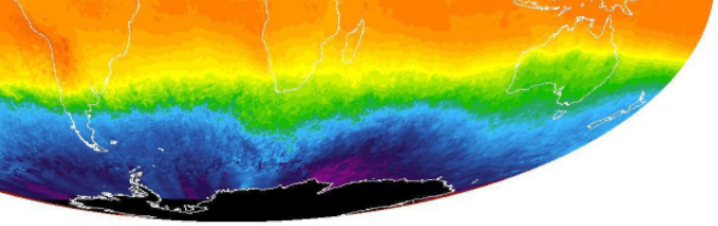


PROJECT | EXPLORING WEATHER TRENDS

Term 1 : Data Analyst Nanodegree Program

Submitted by : Rajendra B Pal

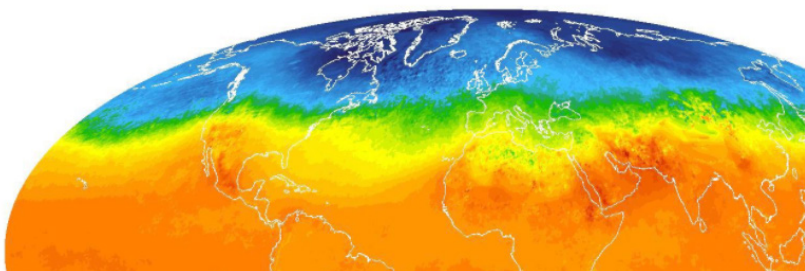
1. Data Extraction

2. Opening CSV Files

3. Line Chart

4. Making observations

5. Extra



1. DATA EXTRACTION

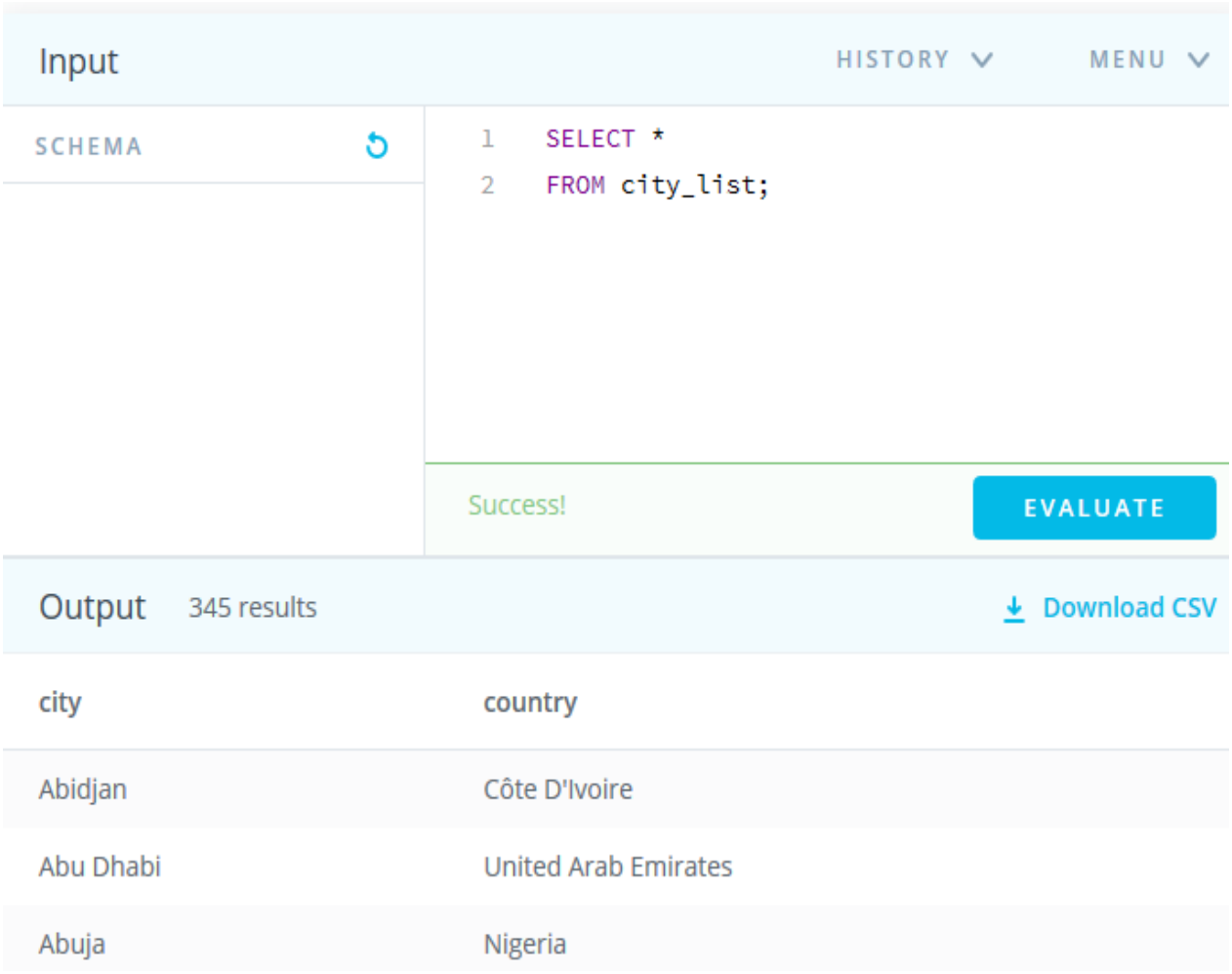


Fig 1-01 : SQL query to output columns from table city_list.

Extracting city data from the temperatures database, city_list table. SQL queries shown in Fig 1-01. Clicked 'Download CSV' link to download CSV file. Repeated above steps for table 'city_data' and 'global_data'.

