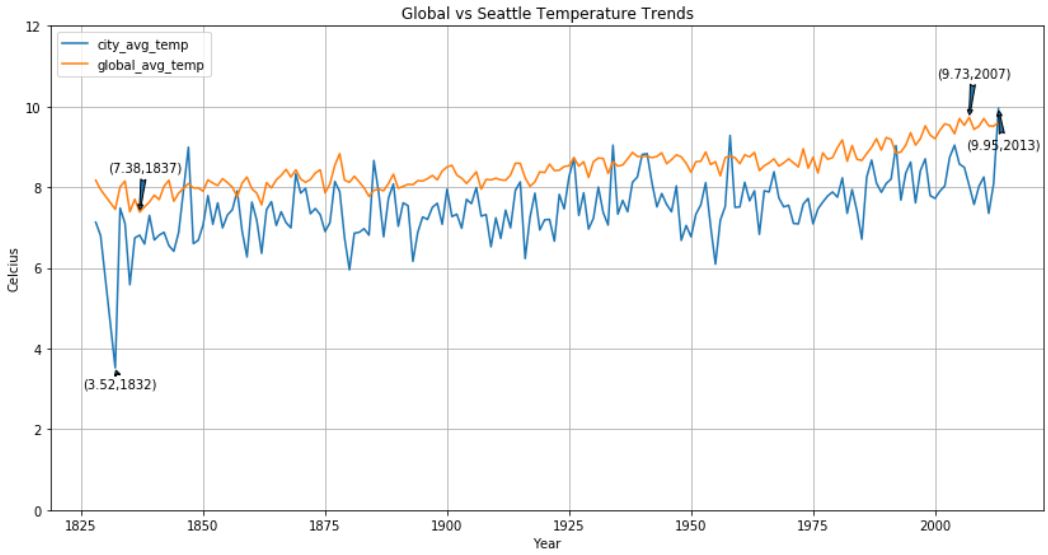
	city_avg_temp	global_avg_temp
count	183.000000	183.000000
mean	7.501366	8.481311
std	0.739375	0.497103
min	3.520000	7.380000
50%	7.500000	8.440000
max	9.950000	9.730000

Even though Seattle average temperature is cooler than global average temperature over the centuries, the highest recorded average temperature in Seattle is greater than global highest recorded average temperature.

Seattle city's standard deviation is almost a quarter higher than global avg standard deviation. This implies that people in Seattle experience more temperature fluctuations than global avg temperature, which can be seen in next line graph.

3.) It would be interesting to see when min and max temperatures occurred for both Seattle and global data to see if there is any correlation between them.

```
#Create figure and axes object
fig1,ax1 = plt.subplots(nrows=1,ncols=1,figsize=(14,7))

#Plot temperature data without moving averages because we want find the highest and lowest values
temp_data.plot(kind='line', x='year',y=['city_avg_temp','global_avg_temp'], ax=ax1)

ax1.set(title='Global vs Seattle Temperature Trends',xlabel='Year',ylabel='Celcius',ylim=(0,12))
ax1.grid(True)
ax1.legend(loc='upper left')
#Get max and min for Seattle and Global
Seattle_max = temp_data['city_avg_temp'].idxmax()
Seattle_min = temp_data['city_avg_temp'].idxmin()

Global_max = temp_data['global_avg_temp'].idxmax()
Global_min = temp_data['global_avg_temp'].idxmin()

#Annotating min and max values for Seattle
for m,offset in zip([Seattle_max,Seattle_min],[-1,-.5]):
    year = temp_data.loc[m,'year']
    temp = temp_data.loc[m, 'city_avg_temp']
    ax1.annotate('('+str(temp)+','+str(year)+')', xy=(year,temp),xytext=(year+1,temp+offset),
                 ha='center',arrowprops = dict(arrowstyle='fancy'))
#Annotating min and max values for Global
for m,offset in zip([Global_max,Global_min],[1,1]):
    year = temp_data.loc[m,'year']
    temp = temp_data.loc[m, 'global_avg_temp']
    ax1.annotate('('+str(temp)+','+str(year)+')', xy=(year,temp),xytext=(year+1,temp+offset),
                 ha='center',arrowprops = dict(arrowstyle='fancy'))
```



Lowest temperatures are recorded in 1832 in Seattle and 1837 in global. Similarly, highest temperatures are recorded 2013 and 2017 in Seattle and global, respectively. Above graph proves that the world and Seattle are getting hotter. There is a strong correlation between Seattle and Global temperature data.

