Thursday, November 24, 2020 | **Class #15**

`SciPy` (2-D interpolation), `pandas` (correlating, resampling, and smoothing data)

OCEAN 215 | Autumn 2020

Ethan Campbell and Katy Christensen

Calculating correlations using `.corr()`

Recall that a column of a Pandas DataFrame is a Pandas Series.

We can correlate two Pandas Series objects, `s1` and `s2`, using the following syntax:

```
s1.corr(s2)
```

The result is the standard Pearson correlation coefficient, r.

More commonly, however, you'll use r^2. As I introduced in Lesson #14, r^2 represents the proportion of variance in one variable that is explained by the other variable.

API documentation: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.corr.html

Changing the time resolution using `.resample()`

We can down-sample the time resolution of a Pandas Series or DataFrame if it has datetimes as its index, i.e. a `DatetimeIndex`.

The function `.resample()` takes a "frequency alias" or "offset alias" as its argument. It behaves similar to `.groupby()`, so after you group the data, you have to apply a function like `.mean()`, `.max()`, or `.sum()`.

Here are the available frequency aliases: https://pandas.pydata.org/docs/user_guide/timeseries.html#offset-aliases

Some common options are:

- 'H': hourly frequency
- 'D': daily frequency
- 'W': weekly frequency
- 'MS': monthly frequency (use start of month as resulting date)
- 'YS': yearly frequency (use start of year as resulting date)

API documentation: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.resample.html

Smoothing data using `.rolling()` averages

A common technique to smooth a time series (or other data) is to calculate a rolling average, also called a running average, running mean, or moving average:

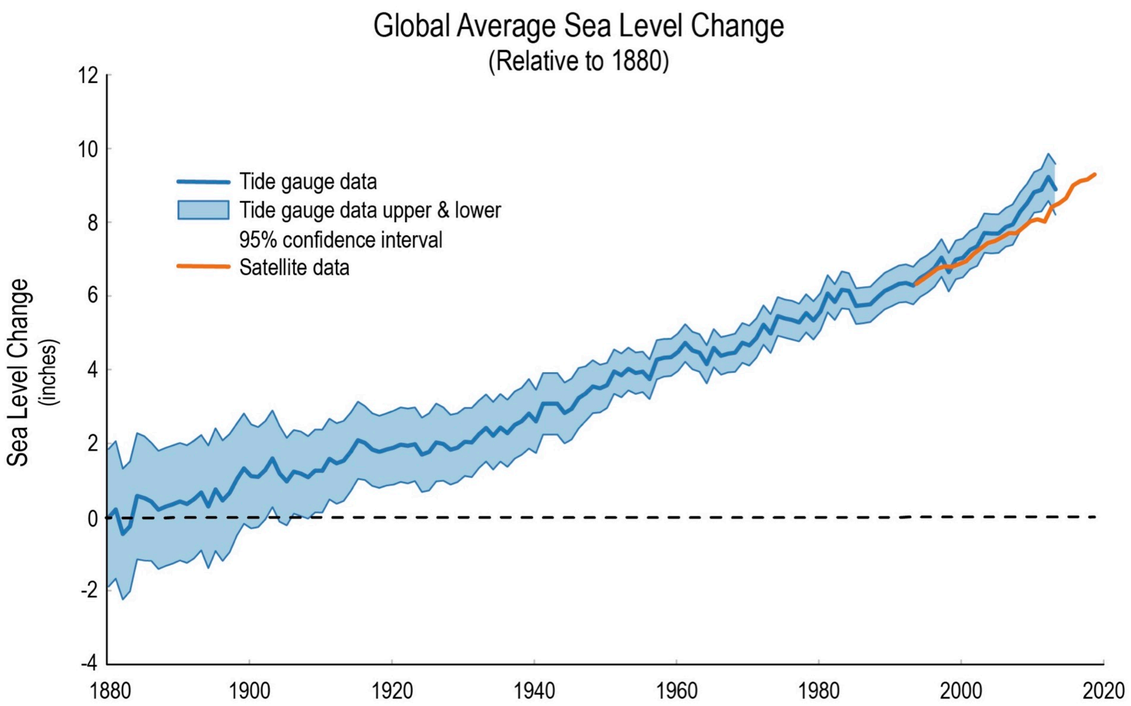
```
.rolling(window,min_periods=1,center=False)
```

- `window` specifies the size of the moving window in number of rows
- `min_periods` specifies the minimum number of rows required to have data (otherwise the result is np.NaN); this is important at the start and end of the time series
- if `center` is True (recommended), the date will be set to center of each window; if False, the date will be set to the end of each window

Again, `.rolling()` behaves similar to `.groupby()` and `.resample()` in that you have to apply a function like `.mean()` or `.median()` to the grouped data to get a result.