1. DATA EXTRACTION

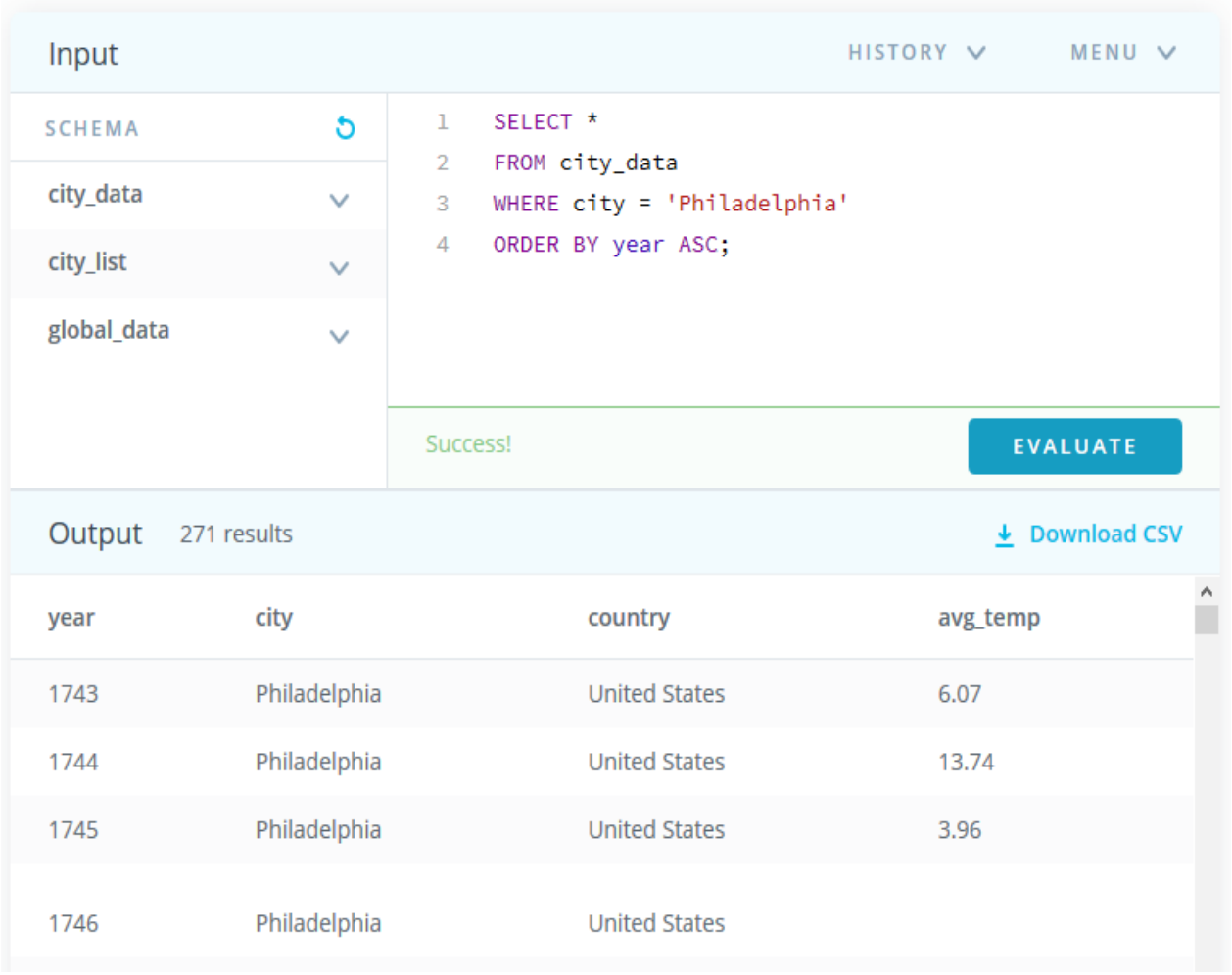


Fig 1-01 - a : SQL query to output columns from table city_list.

Extracting city Philadelphia-USA data from the temperatures database, city_list table. SQL queries shown in Fig 1-01 - a. Clicked 'Download CSV' link to download CSV file specific to city Philadelphia-USA. Repeated above steps for table 'city_data' and 'global_data'.

2. OPENING CSV FILES

Please refer Jupyter notebook "DAND-rbpal-Project Explore Weather Trends-two.ipynb" for details.

```
df_city_list = pd.read_csv('csv/ProjectExploreWeatherTrends/city_list.csv')
```

```
df_city_list.head()
```

	city	country
0	Abidjan	Côte D'Ivoire
1	Abu Dhabi	United Arab Emirates
2	Abuja	Nigeria
3	Accra	Ghana
4	Adana	Turkey

Fig 1-02 : Reading city_list CSV file

```
df_city_data = pd.read_csv('csv/ProjectExploreWeatherTrends/city_data.csv')
```

```
df_city_data.head()
```

	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

Fig 1-03 : Reading city_data CSV file

Used Python Pandas package to open CSV files as shown above.

2. OPENING CSV FILES

```
df_city_list = pd.read_csv('csv/ProjectExploreWeatherTrends/city_list.csv')
```

```
df_city_list.head()
```

	city	country
0	Abidjan	Côte D'Ivoire
1	Abu Dhabi	United Arab Emirates
2	Abuja	Nigeria
3	Accra	Ghana
4	Adana	Turkey

Fig 1-02 : Reading city_list CSV file

```
df_city_data = pd.read_csv('csv/ProjectExploreWeatherTrends/city_data.csv')
```

```
df_city_data.head()
```

	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

Fig 1-03 : Reading city_data CSV file

Used Python Pandas package to open CSV files as shown above.

3. LINE CHART

STEP 1 - DATA CLEANING

```
df_city_list.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 345 entries, 0 to 344
Data columns (total 2 columns):
city       345 non-null object
country    345 non-null object
dtypes: object(2)
memory usage: 5.5+ KB
```

```
df_city_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 71311 entries, 0 to 71310
Data columns (total 4 columns):
year        71311 non-null int64
city        71311 non-null object
country     71311 non-null object
avg_temp    68764 non-null float64
dtypes: float64(1), int64(1), object(2)
memory usage: 2.2+ MB
```

```
df_global_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 266 entries, 0 to 265
Data columns (total 2 columns):
year        266 non-null int64
avg_temp    266 non-null float64
dtypes: float64(1), int64(1)
memory usage: 4.2 KB
```

Fig 1-04

A. Data information of 'city_data' shows that column 'avg_temp' has '68764' non-null values out of total '71311'

3. LINE CHART

STEP 1 - DATA CLEANING

```
df_city_data.isnull().values.any()
```

```
True
```

```
df_city_data.isnull().sum()
```

```
year            0
city            0
country         0
avg_temp     2547
dtype: int64
```

```
df_city_data_no_null = df_city_data[pd.notnull(df_city_data['avg_temp'])]
```

```
df_city_data_no_null.isnull().values.any()
```

```
False
```

```
df_city_data_no_null.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 68764 entries, 0 to 71310
Data columns (total 4 columns):
year         68764 non-null int64
city         68764 non-null object
country      68764 non-null object
avg_temp     68764 non-null float64
dtypes: float64(1), int64(1), object(2)
memory usage: 2.6+ MB
```

Fig 1-05

B. From Dataframe 'df_city_data' discarded all rows having 'null' value against column 'avg_temp.' .

3. LINE CHART

STEP 1 - DATA CLEANING

```
df_city_data_no_null_philadelphia =
df_city_data_no_null[df_city_data_no_null['city'] == 'Philadelphia']
```

```
df_city_data_no_null_philadelphia.head()
```

	year	city	country	avg_temp
50194	1743	Philadelphia	United States	6.07
50195	1744	Philadelphia	United States	13.74
50196	1745	Philadelphia	United States	3.96
50201	1750	Philadelphia	United States	12.36
50202	1751	Philadelphia	United States	13.05

Fig 1-06

C. Extracting temperature weather trend information for city 'Philadelphia,' closest to my hometown 'Pittsburgh', from Dataframe 'df_city_data_no_null.' .

