Explore Weather Trends

This project presents an analysis of local and global temperatures for a period of more than 250 years. The global temperatures trends are compared with local trends for Columbus (Ohio, US) and two other cities: Madrid (Spain) and Helsinki (Finland). This presentation is written in Jupyter Notebook, on a Windows 7 Professional system. The basic steps of data wrangling were performed using Google sheets.

1. Extracting the Data

The Database Schema provided by Udacity contains three tables:

```
city_list - This contains a list of cities and countries in the database.
city_data - This contains the average temperatures for each city by year (ºC).
global_data - This contains the average global temperatures by year (ºC).
```

In [1]:
```
### Package to parse the SQL statements:

import sqlparse
```

Find the cities in the database

First we check the `city_list` to determine if the cities Columbus, Madrid and Helsinki are in the database. This is done in the Udacity workspace with the following queries.

In [2]:
```
sql = 'select * from city_list where country = "United States" and city="Columbus";'
print(sqlparse.format(sql, reindent=True, keyword_case='upper'))
```

```
SELECT *
FROM city_list
WHERE country = "United States"
  AND city="Columbus";
```

And for the two remaining two cities:

In [3]:
```
sql = 'select * from city_list where city like "Madrid" or city like "Helsinki";'
print(sqlparse.format(sql, reindent=True, keyword_case='upper'))
```

```
SELECT *
FROM city_list
WHERE city LIKE "Madrid"
  OR city LIKE "Helsinki";
```

The reasons behind chosing these three cities

Columbus, Ohio, is the city I live in and I am familiar with the weather patterns in this area.

In the article "Explainer: How do scientists measure global temperature?", from CarbonBrief, linked to the Udacity website, it is mentioned that the Artic is warming more than twice as fast as the global average. It seems plausible to compare weather trends for Columbus and for a city closer to the North Pole, such as Helsinki, Finland. The latitude of Columbus is about 40º while the latitude of Helsinki is close to 60º.

Finally, Madrid, Spain and Columbus have almost the same latitude.

Download the data in csv files and process them in a single Google spreadsheet

Download the global average temperatures as `global_data.csv`. This is done in two steps, a SQL query, followed by downloading the file from the Udacity website.

In [4]:
```
sql = 'select * from global_data;'
print(sqlparse.format(sql, reindent=True, keyword_case='upper'))
```

```
SELECT *
FROM global_data;
```

Download the data for the three cities mentioned above, the SQL queries are given below.

In [5]:
```
cities = ["Columbus", "Helsinki", "Madrid"]
for cit in cities:
    sql = 'select year. avg_temp from city_data where city = {}'.format(cit)
    print(sqlparse.format(sql, reindent=True, keyword_case='upper'))
    print("")
```

```
SELECT year. avg_temp
FROM city_data
WHERE city = Columbus

SELECT year. avg_temp
FROM city_data
WHERE city = Helsinki

SELECT year. avg_temp
FROM city_data
WHERE city = Madrid
```

Following the project recommendations, Google spreadsheets are used to join the data for the three cities together with the global data in a single spreadsheet. The new average temperatures column headers are: avg_temp_cit where cit is Col, Mad, Hel, while avg_temp denotes the global average temperatures.

The data is combined using the `Import/Replace data starting at selected cell` feature in Google spreadsheets. The column headers are renamed and the extra years columns are deleted. The new spreadsheet is saved in `combined_data.csv`.

2. Determine the Optimal Interval for Moving Averages

To start with, we will use the data for Columbus only, available in `columbus_data.csv`, to determine the optimal interval for moving averages. We look at 5 years, 10 years and 20 years moving averages graphs.

In [6]:
```
### Set the working directory in Anaconda/Python3:

import sys
sys.path.append("../nanodegree/project1_weather")
```

In [7]:
```
### To inspect the table we use Pandas and SqlAlchemy.

import pandas as pd
from sqlalchemy import create_engine
engine = create_engine('sqlite:///:memory:')

data_col = pd.read_csv('columbus_data.csv', index_col=0, parse_dates=False)
```

```
data_col.to_sql('data_col', engine)

with engine.connect() as conn, conn.begin():
    pd.read_sql_table('data_col', conn)
```

Observation:

The temperature measurements for Columbus start in 1743, but several values are missing between 1743 and 1750, also the global averages are available from 1750. We will take into account the data avaliable from 1750, the few missing values will not have a significant effect on the weather trends.

