UDACITY

‹ Return to Classroom

DISCUSS ON STUDENT HUB ›

Explore Weather Trends

REVIEW

HISTORY

Meets Specifications

Congratulations

You have met all of the requirements for this project! This is an excellent submission.

Your work clearly highlights the difference between global temperatures and temperatures in Lagos, and your observations highlight key insights from your visualization. Excellent work!

As noted below, especially interesting is: *"the global temperature has a wider range (4.05 deg. C) than my city temperature (2.93 deg. C)"*

Congrats again and best wishes for your next project

---

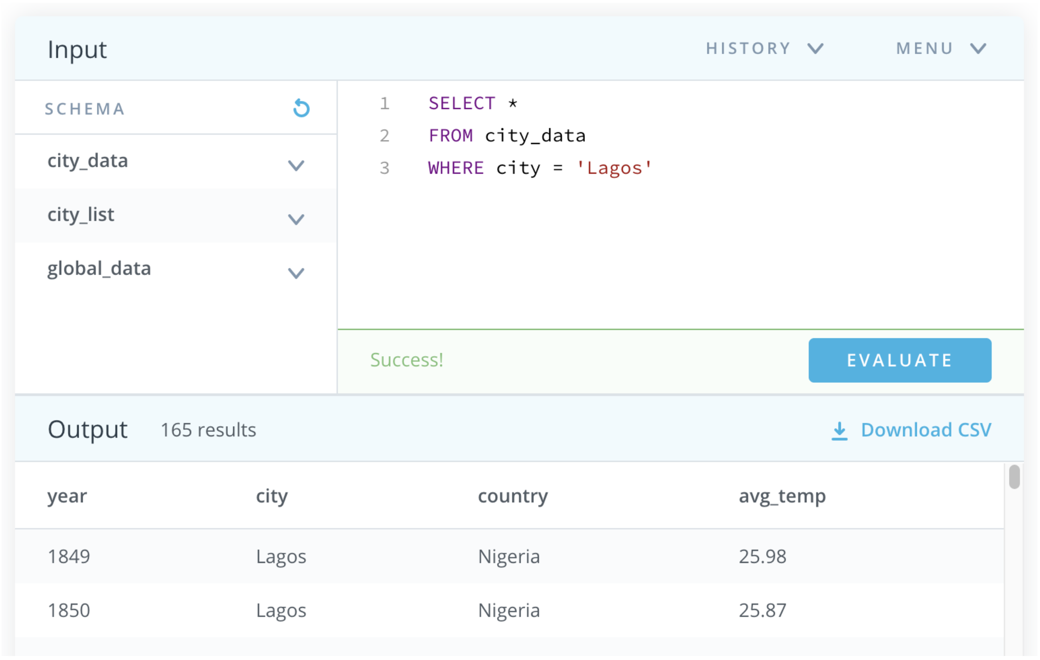
p.s. If you wish, you can continue your exploration of SQL queries by working through the examples on this site.

Analysis

✓

- The SQL query used to extract the data is included.
- The query runs without error and pulls the intended data.

Great work here in extracting the data for Lagos and comparing that to global temperatures. Your queries were spot on!



```sql
SELECT *
FROM city_data
WHERE city = 'Lagos'
```

year	city	country	avg_temp
1849	Lagos	Nigeria	25.98
1850	Lagos	Nigeria	25.87

ADDITIONAL NOTES

.FILLNA()

You use `.fillna()` to replace missing values:

```python
#replacing missing data with mean:
for i in ['avg_temp']:
    lagos_temp[i] = lagos_temp[i].fillna(value = lagos_temp[i].mean())
```

Kudos for using advanced methods in pandas!!

However, dealing with missing values would be a course on its own. It is a complex process.