3. LINE CHART

STEP 1 - DATA CLEANING

```
df_global_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 266 entries, 0 to 265
Data columns (total 2 columns):
year         266 non-null int64
avg_temp     266 non-null float64
dtypes: float64(1), int64(1)
memory usage: 4.2 KB
```

```
df_global_data.head()
```

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

Fig 1-07

D. Data information of 'global_data' shows that there are two columns 'year' and 'avg_temp' and has '266' rows which are non-null.

3. LINE CHART

STEP 1 - DATA CLEANING

```
df_global_data_sortYear_1750_2013_index_year.head()
```

year	avg_temp
1750	8.72
1751	7.98
1752	5.78
1753	8.39
1754	8.47

```
df_global_data_sortYear_1750_2013_index_year.tail()
```

year	avg_temp
2009	9.51
2010	9.70
2011	9.52
2012	9.51
2013	9.61

Fig 1-08

E. In 'global_data', taking note of start year '1750' and end year '2013' to match against 'df_city_data' for 'Philadelphia,' Note: removed row for year 2014 and 2015.

3. LINE CHART

STEP 1 - DATA CLEANING

```
# rename global 'avg_temp_ name to 'global_avg_temp'
df_global_data_sortYear_1750_2013_index_year_ColRename =
df_global_data_sortYear_1750_2013_index_year.rename(columns=
{'avg_temp':'global_avg_temp'})
```

```
df_global_data_sortYear_1750_2013_index_year_ColRename.head()
```

year	global_avg_temp
1750	8.72
1751	7.98
1752	5.78
1753	8.39
1754	8.47

Fig 1-09

F. In 'global_data', renaming column 'avg_temp' to 'global_avg_temp' for ease.

3. LINE CHART

STEP 1 - DATA CLEANING

```
# rename column 'avg_temp' to 'phil_avg_temp'
df_city_data_no_null_philadelphia_sortYear_year_avg_temp_1750_2013_index_year_
ColRename =
df_city_data_no_null_philadelphia_sortYear_year_avg_temp_1750_2013_index_year.
rename(columns={'avg_temp':'phil_avg_temp'})
```

```
df_city_data_no_null_philadelphia_sortYear_year_avg_temp_1750_2013_index_year_
ColRename.head()
```

year	phil_avg_temp
1750	12.36
1751	13.05
1752	5.64
1753	11.85
1754	12.29

```
df_city_data_no_null_philadelphia_sortYear_year_avg_temp_1750_2013_index_year_
ColRename.tail()
```

year	phil_avg_temp
2009	12.50
2010	13.28
2011	13.65
2012	14.05
2013	14.01

```
df_city_data_no_null_philadelphia_sortYear_year_avg_temp_1750_2013_index_year_
ColRename.shape
```

```
(263, 1)
```

Fig 1-10

G. In 'city_data' for city 'Philadelphia, renaming column 'avg_temp' to 'global_avg_temp' for ease. Note the of start year '1750' and end year '2013'. Checking the shape- it comes out 263, one more than 'global_data'.

3. LINE CHART

STEP 1 - DATA CLEANING

```
df_final_global_city_philadelphia =
pd.concat([df_global_data_sortYear_1750_2013_index_year_ColRename,df_c
ity_data_no_null_philadelphia_sortYear_year_avg_temp_1750_2013_index_y
ear_ColRename],axis=1)
```

```
df_final_global_city_philadelphia.head()
```

year	global_avg_temp	phil_avg_temp
1750	8.72	12.36
1751	7.98	13.05
1752	5.78	5.64
1753	8.39	11.85
1754	8.47	12.29

```
df_final_global_city_philadelphia.tail()
```

year	global_avg_temp	phil_avg_temp
2009	9.51	12.50
2010	9.70	13.28
2011	9.52	13.65
2012	9.51	14.05
2013	9.61	14.01

```
df_final_global_city_philadelphia.isnull().sum()
```

```
global_avg_temp    0
phil_avg_temp      1
dtype: int64
```

Fig 1-11

H. Creating final Dataframe, 'df_final_global_city_philadelphia' by concatenating Dataframe 'df_global_data_sortYear_1750_2013_index_year_ColRename' and 'df_city_data_no_null_philadelphia_sortYear_year_avg_temp_1750_2013_index_year_ColRename' along axis=1.

3. LINE CHART

STEP 1 - DATA CLEANING

```
# Finding 'null' row
null_columns =
df_final_global_city_philadelphia.columns[df_final_global_city_philade
lphia.isnull().any()]
df_final_global_city_philadelphia[df_final_global_city_philadelphia['p
hil_avg_temp'].isnull()][null_columns]
```

year	phil_avg_temp
1780	NaN

```
# Removing 'null' row for year '1780'
df_final_global_city_philadelphia_no_null =
df_final_global_city_philadelphia[pd.notnull(df_final_global_city_phil
adelphia['phil_avg_temp'])]
```

```
# checking again for null row
df_final_global_city_philadelphia_no_null.isnull().sum()
```

```
global_avg_temp    0
phil_avg_temp      0
dtype: int64
```

Fig 1-12

I. From 'df_final_global_city_philadelphia' , removing year '1780' for null data. Rechecking for null value for column 'phil_avg_temp'.

3. LINE CHART

STEP 1 - DATA CLEANING

```
df_final_global_city_philadelphia.head()
```

year	global_avg_temp	phil_avg_temp
1750	8.72	12.36
1751	7.98	13.05
1752	5.78	5.64
1753	8.39	11.85
1754	8.47	12.29

```
df_final_global_city_philadelphia.tail()
```

year	global_avg_temp	phil_avg_temp
2009	9.51	12.50
2010	9.70	13.28
2011	9.52	13.65
2012	9.51	14.05
2013	9.61	14.01

Fig 1-13

J. Final Dataframe having both column 'global_avg_temp' , 'phil_avg_temp', and 'year' as index.

3. LINE CHART

STEP 2 - FINDING MOVING AVERAGES

```
# Using new method of Moving Average
df_final_global_city_philadelphia_no_null_rolling_newMethod =
df_final_global_city_philadelphia_no_null.rolling(window=7).mean()
```

```
df_final_global_city_philadelphia_no_null_rolling_newMethod.head(10)
```

year	global_avg_temp	phil_avg_temp
1750	NaN	NaN
1751	NaN	NaN
1752	NaN	NaN
1753	NaN	NaN
1754	NaN	NaN
1755	NaN	NaN
1756	8.078571	10.960000
1757	8.121429	10.802857
1758	7.944286	10.428571
1759	8.260000	11.244286

```
# checking total null
df_final_global_city_philadelphia_no_null_rolling_newMethod.isnull().s
um()
```

```
global_avg_temp     6
phil_avg_temp       6
dtype: int64
```

Fig 1-14

A. Using Python package Pandas rolling mean on Dataframe 'df_final_global_city_philadelphia_no_null_rolling_newMethod' for 7 days – resulting 6 rows null/NaN..