In [8]:
```
pd.read_sql_query('SELECT * FROM data_col WHERE year >= 1750 LIMIT 20;', engine)
```

Out[8]:

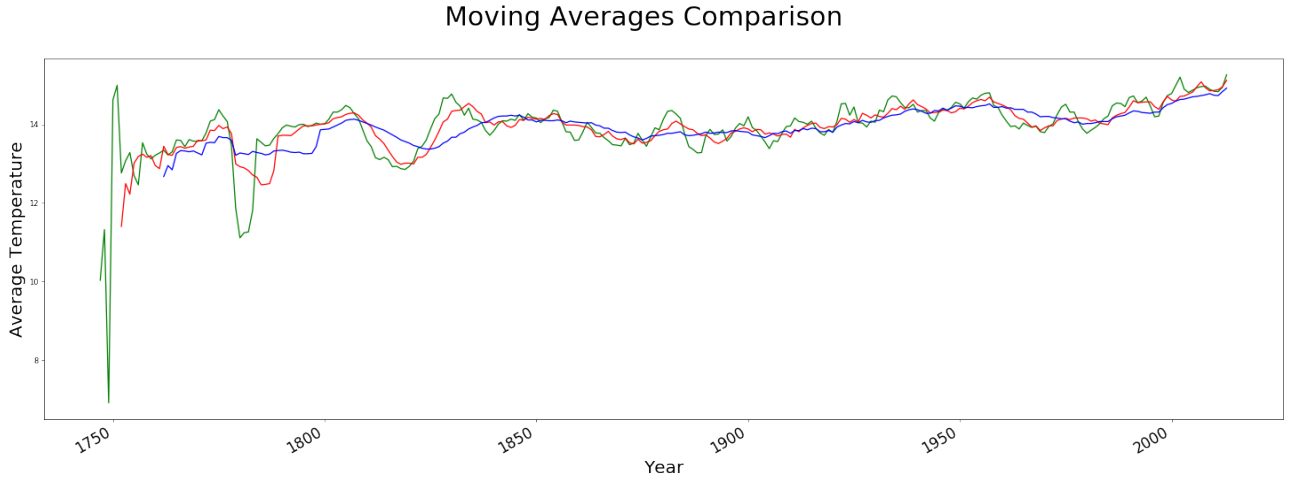
	year	avg_temp	5yr_ma	10yr_ma	20yr_ma
0	1750	14.62	14.62	NaN	NaN
1	1751	15.36	14.99	NaN	NaN
2	1752	8.30	12.76	11.40	NaN
3	1753	14.00	13.07	12.49	NaN
4	1754	14.11	13.28	12.22	NaN
5	1755	11.66	12.69	13.01	NaN
6	1756	14.24	12.46	13.18	NaN
7	1757	13.62	13.53	13.24	NaN
8	1758	12.55	13.24	13.16	NaN
9	1759	13.55	13.12	13.20	NaN
10	1760	12.10	13.21	12.95	NaN
11	1761	14.52	13.27	12.87	NaN
12	1762	14.00	13.34	13.44	12.67
13	1763	11.94	13.22	13.23	12.95
14	1764	13.96	13.30	13.21	12.84
15	1765	13.58	13.60	13.41	13.26
16	1766	14.45	13.59	13.43	13.33
17	1767	13.22	13.43	13.39	13.32
18	1768	12.86	13.61	13.42	13.30
19	1769	13.73	13.57	13.44	13.32

To plot the moving averages we use `matplotlib` package for Python3 from Anaconda distribution.