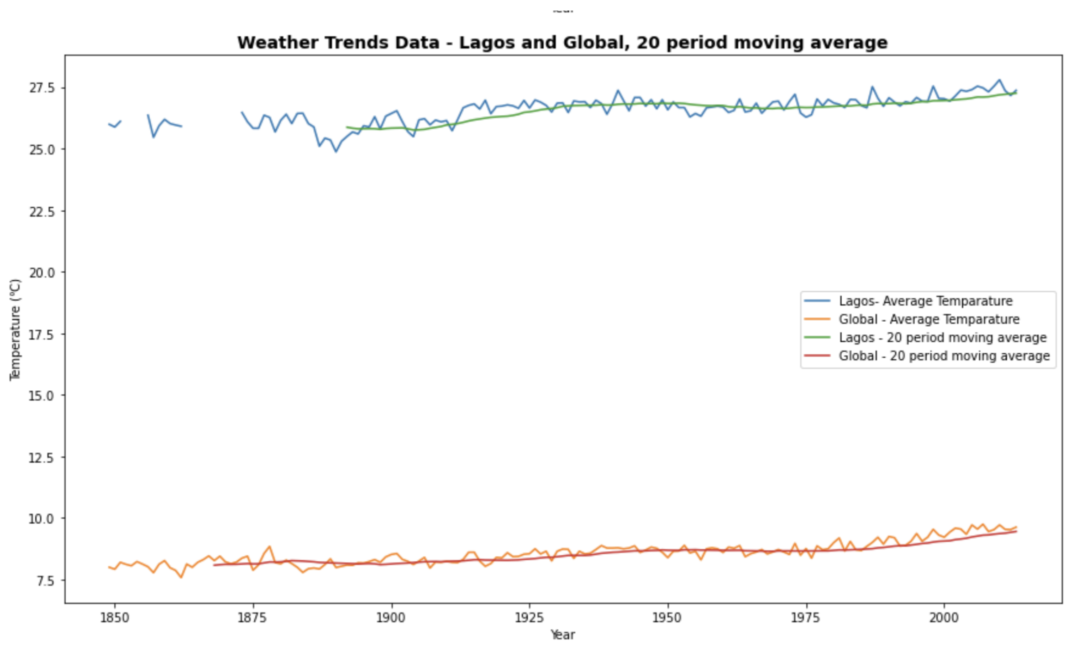
The most important advice from such a course would be: Avoid replacing missing values unless you can validate (i.e. check) the method that you are using. It typically leads to biased results. (given that this is a complicated process, which is not expected for this project, avoid using `.fillna()` . Here is a discussion of the problems associated with this approach.

✓

Moving averages are calculated to be used in the line chart.

Excellent work here in calculating 20 period moving averages for both Lagos and Global temperatures.

A 20 period moving average was a good choice, as you can see when you compare to the original values, most of the *"noise"*, from year-to-year changes, have been smoothed out - leaving a clear trend:



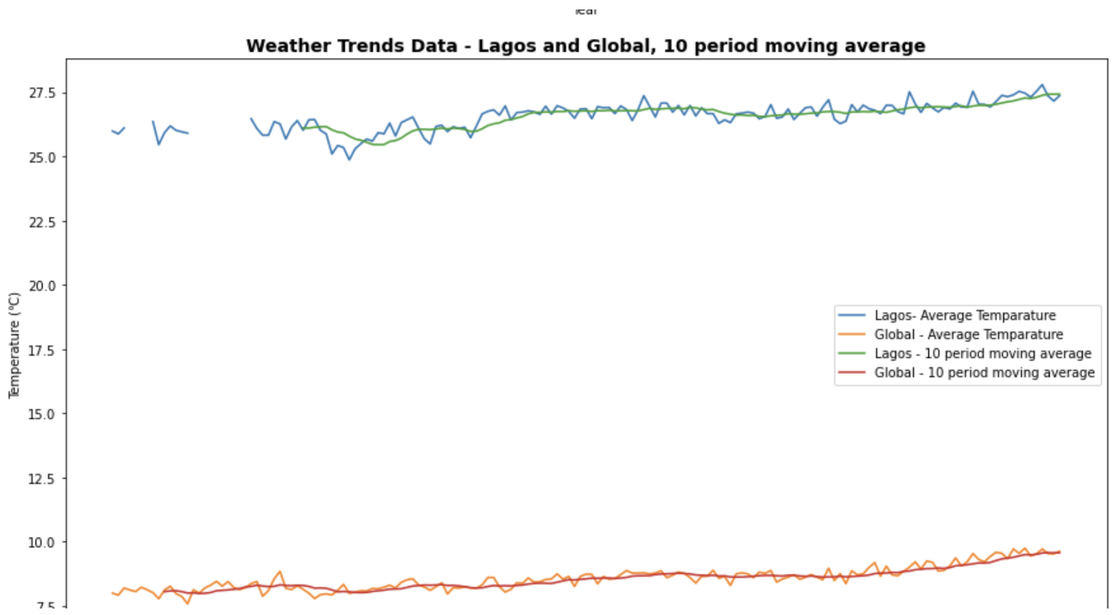
TIP *(NOT REQUIRED)*

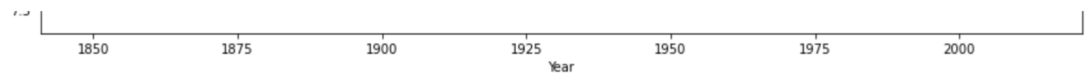
Selecting the correct number of periods for the moving average is a trade-off:

- Removing noise (leaving a trend line), vs
- Losing data (as the number of periods increase)

It is best to experiment with the number of periods in your moving average.