3. LINE CHART

STEP 2 - FINDING MOVING AVERAGES

```
# removing null rows
df_final_global_city_philadelphia_no_null_rolling_newMethod_no_null =
df_final_global_city_philadelphia_no_null_rolling_newMethod[pd.notnull
(df_final_global_city_philadelphia_no_null_rolling_newMethod['global_a
vg_temp'])]
```

```
df_final_global_city_philadelphia_no_null_rolling_newMethod_no_null.he
ad()
```

year	global_avg_temp	phil_avg_temp
1756	8.078571	10.960000
1757	8.121429	10.802857
1758	7.944286	10.428571
1759	8.260000	11.244286
1760	8.088571	10.997143

```
df_final_global_city_philadelphia_no_null_rolling_newMethod_no_null.is
null().sum()
```

```
global_avg_temp     0
phil_avg_temp       0
dtype: int64
```

Fig 1-15

B. From Dataframe 'df_final_global_city_philadelphia_no_null_rolling_newMethod', removing these 6 rows having column value 'NaN '. and then checking again for null Values.

3. LINE CHART

STEP 2 - MAKING LINE CHART

```
plt.figure(figsize=(12,6))

plt.plot(df_final_global_city_philadelphia_no_null_rolling_newMethod_
o_null.index,df_final_global_city_philadelphia_no_null_rolling_newMet
od_no_null['global_avg_temp'],'r',color='red',label='Global
Temperature Rise',alpha=0.6,lw=4)

plt.plot(df_final_global_city_philadelphia_no_null_rolling_newMethod_
o_null.index,df_final_global_city_philadelphia_no_null_rolling_newMet
od_no_null['phil_avg_temp'],'b',color='orange',label='Philadelphia
Temperature Rise',alpha=0.6,lw=4)


plt.title('Global Temperature Rise Vs Philadelphia Temperature Rise')
plt.xlabel('Year')
plt.ylabel('Average Temperature')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x13247839c18>
```

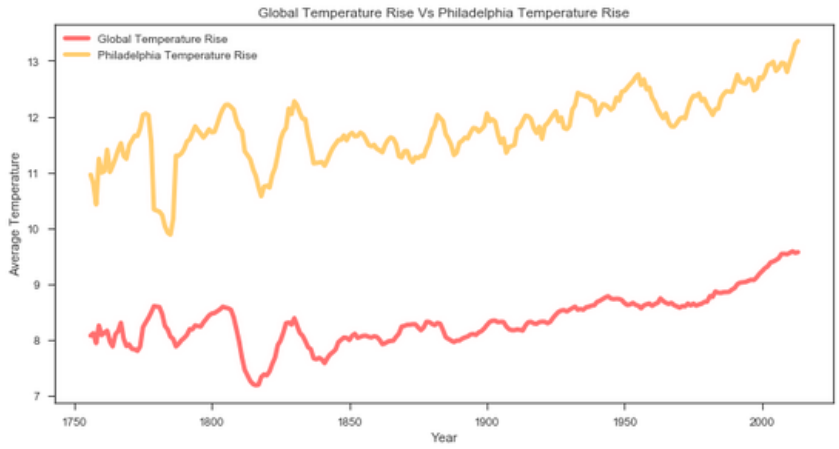


Fig 1-16

A. From Dataframe 'df_final_global_city_philadelphia_no_null_rolling_newMethod', removing these 6 rows having column value 'NaN '. and then checking again for null Values.

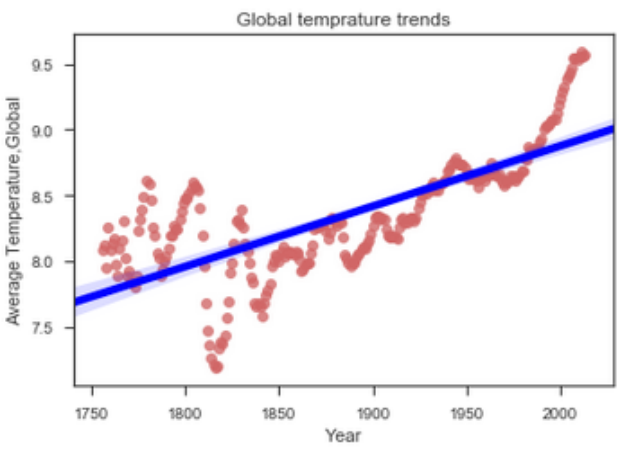
3. LINE CHART

STEP 2 - MAKING LINE CHART

```
# order 1 : global_avg_temp  | Linear Fit
sns.regplot(
    x='year',
    y='global_avg_temp',
    data=df_final_global_city_philadelphia_02,
    order=1,
    scatter_kws={'marker':'0','color':'indianred'},
    line_kws={'linewidth':5,'color':'blue'})

plt.title('Global temprature trends')
plt.xlabel('Year')
plt.ylabel('Average Temperature,Global')
plt.legend()
```

```
# order 1 : phil_avg_temp | Linear Fit
sns.regplot(
    x='year',
     y='phil_avg_temp',
    data=df_final_global_city_philadelphia_02,
    order=1,
    scatter_kws={'marker':'0','color':'indianred'},
    line_kws={'linewidth':5,'color':'blue'})
plt.title('City Philadelphia-USA, temprature trends')
plt.xlabel('Year')
plt.ylabel('Average Temperature,City Philadelphia-USA')
plt.legend()
```