In [9]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

In [10]:
```
data_col = pd.read_csv('columbus_data.csv')

### label the axes
x = data_col['year']
```

```
y1 = data_col['5yr_ma']
y2 = data_col['10yr_ma']
y3 = data_col['20yr_ma']

### plot
plt.figure(figsize=(30, 10), dpi=60, linewidth=2, frameon=True)

plt.plot(x, y1, color="green",  linewidth=1.5) ### 5 years moving averages
plt.plot(x, y2, color="red",  linewidth=1.5) ### 10 years moving averages
plt.plot(x, y3, color="blue",  linewidth=1.5) ### 20 years moving averages

### beautify the x-labels
plt.gcf().autofmt_xdate()
plt.xticks(fontsize = 20)

plt.suptitle('Moving Averages Comparison', fontsize=36)
plt.xlabel('Year', fontsize=24)
plt.ylabel('Average Temperature', fontsize=24)

plt.show()
```



Observation:

From this representation it is clear that a 5 years interval for moving averages does not significantly eliminate the local variations, while a 20 years moving average might overlook some of the relevant variations. We decide to work with 10 years moving averages.

3. Weather Trends Comparison: Global Temperatures vs. Columbus, Ohio

In this section we will work with `combined_data.csv` that was updated with 10 years moving averages for each of the four cities, these are calculated in the columns Col_ma, Hel_ma, Mad_ma and sanD_ma. The measurements for San Diego start in 1849.

In [11]:
```
### A data snippet that includes the first 20 rows and several columns from the table.

combined = pd.read_csv('combined_data.csv', index_col=0, parse_dates=False)
combined.to_sql('combined', engine)

with engine.connect() as conn, conn.begin():
    pd.read_sql_table('combined', conn)

pd.read_sql_query('SELECT year,avg_temp_hel,Hel_ma,avg_temp_mad,Mad_ma FROM combined WHERE year >= 175
0 LIMIT 20;', engine)
```

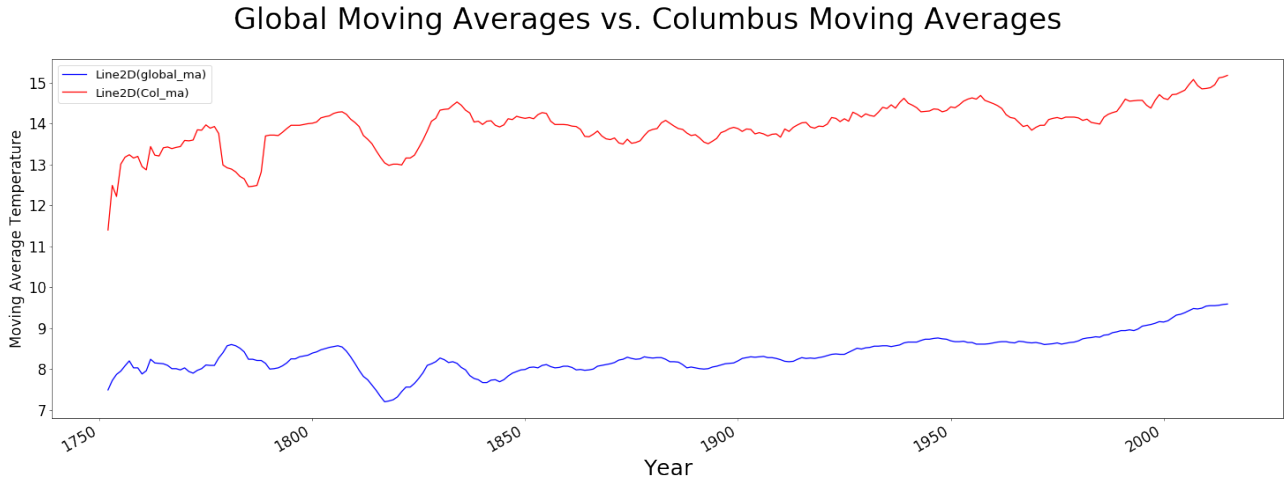
Out[11]:

	year	avg_temp_Hel	Hel_ma	avg_temp_Mad	Mad_ma
0	1750	5.14	NaN	12.01	NaN
1	1751	4.68	NaN	12.71	NaN