API documentation: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.rolling.html

Activity: detecting sea level rise from Florida tide gauge records



Source: *U.S. Global Change Research Program (USGCRP)*

Image: *Miami Beach faces more frequent sunny-day flooding during king tides due to sea level rise. (South Florida Sun Sentinel)*

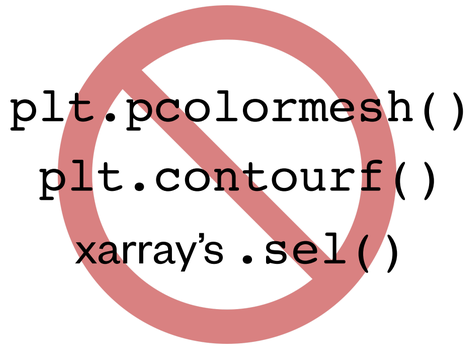
Google Doc with activities (also accessible from Canvas Modules or Google Drive folder):

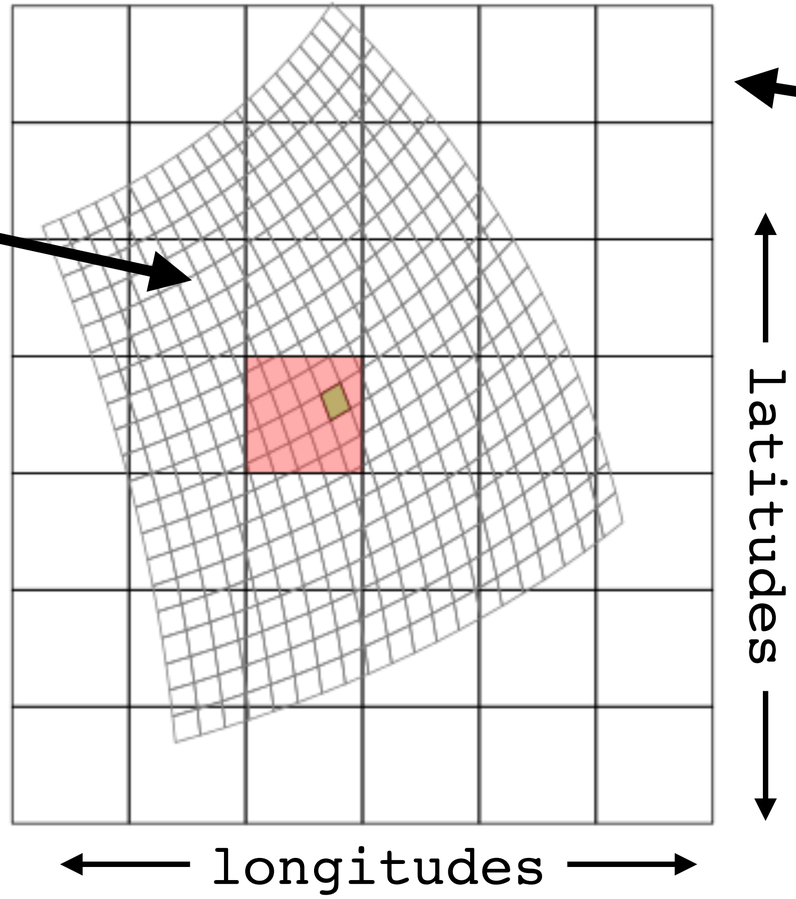
https://tinyurl.com/OCEAN215-Class15

2-D interpolation (a.k.a. 2-D regridding)

You have:

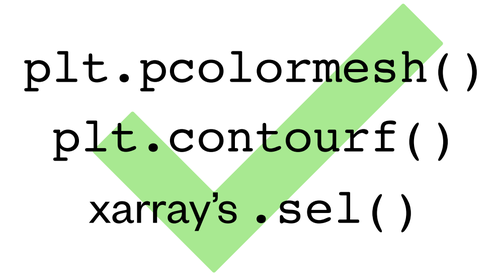
An irregular grid (`lat` and `lon` or other coordinates are usually 2-D arrays)





You want:

A regular grid (`lat` and `lon` can be represented as 1-D coordinates)



For more information on regridding, see Climate Data Guide's "Regridding Overview"

Image credit: Lu et al. (2018)

2-D interpolation in SciPy is a three-step process

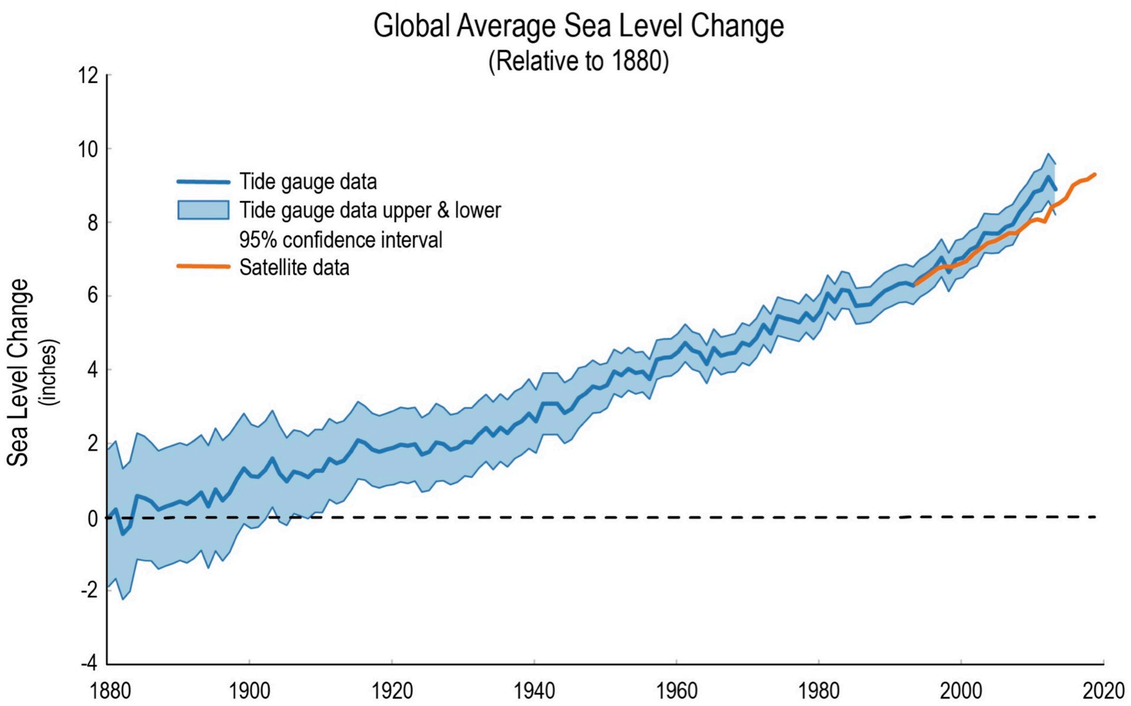
```
x_coord = np.linspace(start,end,num_x_points)
y_coord = np.linspace(start,end,num_y_points)

x_grid,y_grid = np.meshgrid(x_coord,y_coord)

z_gridded = interpolate.griddata((x_flat,y_flat),
                                 z_flat,
                                 (x_grid,y_grid),
                                 method='linear')
```

API references: NumPy meshgrid() and SciPy griddata()

Activity: interpolating scattered global tide gauge measurements



Source: *U.S. Global Change Research Program (USGCRP)*

Image: *Miami Beach faces more frequent sunny-day flooding during king tides due to sea level rise. (South Florida Sun Sentinel)*

Google Doc with activities (also accessible from Canvas Modules or Google Drive folder):

https://tinyurl.com/OCEAN215-Class15