Udacity

Data Analyst Nanodegree

Explore Weather Trends Project

1. INTRODUCTION

The Project is to study and compare the weather trends of the Delhi average temperature with the Global average
temperature. The Data has been extracted by the SQL query provided by the
Udacity Nanodegree program. The Extracted data consists of the both temperatures of the Delhi and Global from 1796 to 2013. The study provides detailed visualization of comparing the both average temperatures. From data analysis We can come to the conclusion about the cities and the global weather trends.

2. DATA SECTION

The Data is Extracted using the SQL query and downloaded as the CSV file to use in the
jupyter notebook.

2.1 Data Extraction from the SQL

The SQL Query to find the cities in India

SQL Query:

```
SELECT DISTINCT city
FROM city_list
WHERE country = 'India'
ORDER BY city ASC
```

The SQL Query to Extract Data:

```
SELECT c.year, c.city, c.avg_temp AS city_avgtemp,
g.avg_temp AS global_avgtemp
FROM city_data AS c
INNER JOIN global_data AS g
ON c.year = g.year
WHERE c.city = 'Delhi'
ORDER BY c.year ASC
```

3. ANALYSIS OF DATA

The data is analysed for plotting the weather trends for both Delhi and Global. The Data is downloaded as CSV file and uploaded into the jupyter notebook by the code below. The first five rows of the csv file is displayed to check the data.

In [51]:

```
import types
import pandas as pd
import matplotlib.pyplot as plt # Data Visualisation
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes you
r credentials.
# You might want to remove those credentials before you share the notebook.
client_740e411a231b49a6ad6dfa4d89ba546f = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='S3xWX5xfJKchyiDGYNBNjln4_puMLuH7XiF01bqFLhTK',
    ibm_auth_endpoint="https://iam.eu-gb.bluemix.net/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.eu-geo.objectstorage.service.networklayer.com')

body = client_740e411a231b49a6ad6dfa4d89ba546f.get_object(Bucket='udacityprojects-dono
tdelete-pr-pi4ed38xnrnlnb',Key='delhi weather.csv')['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )


#The Data is Read from the SQL query and Saved as the temp_avg.csv file

temp_avg = pd.read_csv(body) # reading data from csv file uploaded to the Data in the
 project
temp_avg.head(8) # first 5 rows of the dataframe
```

Out[51]:

	city	year	city_avgtemp	global_avgtemp
0	Delhi	1796	25.03	8.27
1	Delhi	1797	26.71	8.51
2	Delhi	1798	24.29	8.67
3	Delhi	1799	25.28	8.51
4	Delhi	1800	25.21	8.48
5	Delhi	1801	24.22	8.59
6	Delhi	1802	25.63	8.58
7	Delhi	1803	25.38	8.50

The data types of the columns in the weather trends csv file.

In [43]:

```
temp_avg.dtypes # to display the data types of the each column
```

Out[43]:

```
city               object
year                int64
city_avgtemp      float64
global_avgtemp    float64
dtype: object
```

3.1 Calculating of the moving average for the global_avgtemp and city_avgtemp.

The moving average of the data is calculated by the pandas dataframe rooling().mean() function.
The concept of rolling window calculation is most primarily used in signal processing and time series data.
The moving average is calculated for 20 years

In [55]:

```
glb_avg = temp_avg['global_avgtemp'].rolling(20).mean() # moving average of the global
_avgtemp by window size of 5.
glb_avg.head(5) # to display the first 5 rows of the calculated moving average of the
 global_avgtemp
```

Out[55]:

```
0   NaN
1   NaN
2   NaN
3   NaN
4   NaN
Name: global_avgtemp, dtype: float64
```

In [52]:

```
cty_avg = temp_avg['city_avgtemp'].rolling(20).mean() # moving average of the city_avg
temp by window size of 5.
cty_avg.head(8) # to display the first 5 rows of the calculated moving average of the
 city_avgtemp.
```

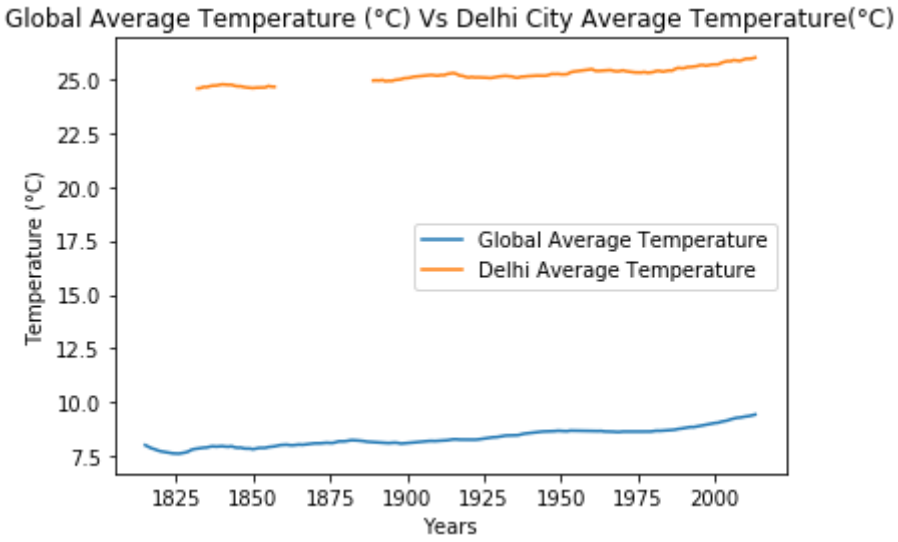
Out[52]:

```
0   NaN
1   NaN
2   NaN
3   NaN
4   NaN
5   NaN
6   NaN
7   NaN
Name: city_avgtemp, dtype: float64
```