2	1752	-0.29	1.91	7.07	9.39
3	1753	4.14	2.34	11.47	10.19
4	1754	4.15	1.98	11.49	10.04
5	1755	4.05	3.65	11.17	10.99
6	1756	4.47	3.76	11.50	11.06
7	1757	4.75	3.89	11.34	11.10
8	1758	2.66	3.75	10.24	11.00
9	1759	4.08	3.78	11.35	11.04
10	1760	2.79	3.55	10.92	10.93
11	1761	4.55	3.54	11.53	10.81
12	1762	4.22	3.99	11.43	11.24
13	1763	3.35	3.91	10.79	11.18
14	1764	4.55	3.95	11.46	11.17
15	1765	4.22	3.96	11.48	11.20
16	1766	4.75	3.99	11.34	11.19
17	1767	3.99	3.92	11.45	11.20
18	1768	3.35	3.99	10.77	11.25
19	1769	4.03	3.98	11.23	11.24

We focus on Columbus for now, to compare the weather trends with the global trends.

In [12]:
```
### read the csv file in a pandas frame
data_combd = pd.read_csv('combined_data.csv')

### the axes
x = data_combd['year']
y1 = data_combd['global_ma']
y2 = data_combd['Col_ma']

### plot
plt.figure(figsize=(30, 10), dpi=60, linewidth=2, frameon=True)

### global 10 years moving averages
line_up,=plt.plot(x, y1, color="blue",  linewidth=1.5, linestyle="-")

 ### Columbus 10 years moving averages
line_down,=plt.plot(x, y2, color="red",  linewidth=1.5, linestyle="-")

### beautify the x-labels
plt.gcf().autofmt_xdate()
plt.xticks(fontsize = 20)
plt.yticks(fontsize = 20)

### the labels, legend
plt.suptitle('Global Moving Averages vs. Columbus Moving Averages', fontsize=40)
plt.xlabel('Year', fontsize=30)
plt.ylabel('Moving Average Temperature', fontsize=20)
plt.legend([line_up,line_down], fontsize=16)

plt.show()
```



Naive observations:

1). Columbus temperatures are higher than the global averages. Columbus moving average temperatures vary between 12(ºC) and 14(ºC), while the moving average global temperatures are within the range of 7(ºC) to 10(ºC).

2). For both graphs we see a lot of variation for the first 100 years, roughly speaking from 1750 to 1850. This might be related to how the measurements were performed at that time. But more interesting, we see a clear drop in the average temperatures on both graphs, around the year 1820. I believe this is due to what is known as the Little Ice Age (see Wikipedia for example). This was a period of several hundred years (with an assumed lifespan from 1300 to 1850) with lower average temperatures.

3). If we focus on the last 150 years, after the end of the Little Ice Age, we see positive slopes for both graphs, showing clear tendencies of warming temperatures.