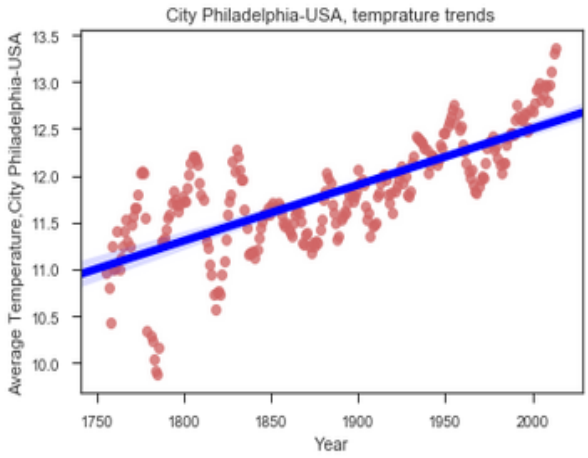


Fig 1-17

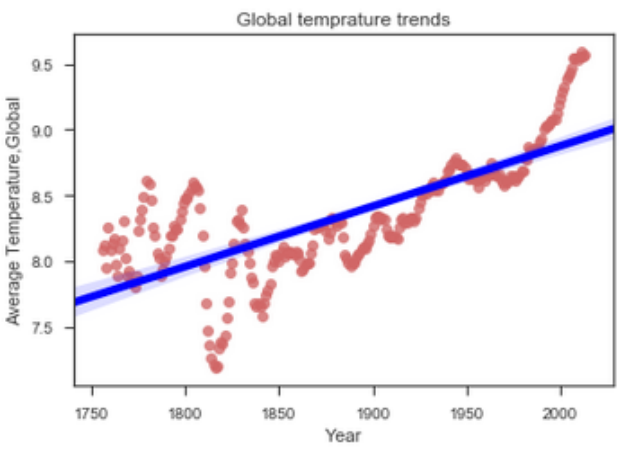
B. Using Seaborn library to plot regression line for 7 Day Moving Average, Global and City Philadelphia-USA temperature trends.

4. MAKING OBSERVATIONS

```
# order 1 : global_avg_temp  | Linear Fit
sns.regplot(
    x='year',
    y='global_avg_temp',
    data=df_final_global_city_philadelphia_02,
    order=1,
    scatter_kws={'marker':'0','color':'indianred'},
    line_kws={'linewidth':5,'color':'blue'})

plt.title('Global temprature trends')
plt.xlabel('Year')
plt.ylabel('Average Temperature,Global')
plt.legend()
```

```
# order 1 : phil_avg_temp | Linear Fit
sns.regplot(
    x='year',
     y='phil_avg_temp',
    data=df_final_global_city_philadelphia_02,
    order=1,
    scatter_kws={'marker':'0','color':'indianred'},
    line_kws={'linewidth':5,'color':'blue'})
plt.title('City Philadelphia-USA, temprature trends')
plt.xlabel('Year')
plt.ylabel('Average Temperature,City Philadelphia-USA')
plt.legend()
```



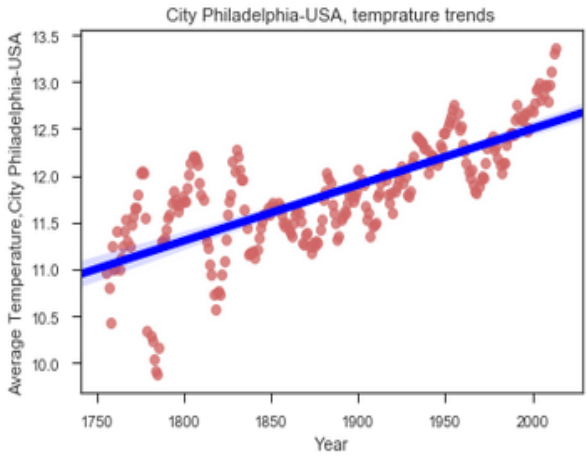


Fig 1-17

A. Figure-1-17, regression line show positive slope implying trend in temperature rise from year 1750 to 2013.

4. MAKING OBSERVATIONS

Slope for Global Temperature trend

```
series_year = df_final_global_city_philadelphia_02['year']
```

```
series_global_avg_temp = df_final_global_city_philadelphia_02['global_avg_temp']
```

```
# finding slop of regression line for global temp
from scipy.stats import linregress
linregress(series_year,series_global_avg_temp)
```

```
LinregressResult(slope=0.0045946002421798238, intercept=-0.31381095465200204, rvalue=0.739924
84785095092, pvalue=8.3255541941042298e-46, stderr=0.00026158010202298254)
```

Finding slope for City Philadelphia

```
series_phil_avg_temp = df_final_global_city_philadelphia_02['phil_avg_temp']
```

```
linregress(series_year,series_phil_avg_temp)
```

```
LinregressResult(slope=0.0059865151472888178, intercept=0.5272342528329208, rvalue=0.74908731
811250906, pvalue=1.6595795744261839e-47, stderr=0.00033154203782684375)
```

Fig 1-18

B. Slope value of city Philadelphia-USA temperature trends is higher than Global temperature trends, thus city Philadelphia-USA temperature are rising at higher rate in comparison to Global temperature trends.

4. MAKING OBSERVATIONS

```
df_final_global_city_philadelphia_02.head()
```

	year	global_avg_temp	phil_avg_temp
0	1756	8.078571	10.960000
1	1757	8.121429	10.802857
2	1758	7.944286	10.428571
3	1759	8.260000	11.244286
4	1760	8.088571	10.997143

```
# finding corelation
df_final_global_city_philadelphia_02.corr()
```

	year	global_avg_temp	phil_avg_temp
year	1.000000	0.739925	0.749087
global_avg_temp	0.739925	1.000000	0.779124
phil_avg_temp	0.749087	0.779124	1.000000

```
# Plotting heat map
sns.heatmap(df_final_global_city_philadelphia_02.corr(),
            annot=True,
            cmap='BrBG',
            linewidths=0.2)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1324c3be0b8>
```

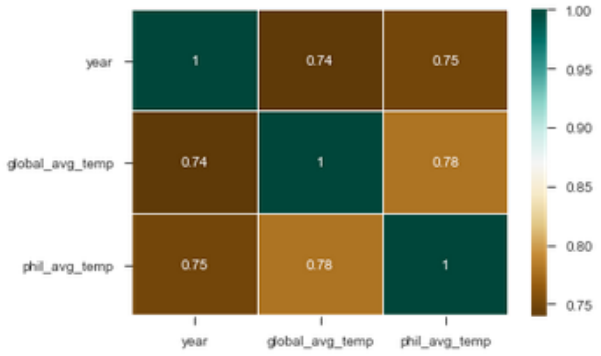


Fig 1-19

C. There is some correlation,0.78, between Global temperature and city Philadelphia-USA temperature as shown above.

D. Overall trend from last few centuries show temperature rise resulting Earth temperature gradually getting warmer.

4. EXTRA

ASSUMPTIONS

- Created Dataframe from city_list CSV and discarded all NaN rows.

- Chosen cities from different continents
  1. Accra - Ghana, Africa
  2. Surat - India, Asia
  3. Sydney-Australia
  4. Wuhan – China, Asia

- Created new Dataframe of the specific city.

- Chosen matching years from city Accra-Ghana, Africa, for parity i.e. from year 1849 to 2013.

