Data Extraction and Exploration

- 3 Queries were used on the Udacity website to extract the data in .csv format:
  - select * from city_data;
  - select * from city_list;
  - select * from global_data;
- Then I used python jupyter notebook to read and explore the resulting .csv

```
In [91]: city_data = pd.read_csv('city_data.csv')
         city_list = pd.read_csv('city_list.csv')
         global_data = pd.read_csv('global_data.csv')
```

```
In [103]: city_data
```

Out[103]:

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

70792 rows × 4 columns

```
In [102]: city_list
```

Out[102]:

	city	country
0	Abidjan	Côte D'Ivoire
1	Abu Dhabi	United Arab Emirates
2	Abuja	Nigeria
3	Accra	Ghana
4	Adana	Turkey
...	...	...
337	Xuzhou	China
338	Yamoussoukro	Côte D'Ivoire
339	Yerevan	Armenia
340	Zagreb	Croatia
341	Zapopan	Mexico

342 rows × 2 columns

```
In [104]: global_data
```

Out[104]:

	year	avg_temp	global moving avg temp
0	1750	8.72	NaN
1	1751	7.98	NaN
2	1752	5.78	NaN
3	1753	8.39	NaN
4	1754	8.47	7.868
...	...	...	...
261	2011	9.52	9.578
262	2012	9.51	9.534
263	2013	9.61	9.570
264	2014	9.57	9.582
265	2015	9.83	9.608

266 rows × 3 columns

Data Creation

- Created moving average temperatures for Los Angeles and Global

- la_data['la moving avg temp'] = la_data.iloc[:,3].rolling(window=5).mean()
- global_data['global moving avg temp'] = global_data.iloc[:,1].rolling(window=5).mean()

- The moving average was calculated using pandas .rolling() method, where the “windows=“ is used to select how many rows of the average temperature column column I would like to select, then use the mean() method to get the averages among those rows of year

- Results:

```
In [94]: la_data
```

Out[94]:

	year	city	country	avg_temp	la moving avg temp
36607	1849	Los Angeles	United States	15.71	NaN
36608	1850	Los Angeles	United States	15.28	NaN
36609	1851	Los Angeles	United States	15.53	NaN
36610	1852	Los Angeles	United States	15.61	NaN
36611	1853	Los Angeles	United States	16.27	15.680
...	...	...	...	...	...
36767	2009	Los Angeles	United States	16.68	16.688
36768	2010	Los Angeles	United States	15.89	16.580
36769	2011	Los Angeles	United States	15.87	16.430
36770	2012	Los Angeles	United States	17.09	16.508
36771	2013	Los Angeles	United States	18.12	16.730

165 rows × 5 columns

```
In [96]: global_data
```

Out[96]:

	year	avg_temp	global moving avg temp
0	1750	8.72	NaN
1	1751	7.98	NaN
2	1752	5.78	NaN
3	1753	8.39	NaN
4	1754	8.47	7.868
...	...	...	...
261	2011	9.52	9.578
262	2012	9.51	9.534
263	2013	9.61	9.570
264	2014	9.57	9.582
265	2015	9.83	9.608

266 rows × 3 columns

Data Merging and Cleaning

- I used the .merge() method and performed a left join with the global_data and la_data tables, with the moving averages added
- I noticed that Los Angeles only has average temperature data from 1853 to 2013, so I selected only the rows with that range of years

-combined = global_data.merge(la_data, left_on='year', right_on='year', how='left')[103:-2], results:

```
In [109]: combined
```

Out[109]:

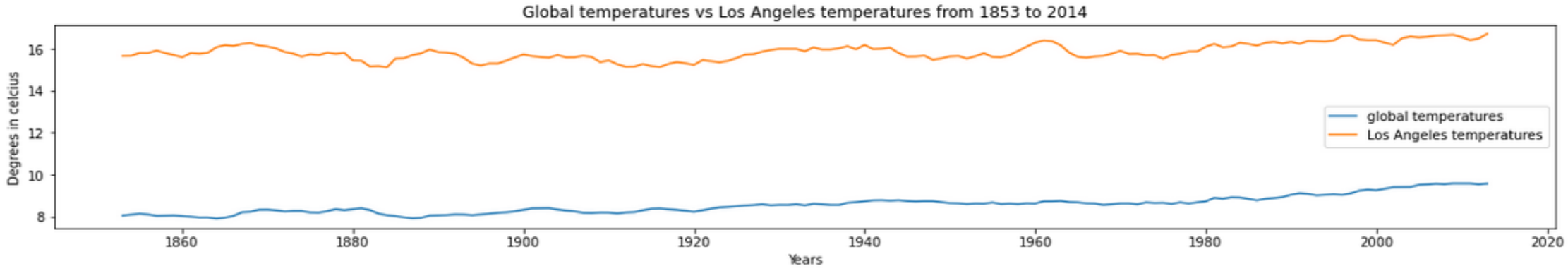
	year	avg_temp_x	global moving avg temp	city	country	avg_temp_y	la moving avg temp
103	1853	8.04	8.040	Los Angeles	United States	16.27	15.680
104	1854	8.21	8.086	Los Angeles	United States	15.74	15.686
105	1855	8.11	8.128	Los Angeles	United States	15.94	15.818
106	1856	8.00	8.092	Los Angeles	United States	15.52	15.816
107	1857	7.76	8.024	Los Angeles	United States	16.19	15.932
...	...	...	...	...	...	...	...
259	2009	9.51	9.580	Los Angeles	United States	16.68	16.688
260	2010	9.70	9.580	Los Angeles	United States	15.89	16.580
261	2011	9.52	9.578	Los Angeles	United States	15.87	16.430
262	2012	9.51	9.534	Los Angeles	United States	17.09	16.508
263	2013	9.61	9.570	Los Angeles	United States	18.12	16.730

161 rows × 7 columns

Data Visualization

- I used matplotlib.pyplot library visualize the Los Angeles' and Global's moving average temperature changes from 1853 to 2013,
- A line chart was chosen because it is the most appropriate for telling changes in values over time

In [101]:
```
plt.plot(combined['year'], combined['global moving avg temp'], label = 'global temperatures')
plt.plot(combined['year'], combined['la moving avg temp'], label = 'Los Angeles temperatures')
plt.title('Global temperatures vs Los Angeles temperatures from 1853 to 2014')
plt.xlabel('Years')
plt.ylabel('Degrees in celcius')
plt.legend()
plt.figure(figsize=(50, 50))
plt.show()
```




```
<Figure size 3600x3600 with 0 Axes>
```

Findings

- Los Angeles, from the years 1853 to 2013, is about 7.32 degrees Celsius hotter than the global average

```
In [10]: combined['la moving avg temp'].mean() -
         combined['global moving avg temp'].mean()

Out[10]: 7.321130434782601
```

- Los Angeles has a range in average temperature fluctuations of about 1.6 degrees Celsius, while the world has 1.69 degrees Celsius

```
In [111]: combined['la moving avg temp'].max() -
          combined['la moving avg temp'].min()

Out[111]: 1.5979999999999972
```

```
In [110]: combined['global moving avg temp'].max() -
          combined['global moving avg temp'].min()

Out[110]: 1.6879999999999935
```

- Los Angeles's hottest year on record was in 2013, while the coldest year was in 1880, the world's hottest year on record was in 2015, while the coldest year was in 1752.

```
In [24]: la_data.loc[la_data['avg_temp'].idxmax()]

Out[24]: year    2013
```

```
In [25]: la_data.loc[la_data['avg_temp'].idxmin()]

Out[25]: year    1880
```

```
In [22]: global_data.loc[global_data['avg_temp'].idxmax()]

Out[22]: year    2015.000
```

```
In [23]: global_data.loc[global_data['avg_temp'].idxmin()]

Out[23]: year    1752.00
```

- The temperatures all around the world have been steadily increasing, I believe that the hottest year on record would be 2020 if there was data available in this dataset.