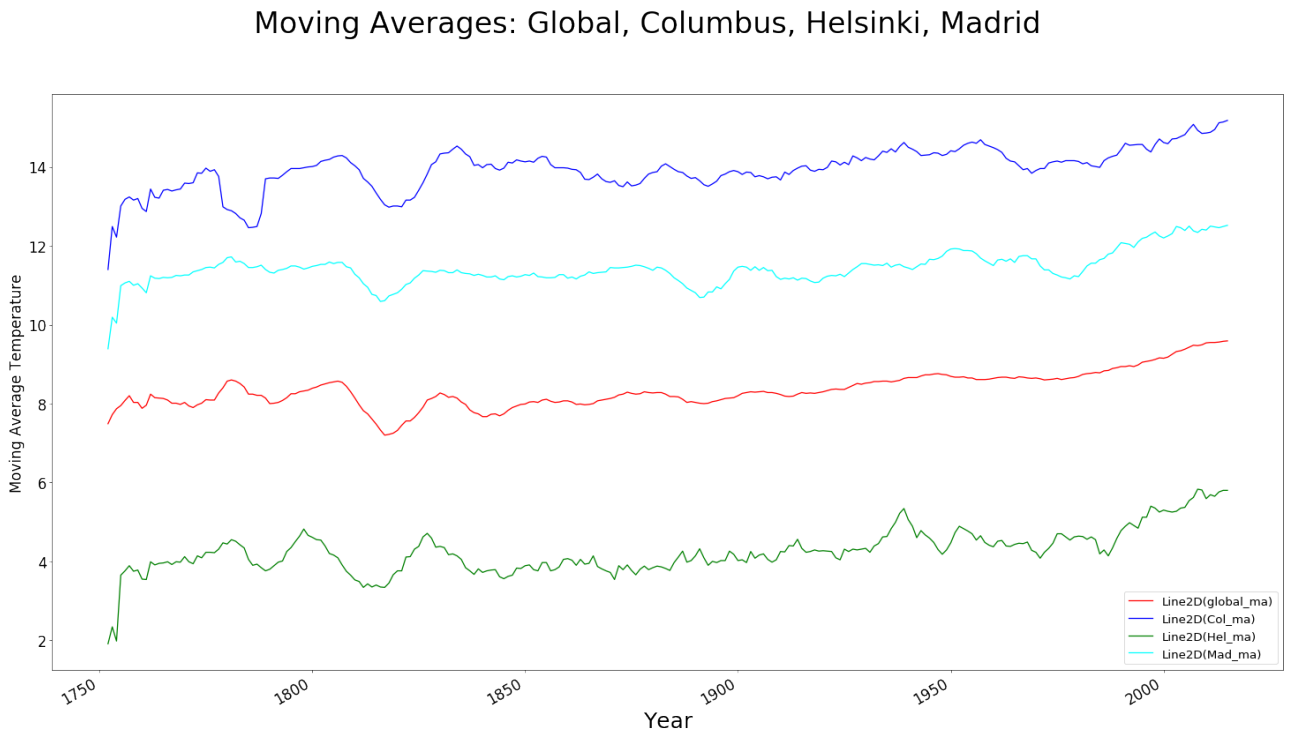
4). The global temperatures seem to have an almost linear behavior, with an increase in the slope of the line that happened around 1970. A similar trend can be observed for Columbus, but the data is more zigzagged.

4. Weather Trends: Global Data, Columbus, Helsinki, Madrid

We start by plotting the 4 line graphs, the global moving averages and the moving averages for the three cities.

In [13]:
```
### read the csv file in a pandas frame
data_combd = pd.read_csv('combined_data.csv')

### the axes
x = data_combd['year']
yg = data_combd['global_ma']
yc = data_combd['Col_ma']
yh = data_combd['Hel_ma']
ym = data_combd['Mad_ma']

### plot
plt.figure(figsize=(30, 16), dpi=60, linewidth=2, frameon=True)

### global 10 years moving averages
line_g,=plt.plot(x, yg, color="red",  linewidth=1.5, linestyle="-")

### Columbus 10 years moving averages
line_c,=plt.plot(x, yc, color="blue",  linewidth=1.5, linestyle="-")

### Helsinki 10 years moving averages
line_h,=plt.plot(x, yh, color="green",  linewidth=1.5, linestyle="-")

### Madrid 10 years moving averages
line_m,=plt.plot(x, ym, color="cyan",  linewidth=1.5, linestyle="-")
```

```
### beautify the x-labels
plt.gcf().autofmt_xdate()
plt.xticks(fontsize = 20)
plt.yticks(fontsize = 20)

### the labels, legend
plt.suptitle('Moving Averages: Global, Columbus, Helsinki, Madrid', fontsize=40)
plt.xlabel('Year', fontsize=30)
plt.ylabel('Moving Average Temperature', fontsize=20)
plt.legend([line_g,line_c, line_h, line_m], fontsize=16)

plt.show()
```



Observations:

1). The local averages show more variations than the global temperatures.

2). In all these line graphs there is a temperature drop in the period between 1800 and 1850 which could be associated to a localized effect of the Little Ice Age.

3). The shapes of the graphs are similar, they show consistent increase in the average temperatures. A change in the slope can also be observed around 1980, and it seems to be steepest for Helsinki.