- Rename cities column name “avg_temp” to identify Colum correctly.

- Then created final Dataframe with columns “df_global_data_final”, “df_city_data_Surat_final”, “df_city_data_Sydney_final”, “df_city_data_Wuhan_final”

- Taken seven-day rolling average for smooth plot. Removed 6 rows NaN rows as result after applying seven-day rolling average

4. EXTRA

Please refer Jupyter notebook "DAND-rbpal-Project Explore Weather Trends -ExtrafavoriteCities.ipynb" for details.

```
#plot

plt.figure(figsize=(10,6))

plt.plot(df_final_mean_rolling_7days_mean.index,df_final_mean_rolling_7days_mean['global_avg_temp'],'r',color='red',lab
el='Global Temperature Trends',lw=4)

plt.plot(df_final_mean_rolling_7days_mean.index,df_final_mean_rolling_7days_mean['Accra_avg_temp'],'b',color='blue',lab
el='City:Accra-Ghane Temperature trend',lw=4)

plt.plot(df_final_mean_rolling_7days_mean.index,df_final_mean_rolling_7days_mean['Surat_avg_temp'],'b',color='green',la
bel='City:Surat-India Temperature trend',lw=4)

plt.plot(df_final_mean_rolling_7days_mean.index,df_final_mean_rolling_7days_mean['Sydney_avg_temp'],'b',color='silver',
label='City:Sydney-Australia Temperature trend',lw=4)

plt.plot(df_final_mean_rolling_7days_mean.index,df_final_mean_rolling_7days_mean['Wuhan_avg_temp'],'b',color='teal',lab
el='City:Wuhan-China Temperature trend',lw=4)

plt.title('Global Temperature trend Vs Different cities in the world')
plt.xlabel('Year')
plt.ylabel('Average Temperature')
plt.legend(loc='center left', bbox_to_anchor=(1.0, 0.5))
```

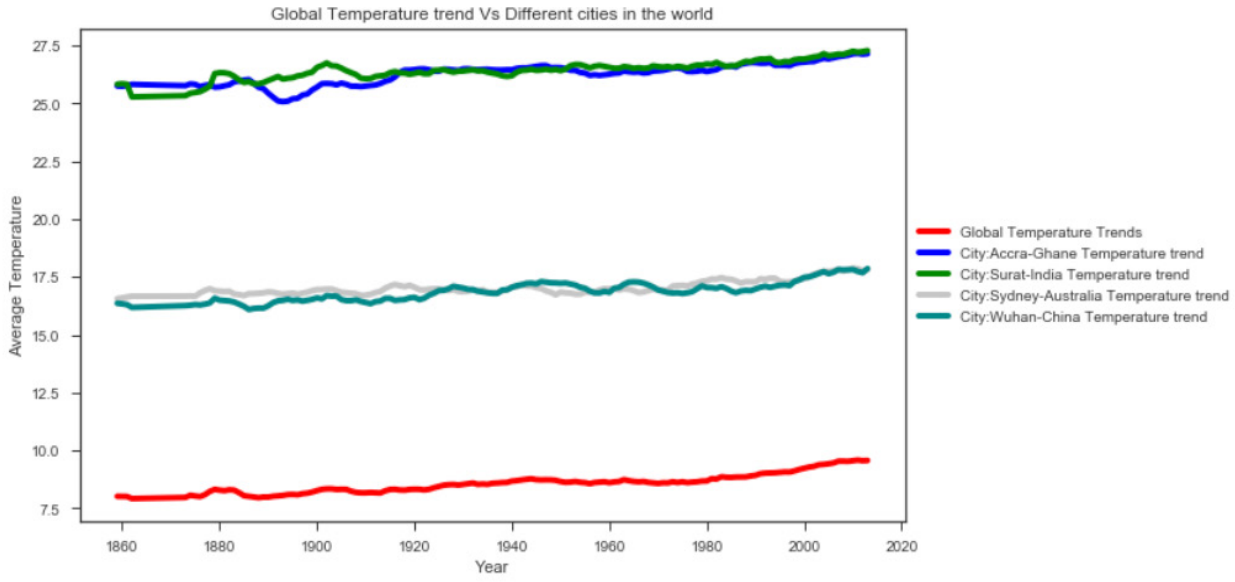


Fig 1-20

A. The above line plot show temperature trends for "global," "Accra - Ghana, Africa,"Surat - India, Asia," "Sydney-Australia," and "Wuhan – China, Asia." over two centuries. There appears to be temperature rising trend.

4. EXTRA

Correlation

```
sns.heatmap(df_final_mean_rolling_7days_mean.corr(),
            annot=True,
            cmap='BrBG',
            linewidths=0.2)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x294e6cb00b8>
```

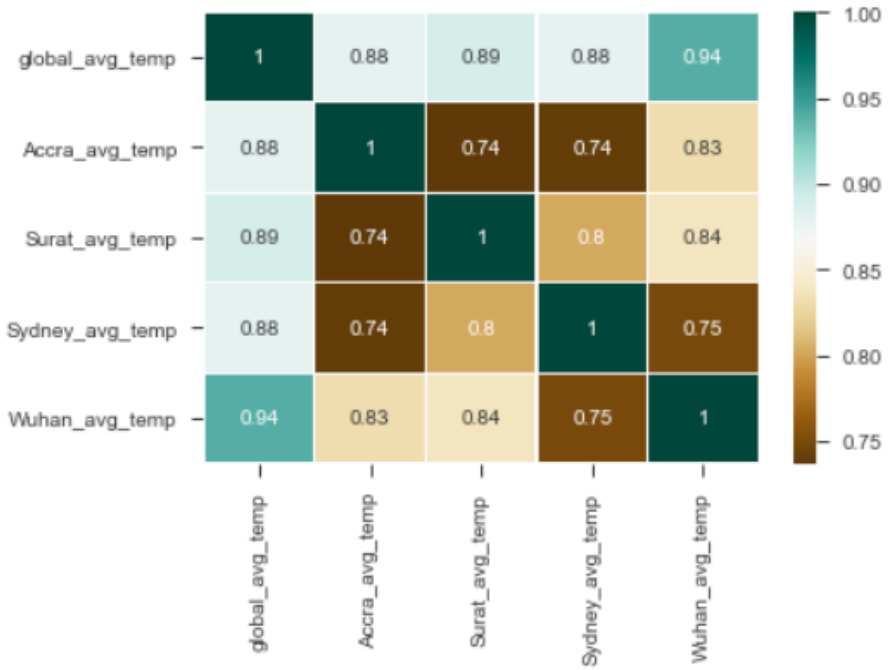


Fig 1-21

B. The value of correlation coefficient(r) is strong i.e. a positive linear relationship between
   1. "global_avg_data" and "Wuhan_avg_temp" is 0.94.
   2. "global_avg_data" and "Sydney_avg_temp" is 0.88
   3. "global_avg_data" and "Surat_avg_temp" is 0.90
   4. "global_avg_data" and "Accra_avg_temp" is 0.88

