Extract Data Using SQL From Udacity Workspace

The data is available on udacity workspace. Only students from this course can directly download from it. Below codes were used to extract the data for this project. The extracted data are available on github.

```
-- download global data
SELECT * FROM global_data;

-- download Berlin data
SELECT *
FROM city_data
WHERE city = 'Berlin';
```

Set up

In [0]:

```
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import urllib

PROJECT_ROOT_DIR = '.'
CHAPTER_ID = 'WEATHER'
IMAGES_PATH = os.path.join(PROJECT_ROOT_DIR, CHAPTER_ID, 'IMAGES')
DATA_PATH = os.path.join(PROJECT_ROOT_DIR, CHAPTER_ID, 'DATASETS')
BERLIN_URL = 'https://raw.githubusercontent.com/AilingLiu/Data_Analyst_NanoDegree_Udacity/master/Project_Explore_Weather_Trends/data/berlin_temperature.csv'
GLOBAL_URL = 'https://raw.githubusercontent.com/AilingLiu/Data_Analyst_NanoDegree_Udacity/master/Project_Explore_Weather_Trends/data/global_temperature.csv'

if not os.path.isdir(IMAGES_PATH):
  os.makedirs(IMAGES_PATH)

if not os.path.isdir(DATA_PATH):
  os.makedirs(DATA_PATH)

#images
def save_fig(file_name, path=IMAGES_PATH, dpi=300, fmt='png'):
  file_path = os.path.join(path, file_name+'.'+fmt)
  plt.savefig(file_path, dpi=dpi, format=fmt)

#datasets
def fetch_data(file_name, data_link, path=DATA_PATH, fmt='csv'):
  file_path = os.path.join(path, file_name+'.'+fmt)
  urllib.request.urlretrieve(data_link,file_path)
  print('The data is downloaded in ', file_path)

def load_data(file_name, path = DATA_PATH, fmt='csv'):
  file_path=os.path.join(path, file_name+'.'+fmt)
  return pd.read_csv(file_path)
```

In [18]:

```
fetch_data('berlin_temp', BERLIN_URL)
fetch_data('global_temp', GLOBAL_URL)
```

```
The data is downloaded in  ./WEATHER/DATASETS/berlin_temp.csv
The data is downloaded in  ./WEATHER/DATASETS/global_temp.csv
```

In [19]:

```
berlin_temp = load_data('berlin_temp')
global_temp = load_data('global_temp')
berlin_temp.head()
```

Out[19]:

	year	city	country	avg_temp
0	1743	Berlin	Germany	6.33
1	1744	Berlin	Germany	10.36
2	1745	Berlin	Germany	1.43
3	1746	Berlin	Germany	NaN
4	1747	Berlin	Germany	NaN

In [0]:

```
global_temp.head()
```

Out[0]:

	year	avg_temp
0	1750	8.72
1	1751	7.98
2	1752	5.78
3	1753	8.39
4	1754	8.47

In [0]:

```
inds = np.where(berlin_temp.year <1750)[0] #drop the years from berlin that were not
 available in global record
berlin_temp = berlin_temp.drop(index=inds, axis=0)
```

Line chart

In [0]:

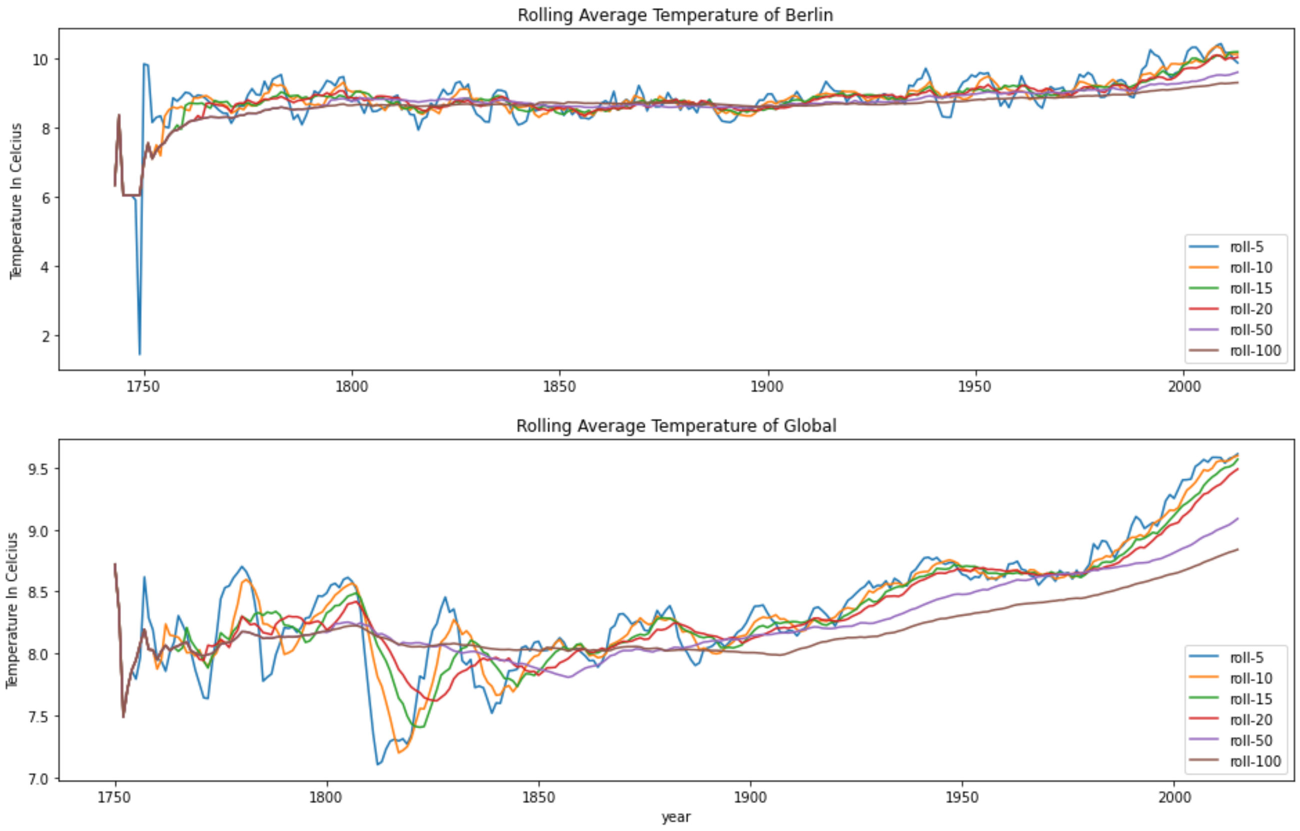
```
berlin_temp['roll_5y']=berlin_temp['avg_temp'].rolling(5, min_periods=1).mean()
berlin_temp['roll_10y']=berlin_temp['avg_temp'].rolling(10, min_periods=1).mean()
berlin_temp['roll_15y']=berlin_temp['avg_temp'].rolling(15, min_periods=1).mean()
berlin_temp['roll_20y']=berlin_temp['avg_temp'].rolling(20, min_periods=1).mean()
berlin_temp['roll_50y']=berlin_temp['avg_temp'].rolling(50, min_periods=1).mean()
berlin_temp['roll_100y']=berlin_temp['avg_temp'].rolling(100, min_periods=1).mean()

global_temp['roll_5y']=global_temp['avg_temp'].rolling(5, min_periods=1).mean()
global_temp['roll_10y']=global_temp['avg_temp'].rolling(10, min_periods=1).mean()
global_temp['roll_15y']=global_temp['avg_temp'].rolling(15, min_periods=1).mean()
global_temp['roll_20y']=global_temp['avg_temp'].rolling(20, min_periods=1).mean()
global_temp['roll_50y']=global_temp['avg_temp'].rolling(50, min_periods=1).mean()
global_temp['roll_100y']=global_temp['avg_temp'].rolling(100, min_periods=1).mean()
```