5. Statistical Explorations

Pearson correlation coefficient

Display the correlation matrix using Pandas. This will compute pairwise correlation coefficients for selected columns from the `combined` table (the NA and null values are excluded).

In [14]:
```
my_data = pd.read_sql_query('SELECT global_ma, Col_ma, Hel_ma, Mad_ma FROM combined WHERE year >= 175
0;', engine)
my_data.corr()
```

Out[14]:

	global_ma	Col_ma	Hel_ma	Mad_ma
global_ma	1.000000	0.679751	0.835844	0.862759

Col_ma	0.679751	1.000000	0.710972	0.701754
Hel_ma	0.835844	0.710972	1.000000	0.826125
Mad_ma	0.862759	0.701754	0.826125	1.000000

Remarks:

From the above table we notice that the temperatures in Madrid are the closest to the global trend, while Columbus are the least correlated to the global averages.

Linear Regression

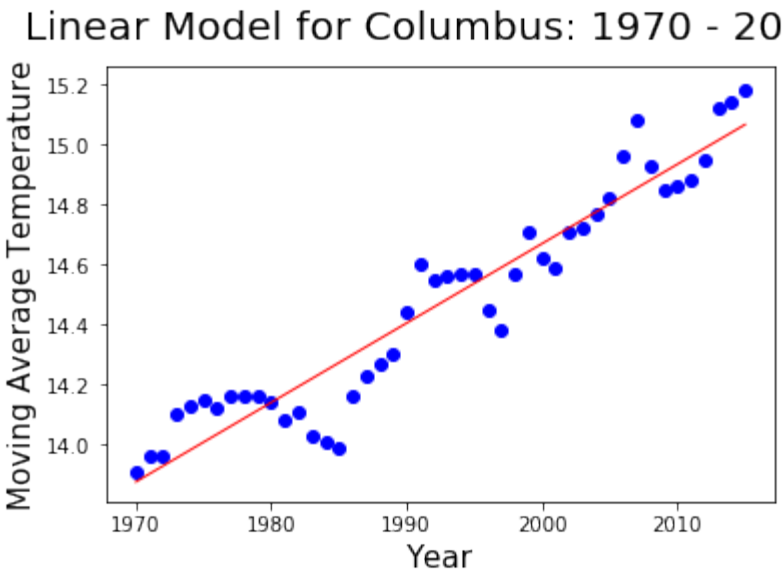
We fit simple linear models using `sklearn.linear_model.LinearRegression`.

Since the data, for both global averages and local measurements seem to fit best on a line if restricted to the past 50 years, we shall construct this model for the 10 years moving averages that correspond to the period between 1970 and 2013. We will work with the global data and with the data corresponding to Columbus.

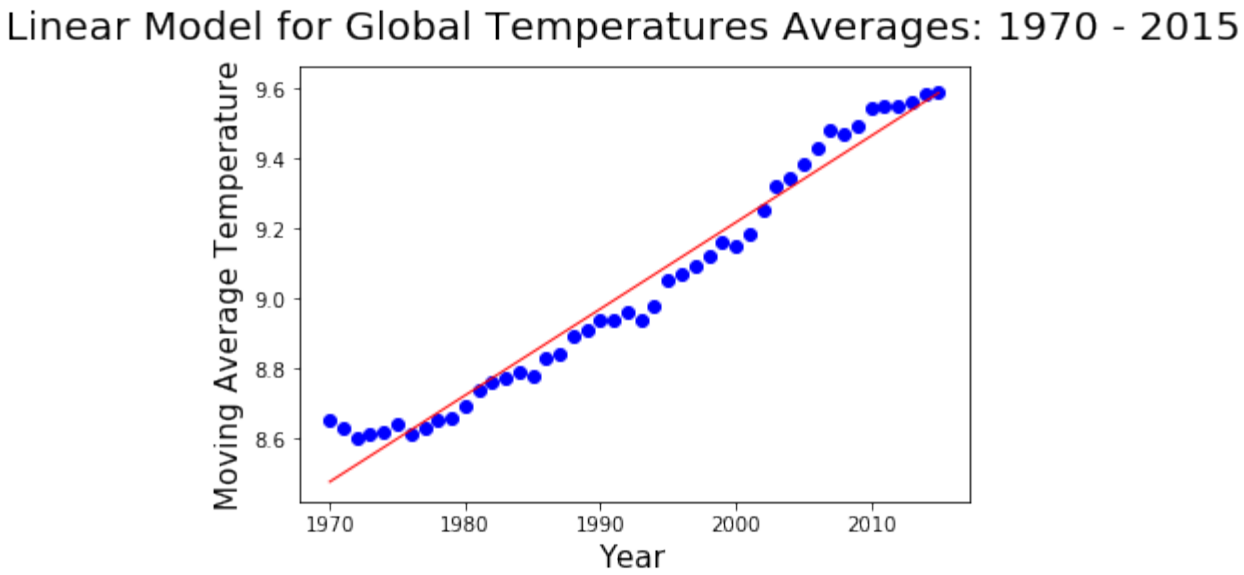
In [15]:
```
### get the necessary packages

import numpy as np
import pandas as pd
from sklearn import datasets, linear_model
import matplotlib.pyplot as plt
```