3.2 Visualization of the data

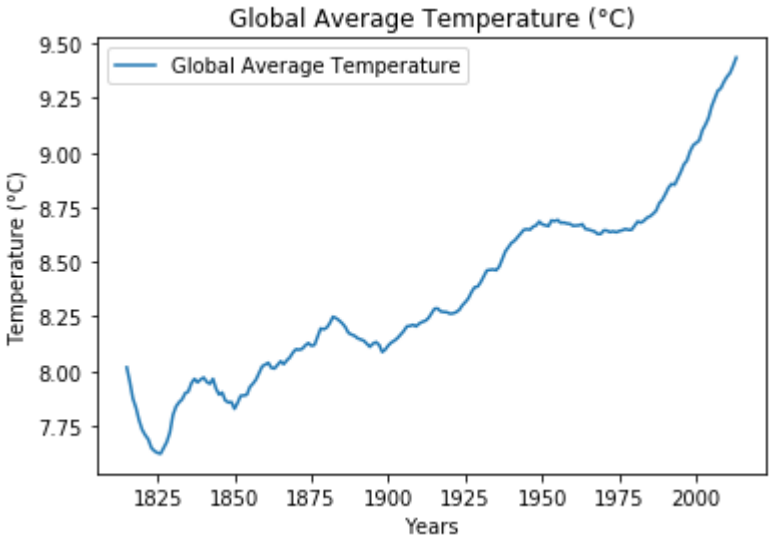
In [54]:

```
plt.plot(temp_avg['year'],glb_avg,label='Global Average Temperature') # Plotting the
 "year" and the calculated moving average of the global_avgtemp "glb_avg".
plt.plot(temp_avg['year'],cty_avg,label='Delhi Average Temperature') # Plotting the "y
ear" and the calculated moving average of the city_avgtemo "cty_avg".
plt.legend()
plt.xlabel("Years") # Years as the Label for X axis
plt.ylabel("Temperature (°C)") # Temperature (°C) as the Label For Y axis
plt.title("Global Average Temperature (°C) Vs Delhi City Average Temperature(°C)") # T
itle of the Line Plot
plt.show()
```



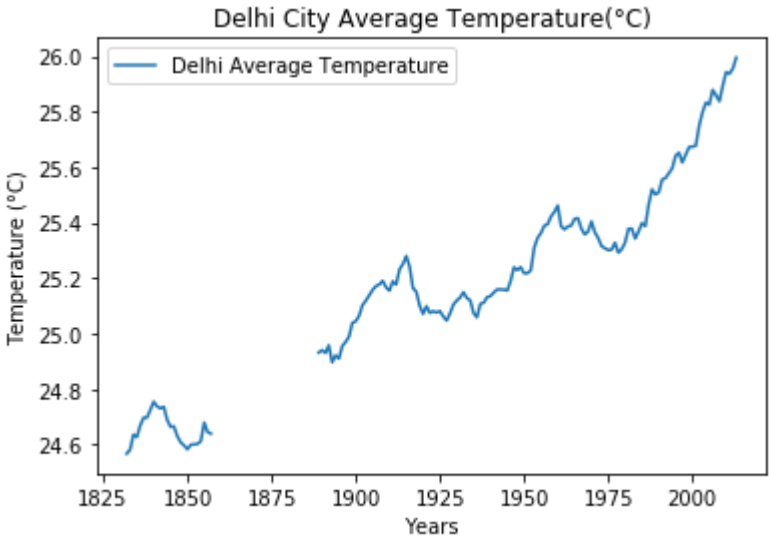
In [47]:

```
plt.plot(temp_avg['year'],glb_avg,label='Global Average Temperature') # Plotting the
 "year" and the calculated moving average of the global_avgtemp "glb_avg".
plt.legend()
plt.xlabel("Years") # Years as the Label for X axis
plt.ylabel("Temperature (°C)") # Temperature (°C) as the Label For Y axis
plt.title("Global Average Temperature (°C)") # Title of the Line Plot
plt.show()
```



In [48]:

```
plt.plot(temp_avg['year'],cty_avg,label='Delhi Average Temperature') # Plotting the "y
ear" and the calculated moving average of the city_avgtemo "cty_avg".
plt.legend()
plt.xlabel("Years") # Years as the Label for X axis
plt.ylabel("Temperature (°C)") # Temperature (°C) as the Label For Y axis
plt.title("Delhi City Average Temperature(°C)") # Title of the Line Plot
plt.show()
```



4. OBSERVATION

The result of the data visualized from the weather trends shows that there is an increase in the temperature both Delhi and the Global. The following are some accurate observation from the line plot.

- There is fluctuation of the temperature in both golbal and in Delhi between 1800-1900.
- Between the period of 1900-1950s there is increase in global temperature whereas rise in temperature of Delhi is very low.
- There is an optimum increase in temperature between 1950-2000s both globally and in Delhi
- There is drastic increase in temperature after 2000s, both globally and Delhi.

5. CONCLUSION

The given datais extracted, analysed and visualised. From the Observation its is clear that there is a drastic increase in temperature after 2000s. If this continues there will be a desperate increase in temperature in the upcoming year leading to great disasters.