In [21]:
```
fig, axes = plt.subplots(2, 1, figsize=(16, 10))

berlin_temp.plot(x='year', y='roll_5y', kind='line', ax=axes[0], label='roll-5')
berlin_temp.plot(x='year', y='roll_10y', kind='line', ax=axes[0], label='roll-10')
berlin_temp.plot(x='year', y='roll_15y', kind='line', ax=axes[0], label='roll-15')
berlin_temp.plot(x='year', y='roll_20y', kind='line', ax=axes[0], label='roll-20')
berlin_temp.plot(x='year', y='roll_50y', kind='line', ax=axes[0], label='roll-50')
berlin_temp.plot(x='year', y='roll_100y', kind='line', ax=axes[0], label='roll-100')
axes[0].set(title='Rolling Average Temperature of Berlin', xlabel=None, ylabel='Tempe
rature In Celcius')
axes[0].legend(loc='lower right')

global_temp.plot(x='year', y='roll_5y', kind='line', ax=axes[1], label='roll-5')
global_temp.plot(x='year', y='roll_10y', kind='line', ax=axes[1], label='roll-10')
global_temp.plot(x='year', y='roll_15y', kind='line', ax=axes[1], label='roll-15')
global_temp.plot(x='year', y='roll_20y', kind='line', ax=axes[1], label='roll-20')
global_temp.plot(x='year', y='roll_50y', kind='line', ax=axes[1], label='roll-50')
global_temp.plot(x='year', y='roll_100y', kind='line', ax=axes[1], label='roll-100')
axes[1].set(title='Rolling Average Temperature of Global', ylabel='Temperature In Cel
cius')
axes[1].legend(loc='lower right')
save_fig('Rolling_Summary_of_Berlin_and_Global')
```



In [0]:
```
global_temp.loc[[59, 69], ['year', 'roll_10y']]
```

Out[0]:

	year	roll_10y
59	1809	8.297
69	1819	7.252

- How do the changes in your city’s temperatures over time compare to the changes in the global average?

Temperature of Berlin and Global were both increasing over the years. The slope of increase were much higher after 1950s.

- What does the overall trend look like? Is the world getting hotter or cooler? Has the trend been consistent over the last few hundred years?

It became hotter over the years, but there was exception at around 1810s. In the world, the average temperature was 8.3 in 1800s, then it dropped to 7.3 in 1810s. During this time, Berlin maintained at 8 degree.

In [0]:

```
rolling_cols = [col for col in global_temp if col.startswith('roll_')]

print('Global rolling average temperature summary: ')

global_temp[rolling_cols].describe()
```

```
Global rolling average temperature summary:
```

Out[0]:

	roll_5y	roll_10y	roll_15y	roll_20y	roll_50y	roll_100y
count	266.000000	266.000000	266.000000	266.000000	266.000000	266.000000
mean	8.358853	8.343519	8.328898	8.315215	8.255332	8.194518
std	0.491357	0.452525	0.418972	0.391266	0.305054	0.221627
min	7.108000	7.203000	7.408667	7.493333	7.493333	7.493333
25%	8.041000	8.045250	8.051333	8.050708	8.037600	8.034920
50%	8.320000	8.269000	8.252333	8.230750	8.155800	8.121600
75%	8.627000	8.637250	8.637833	8.640750	8.478750	8.312475
max	9.608000	9.594000	9.564667	9.486000	9.086800	8.838500

In [0]:

```
print('Berlin rolling average temperature summary: ')

berlin_temp[rolling_cols].describe()
```

```
Berlin rolling average temperature summary:
```

Out[0]:

	roll_5y	roll_10y	roll_15y	roll_20y	roll_50y	roll_100y
count	264.000000	264.000000	264.000000	264.000000	264.000000	264.000000
mean	8.912549	8.897517	8.882571	8.868628	8.812363	8.769417
std	0.509030	0.422682	0.379882	0.353316	0.251651	0.185368
min	7.920000	8.140000	8.140000	8.140000	8.140000	8.140000
25%	8.590000	8.607750	8.613500	8.623875	8.630578	8.679250
50%	8.849000	8.829000	8.825333	8.809000	8.745100	8.723363
75%	9.188500	9.063750	9.023500	8.987250	8.957600	8.805579
max	10.414000	10.339000	10.180000	10.087000	9.830000	9.830000

- What's the trend among all the rolling scale, 5 rolling average, 10 rolling average, etc.?