In [16]:
```
### extract the data for 1970 to 2015

world = pd.read_csv('combined_data.csv', index_col=False, header=0, na_values=0)

x = world['year'][-46:]

y1=world['Col_ma'][-46:]  ### for Columbus, 10 years moving averages
y2 = world['global_ma'][-46:] ### global 10 years moving averages

x = x.values.reshape(len(x), 1)
y1 = y1.values.reshape(len(y1), 1)
y2 = y2.values.reshape(len(y2), 1)
```

In [17]:
```
### the linear model for Columbus

regr1 = linear_model.LinearRegression()
regr1.fit(x, y1)

### plot

plt.scatter(x, y1,  color='blue')
plt.plot(x, regr1.predict(x), color='red', linewidth=1)

### the labels
plt.suptitle('Linear Model for Columbus: 1970 - 2015', fontsize=20)
plt.xlabel('Year', fontsize=15)
plt.ylabel('Moving Average Temperature', fontsize=15)


plt.show()
```



In [18]:
```
### the linear model for the global temperatures

regr2 = linear_model.LinearRegression()
regr2.fit(x, y2)

### plot

plt.scatter(x, y2,  color='blue')
plt.plot(x, regr2.predict(x), color='red', linewidth=1)

### the labels
plt.suptitle('Linear Model for Global Temperatures Averages: 1970 - 2015', fontsize=20)
plt.xlabel('Year', fontsize=15)
plt.ylabel('Moving Average Temperature', fontsize=15)


plt.show()
```



Temperature Estimates

We can now estimate the average temperatures in 2017 for both the world and Columbus as follows:

In [19]:
```
lr1 = linear_model.LinearRegression().fit(x,y1)
tC = lr1.predict(2017)[0][0]
```

```
print('The predicted average temperature for Columbus in 2017 is: {}'.format(tC))

lr2 = linear_model.LinearRegression().fit(x,y2)
tG = lr2.predict(2017)[0][0]
print('The predicted globall average temperature in 2017 is: {}'.format(tG))
```

```
The predicted average temperature for Columbus in 2017 is: 15.11992599444958
The predicted globall average temperature in 2017 is: 9.635136601911803
```

6. Closing Remarks

The similarities and differences between the global temperature trend and the temperature trends in Columbus, Ohio are discussed. Two other cities Madrid (Spain) and Helsinki (Finland) are also included in this investigation.

The analysis is performed using 10 years moving averages. The line graphs for the four sets of data (global and the three cities) have similar shapes; the global temperatures have a smoother shape than the individual city data. There is a clear pattern of increasing temperatures, especially for the past 100 years.

The correlation coefficient matrix is computed. Simple linear models are constructed for the global data and for Columbus, for the period of time from 1970 to 2015.

The `combined_data.csv` is attached to this document.