4.) We understand that temperature is increasing both in Seattle and Global, but we can also know the rate of change by calculating the slope of the curve.

-First, sort data by year

```
#Sort data by year
temp_data = temp_data.sort_values(['year'])
temp_data.tail()
```

	year	city	country	city_avg_temp	global_avg_temp	Seattle_MA	Global_MA
178	2009	Seattle	United States	8.02	9.51	8.212	9.493
179	2010	Seattle	United States	8.25	9.70	8.265	9.543
180	2011	Seattle	United States	7.35	9.52	8.210	9.554
181	2012	Seattle	United States	8.08	9.51	8.215	9.548
182	2013	Seattle	United States	9.95	9.61	8.336	9.556

- To calculate slope, we select some random points. Let's take points for every 25 years

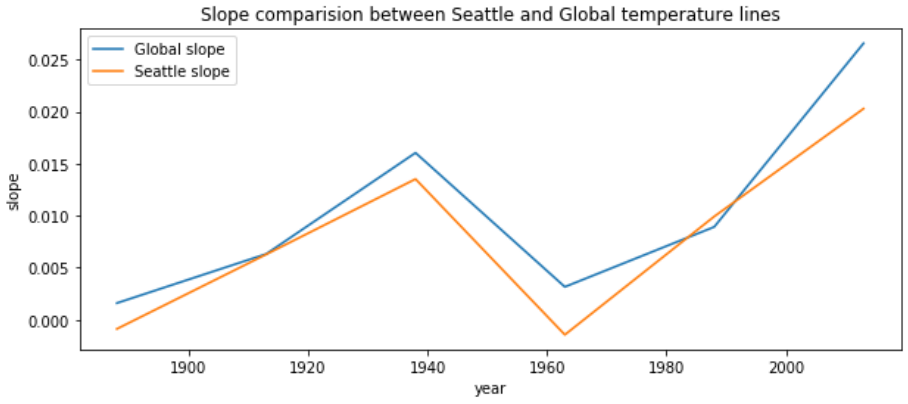
Ex: 2013,1988,1963,1938 (2013 is the most recent data point)

-Slope m = (y2-y1)/x2-x1). Since x2-x1, here is year and we already establish that the difference is 25 years.

Therefore, m = (y2-y1)/25

```
#Function to calcalte slope. Input dataframe must have year and temperature columns only (2 columns)
def cal_slope(dataframe):
    y7 =float(dataframe.iloc[-1:,1:2].values)
    y6 =float(dataframe.iloc[-26:-25,1:2].values)
    y5 =float(dataframe.iloc[-51:-50,1:2].values)
    y4 =float(dataframe.iloc[-76:-75,1:2].values)
    y3 =float(dataframe.iloc[-101:-100,1:2].values)
    y2 =float(dataframe.iloc[-126:-125,1:2].values)
    y1 =float(dataframe.iloc[-151:-150,1:2].values)

    return [(y2-y1)/25,(y3-y2)/25,(y4-y3)/25,(y5-y4)/25,(y6-y5)/25,(y7-y6)/25]

#Get slopes
seattle_slope = cal_slope(temp_data[['year','Seattle_MA']])
global_slope = cal_slope(temp_data[['year','Global_MA']])
```

```
slope_years=[1888,1913,1938,1963,1988,2013]
fig3,ax3 = plt.subplots(figsize=(10,4))

ax3.plot(slope_years,global_slope,label ='Global slope')
ax3.plot(slope_years,seattle_slope,label ='Seattle slope')
ax3.legend(loc='upper left')
ax3.set(title='Slope comparision between Seattle and Global temperature lines',xlabel='year',ylabel='slope')
```



-Seattle and global temperature's rate of change increased during 1960's.

-Rate of change for Seattle temperature is consistent from 1960's.
-Rate of change for global temperature keeps increasing 1960's and 1980's. The global rate of change in 1980's is alarming as it incremented from 1960's
-From the above graph, we can conclusively witness the effects of global warming in the last 25-50 years.

References:

1.) Pandas_cookbook by Theodore Petrou
2.) https://stackoverflow.com/questions/9538525/calculating-slopes-in-numpy-or-scipy