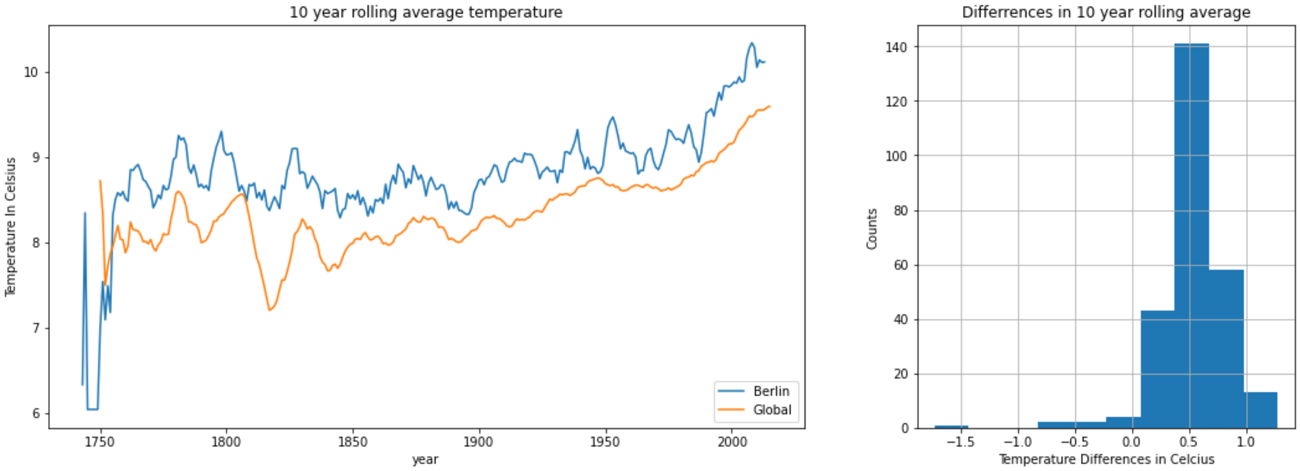
In both plots, the larger scale of rolling window, the smoother the line becomes. For example, the 5 year rolling average line in Berlin was much wavier compared to the global 5 year rolling average line. This was caused by the decreasing variances of the rolling average along with increasing the rolling window.

```
In [22]: fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(18, 6), gridspec_kw={'width_ratios': [2
         , 1]})

         berlin_x = berlin_temp.year.values
         global_x = global_temp.year.values

         ax1.plot(berlin_x, berlin_temp['roll_10y'], label='Berlin')
         ax1.plot(global_x, global_temp['roll_10y'], label='Global')
         ax1.legend(loc='lower right')
         ax1.set(title='10 year rolling average temperature', xlabel='year', ylabel='Temperatu
         re In Celsius')

         gl10y = global_temp[['year', 'roll_10y']]
         bl10y = berlin_temp[['year', 'roll_10y']]
         full10y = gl10y.merge(bl10y, on='year', how='outer', suffixes=('_global', '_berlin'))
         .sort_values(by='year')
         full10y['diff'] = full10y['roll_10y_berlin'] - full10y['roll_10y_global']
         full10y['diff'].hist(ax=ax2)
         ax2.set(title='Differences in 10 year rolling average', xlabel='Temperature Differen
         ces in Celcius', ylabel='Counts')
         save_fig('10_year_rolling_average_temperature')
```



- Is your city hotter or cooler on average compared to the global average? Has the difference been consistent over time?

The 10 year rolling average linechart shows Berlin is above global average in general. In general, Berlin is 0.5 degree more than the global temperature. The largest difference was at 1819, where the average rolling differences in the past decade was 1.28, being Berlin is 1.28 degree hotter than global.

In [0]:

```
full10y.reindex(full10y['diff'].abs().sort_values(ascending=False).index).head(10)
```

Out[0]:

	year	roll_10y_global	roll_10y_berlin	diff
1	1751	8.350	9.790	1.440
69	1819	7.252	8.534	1.282
68	1818	7.223	8.457	1.234
74	1824	7.653	8.841	1.188
76	1826	7.910	9.095	1.185
67	1817	7.203	8.372	1.169
75	1825	7.768	8.935	1.167
70	1820	7.322	8.474	1.152
65	1815	7.482	8.614	1.132
0	1750	8.720	9.830	1.110

In [0]:

```
```