As noted above, you want to ensure that most of the *"noise"*, from year-to-year changes, have been smoothed out - leaving a clear trend. For example, for Lagos, I found that a 10 year moving average worked as well:





ADDITIONAL NOTES

Again, kudos for using vectorized methods in pandas:

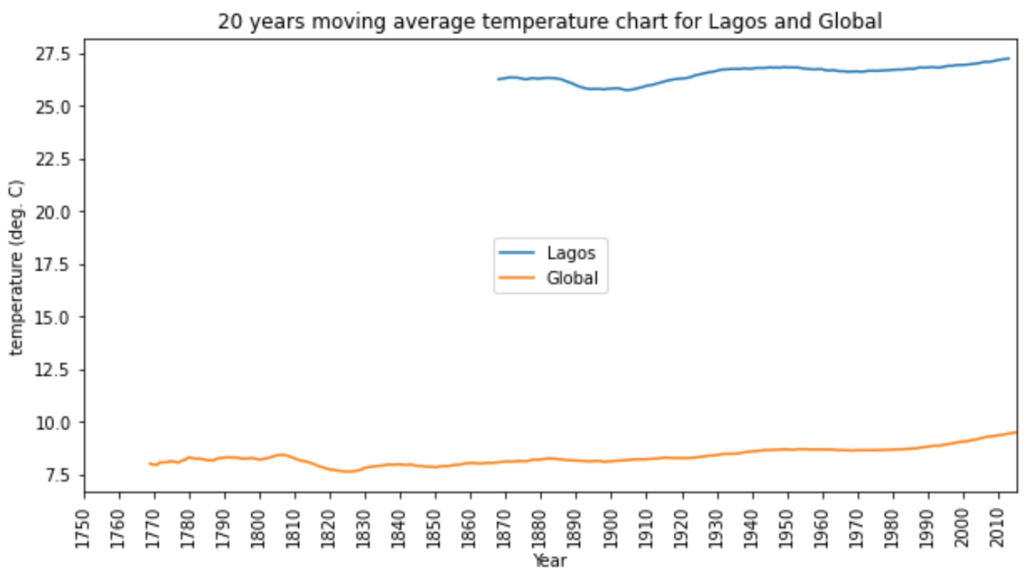
```
#to calculate 20years, 100years moving average temperature
lagos_temp['20years MA_lagos (deg. C)'] = lagos_temp.avg_temp.rolling(window=20).
mean() lagos_temp['100years MA_lagos (deg. C)'] = lagos_temp.avg_temp.rolling(win
dow=100).mean()
```

✓

- A line chart is included in the submission.
- The chart and its axes have titles, and there's a clear legend (if applicable).

The line chart included in your submission looks fantastic! It is appropriately titled, the axes are labeled and the legend is easy to interpret.

```
Text(0, 0.5, 'temperature (deg. C)')
```



ADDITIONAL NOTES

When communicating information using visualizations, the visualizations have to be *self-contained*. That is, the reader should be able to look at the chart and immediately understand what each part of the visualizations represents. This includes:

- A **descriptive** title (for example: `"Lagos and Global Temperatures: (10 Year Moving Average)"`

- A legend with text (so that the reader knows which line represents which variable)
- Full descriptive axes labels
- Units of measurement on the axes

Obviously, you have supplied most of those details

TIPS (NOT REQUIRED)

- Avoid angling tick labels. It makes visualizations difficult to interpret *(Imagine the viewer/reader angling their heads at the same time as trying to take in the rest of the information in the visualizations).*
- All messages when evaluating code should be suppressed.
  - This is done by *ensuring that the final statement in each code cell* ***finishes*** *with a* `;` (that is all that is required).
  - That is, if the last statement in a cell is `plt.xlabel('Credit')` you change that to `plt.xlabel('Credit');`