Ref : https://www.youtube.com/watch?v=4EXNedimDMs

4. EXTRA

```
sns.lmplot(x='global_avg_temp',
           y='Accra_avg_temp',
           data=df_final_mean_rolling_7days_mean
             )
plt.title('Covariance plot average temprature: Global Vs City Accra-
Ghana\n',fontsize=20,y=1.08,horizontalalignment='center',color='red')
plt.xlabel('Global average temprature')
plt.ylabel('City Accra-Ghana average temprature')
plt.legend()
```

```
No handles with labels found to put in legend.
```

```
<matplotlib.legend.Legend at 0x294e8aa0780>
```

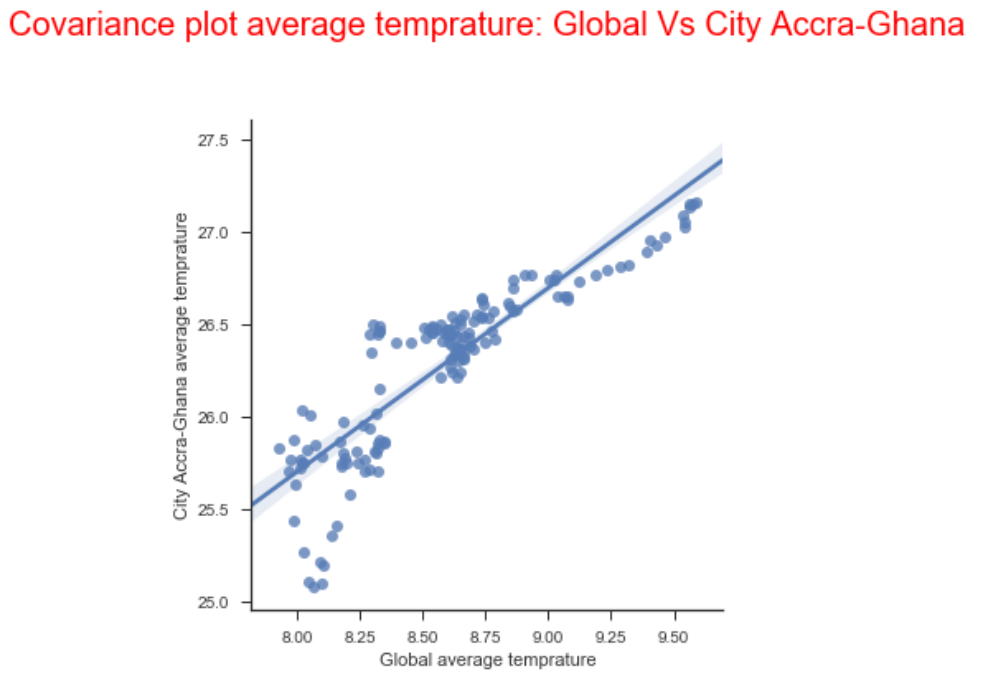


Fig 1-22

C. There is a positive linear covariance between Global temperature and city Accra - Ghana, Africa temperature as shown above. This means that as Global temperature rises so does at city Accra - Ghana, Africa.

Reference help:https://www.youtube.com/watch?v=xGbpuFNR1ME&t=4s

4. EXTRA

```
sns.lmplot(x='global_avg_temp',
           y='Surat_avg_temp',
           data=df_final_mean_rolling_7days_mean
             )
plt.title('Covariance plot average temprature: Global Vs City Surat-
India\n',fontsize=20,y=1.08,horizontalalignment='center',color='red')
plt.xlabel('Global average temprature')
plt.ylabel('City Surat-India average temprature')
plt.legend()
```

```
No handles with labels found to put in legend.
```

```
<matplotlib.legend.Legend at 0x294e8a84710>
```

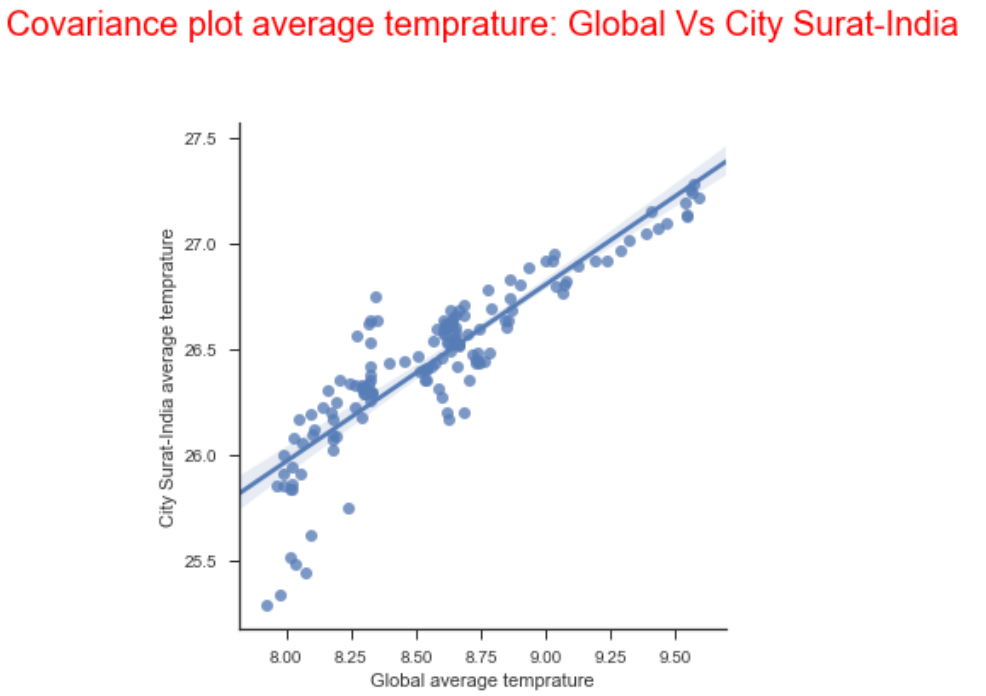


Fig 1-23

D. There is a positive linear covariance between Global temperature and city Surat - India, Asia temperature as shown above. This means that as Global temperature rises so does at Surat - India, Asia.