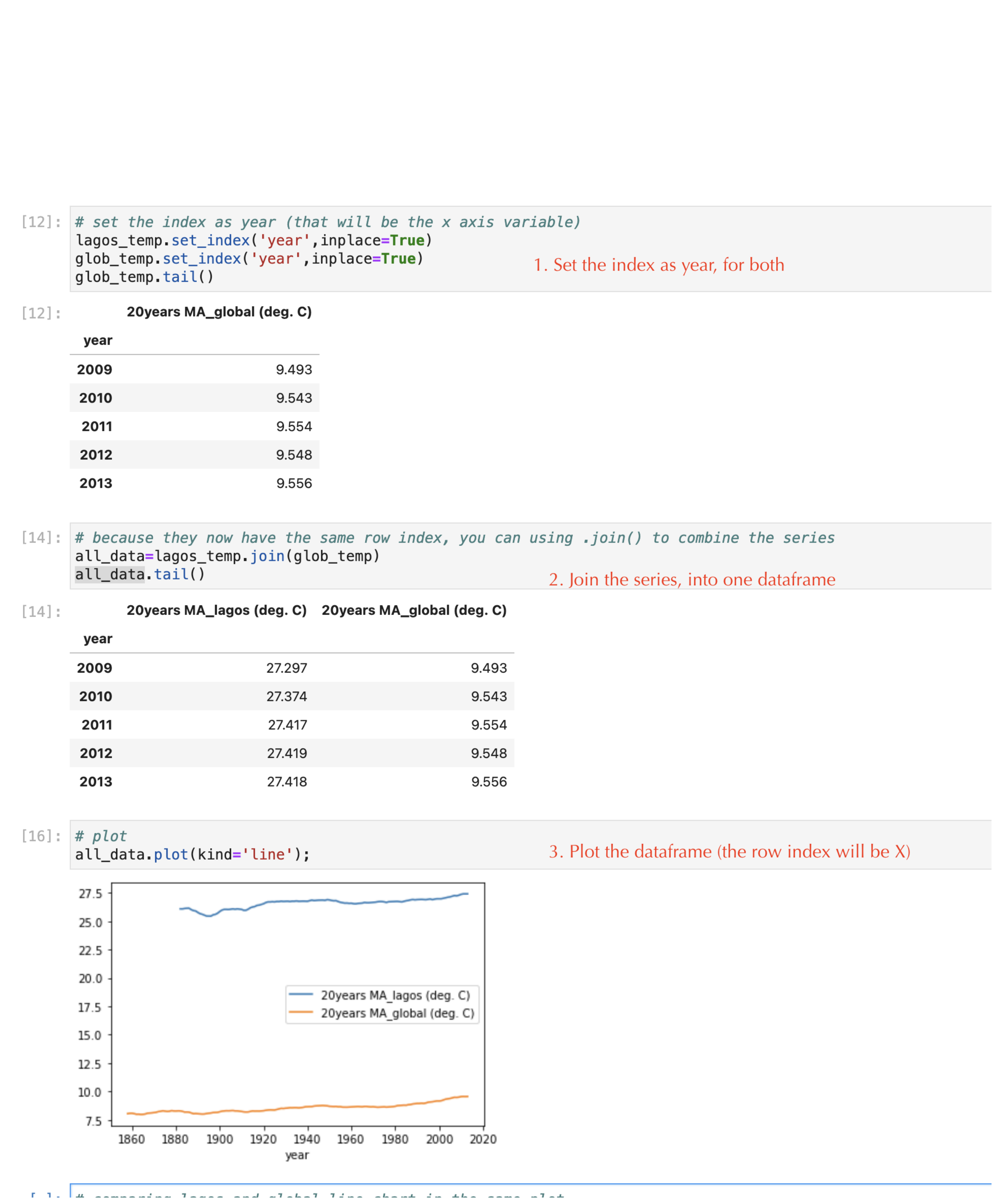
SUBMISSION QUESTION

> "WHAT IS THE BEST WAY TO PLOT THE TEMPERATURE DATA"

The method that you used is excellent!!

In later courses you will get much more experience creating visualizations.

The simplest way is to plot the data as columns of a dataframe:



```
[12]: # set the index as year (that will be the x axis variable)
      lagos_temp.set_index('year',inplace=True)
      glob_temp.set_index('year',inplace=True)
      glob_temp.tail()
```

1. Set the index as year, for both

[12]:

	20years MA_global (deg. C)
year	
2009	9.493
2010	9.543
2011	9.554
2012	9.548
2013	9.556

```
[14]: # because they now have the same row index, you can using .join() to combine the series
      all_data=lagos_temp.join(glob_temp)
      all_data.tail()
```

2. Join the series, into one dataframe

[14]:

	20years MA_lagos (deg. C)	20years MA_global (deg. C)
year		
2009	27.297	9.493
2010	27.374	9.543
2011	27.417	9.554
2012	27.419	9.548
2013	27.418	9.556

```
[16]: # plot
      all_data.plot(kind='line');
```

3. Plot the dataframe (the row index will be X)

```
[ ]: # comparing lagos and global line chart in the same plot
```

Click On Images To Enlarge Them

So that the plot statement is a single line:

1. You don't need to set the tick labels
2. Pandas will automatically space the tick labels (kudos for using `[::10]` to space them!!).