Reference help:https://www.youtube.com/watch?v=xGbpuFNR1ME&t=4s

4. EXTRA

```
sns.lmplot(x='global_avg_temp',
           y='Sydney_avg_temp',
           data=df_final_mean_rolling_7days_mean
             )
plt.title('Covariance plot average temprature: Global Vs City Sydney-
Australia\n',fontsize=20,y=1.08,horizontalalignment='center',color='red')
plt.xlabel('Global average temprature')
plt.ylabel('City Sydney-Australia average temprature')
plt.legend()
```

```
No handles with labels found to put in legend.
```

```
<matplotlib.legend.Legend at 0x294e8b0bf28>
```

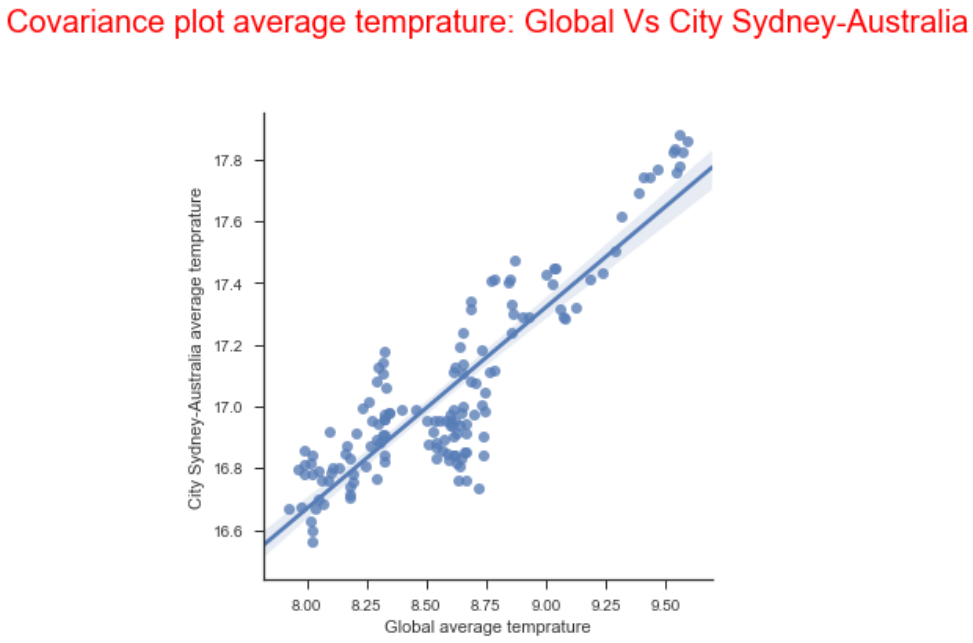


Fig 1-24

E. There is a positive linear covariance between Global temperature and city Sydney-Australia, temperature as shown above. This means that as Global temperature rises so does at Sydney-Australia .

Reference help:https://www.youtube.com/watch?v=xGbpuFNR1ME&t=4s

4. EXTRA

```
sns.lmplot(x='global_avg_temp',
           y='Wuhan_avg_temp',
           data=df_final_mean_rolling_7days_mean
             )
plt.title('Covariance plot average temprature: Global Vs City Wuhan-
China\n',fontsize=20,y=1.08,horizontalalignment='center',color='red')
plt.xlabel('Global average temprature')
plt.ylabel('City Wuhan-China average temprature')
plt.legend()
```

```
No handles with labels found to put in legend.
```

```
<matplotlib.legend.Legend at 0x294e8b50160>
```

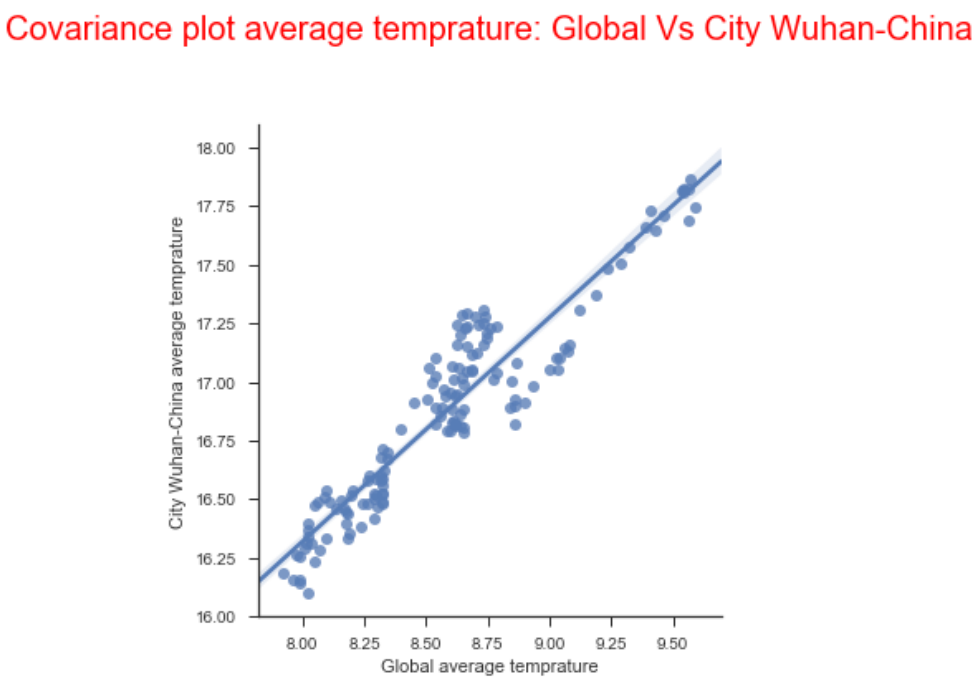


Fig 1-25

F. There is a positive linear covariance between Global temperature and city Wuhan – China, Asia, temperature as shown above. This means that as Global temperature rises so does at Wuhan – China, Asia .

Reference help:https://www.youtube.com/watch?v=xGbpuFNR1ME&t=4s

4. EXTRA

Average Temperature : Global Vs cities

```
df_final_mean_rolling_7days_mean.describe()
```

	global_avg_temp	Accra_avg_temp	Surat_avg_temp	Sydney_avg_temp	Wuhan_avg_temp
count	145.000000	145.000000	145.000000	145.000000	145.000000
mean	8.580680	26.279596	26.453734	17.048108	16.873911
std	0.410866	0.463643	0.387001	0.304628	0.418177
min	7.925714	25.081429	25.284286	16.562857	16.098571
25%	8.290000	25.857143	26.270000	16.841429	16.515714
50%	8.602857	26.418571	26.467143	16.954286	16.891429
75%	8.744286	26.550000	26.642857	17.192857	17.130000
max	9.588571	27.160000	27.284286	17.881429	17.865714

Fig 1-26

E. Above table show mean, standard deviation for “global,” “Accra - Ghana, Africa, ”Surat - India, Asia,” “Sydney-Australia,” and “Wuhan – China, Asia.” over two centuries.

THANK YOUR VERY MUCH