You can then add titles and labels

You could also add an **additional** plot.

Because there is such a gap between the series, it is difficult to see changes in the trend lines.

You can create a **twin axis** *(so that the plot shares a single x axis)*, using each individual y axis for one of the series.

There are only two steps involved:

```
# create the first axes
fig, ax = plt.subplots(1, figsize=(11.69, 8.27), sharex=True);
```

```
# twin the first axes - specifying that you want a common x axis
ax1=ax.twinx()
```

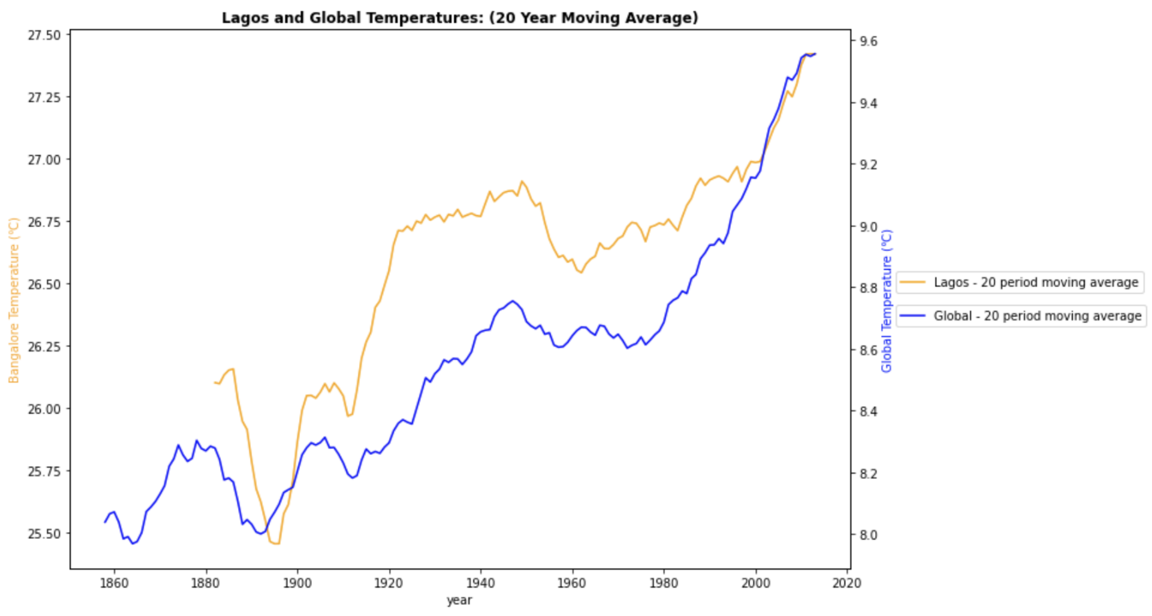
```
20]: fig, ax = plt.subplots(1, figsize=(11.69, 8.27), sharex=True);

     all_data['20years MA_lagos (deg. C)'].plot(ax=ax, color='orange', label='Lagos - 20 period moving average')
     ax.set_ylabel('Bangalore Temperature (℃)',color='orange')

     ax1=ax.twinx()

     all_data['20years MA_global (deg. C)'].plot(ax=ax1, color='b', label='Global - 20 period moving average')
     ax1.set_ylabel('Global Temperature (℃)', color='b')

     plt.title("Lagos and Global Temperatures: (20 Year Moving Average)",weight='bold');
     ax1.legend(loc='upper left', bbox_to_anchor=(1.05, 0.5));
     ax.legend(loc='lower left', bbox_to_anchor=(1.05, 0.5));
```



You would then include both visualizations, one for the overview, the other for the detailed view.

✓

- **The student includes four observations about their provided data visualization.**
- **The four observations are accurate.**

You have analyzed the visualization of the moving averages and made 4, accurate and interesting, observations based on the patterns that you saw.

Especially interesting is:

> "the global temperature has a wider range (4.05 deg. C) than my city temperature (2.93 deg. C)"

SUBMISSION QUESTION

> "WHAT IS THE IMPLICATION OF 0.78 CORRELATION COEFFICIENT"

Correlation is a measure of linear association *(basically: If x is above/below its average, what proportion of the time is x above/below its average)*

Any value above 0.7 is a very strong correlation (it is rare to find such a strong correlation).

Here is an excellent summary

DOWNLOAD PROJECT

RETURN TO PATH