Week 11: What and When

- Today (Tues): code / background for this week's lab: Interpolating data sets.

- Thurs: background and worksheet for assignment 2. I will post assignment 2 on Tuesday. It is due Wednesday November 28, 4pm.

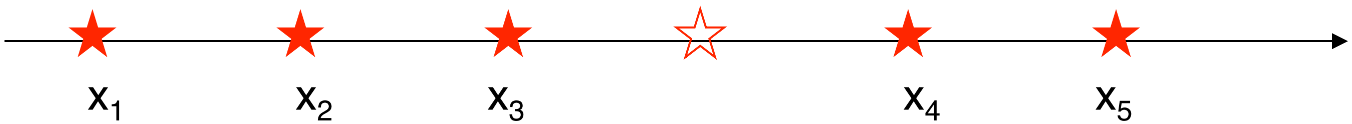
- No new MATLAB this week with exception of using built-in interpolation functions. Lots of review / practice.

- This week (week 11) is the last week with a lab that must be turned in. During weeks 12 and 13 you can use the lab times to work on assignment 2. There will be TAs in the lab but no additional TA hours those weeks.

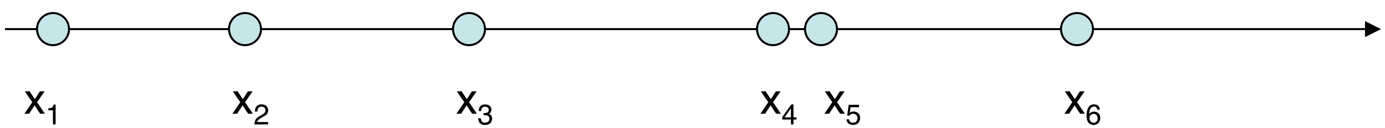
Week 11: Interpolating Data

Common problems in data analysis are

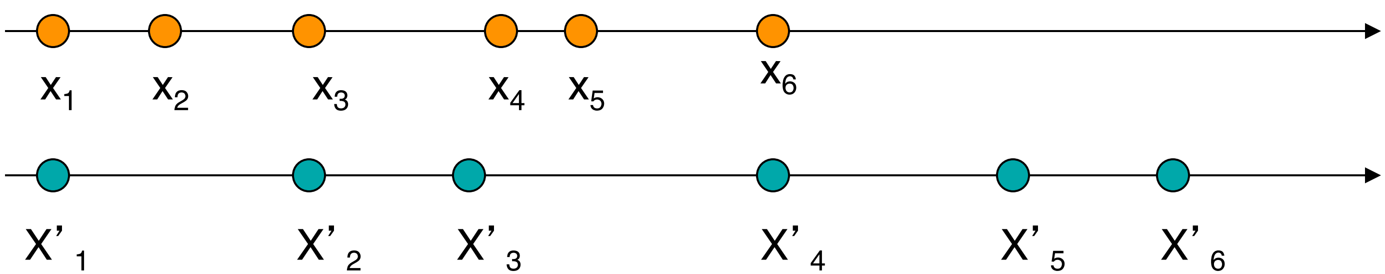
1. Missing data: want to fill in gaps



2. Unevenly-spaced data: would like evenly spaced data



3. Differently spaced data in two data sets: want to compare data from same time, place



This requires ***estimating*** data at locations or times where we ***don't*** have a measurement

Week 11: Interpolating Data

Same underlying issue: want to fill in the gaps or re-grid onto an even sampling interval!

One way to handle this is BINNING your data and then taking the mean / median / mode in each bin.

This is not always practical since don't always have many repeat or nearby measurements

INTERPOLATION: Estimating data between some known measurement points
Done all the time, care needed

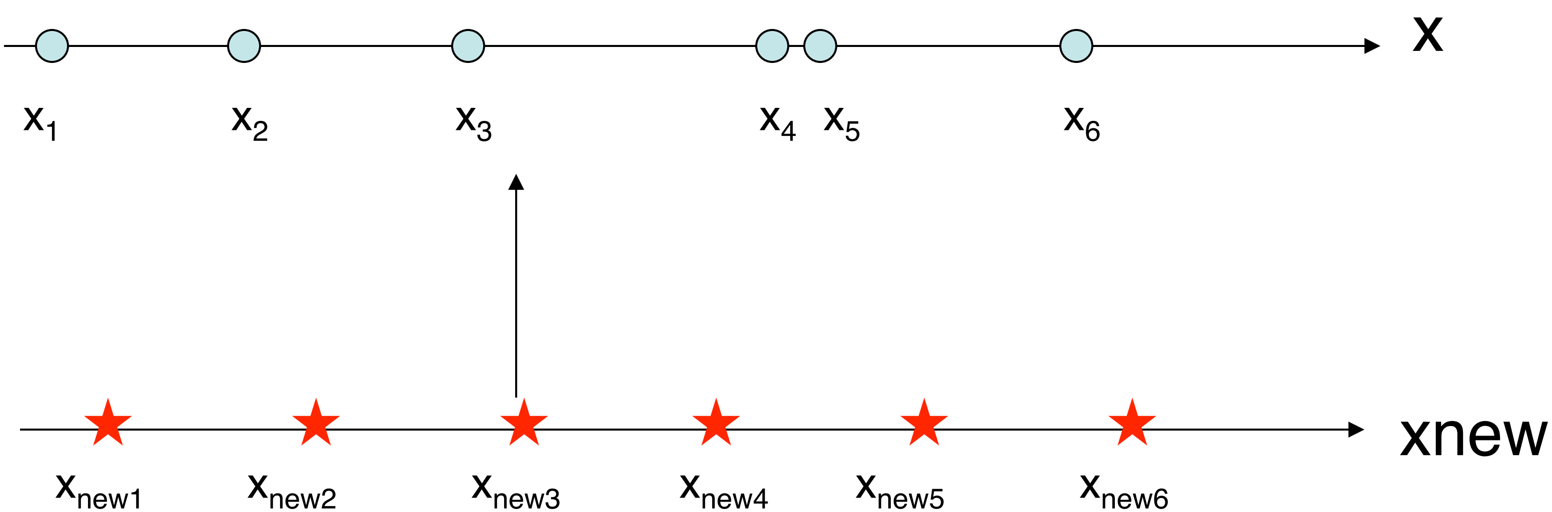
EXTRAPOLATION: Estimating data beyond the end of your measurement set
This is VERY dangerous, and should be avoided

Why interpolate or re-grid data?

1. Comparing / overlaying multiple data sets (maps especially)
2. Create evenly spaced data so we can use our running-mean code for example
3. If we have lots of observations in one time interval and few in another any statistics will be biased toward the time period w/ more observations - want evenly gridded data
4. Doing spectral analysis (fourier transforms etc)

How to do this? The first thing is to define a new, evenly spaced x-axis:

Our original x-axis has a uneven spacing of data points



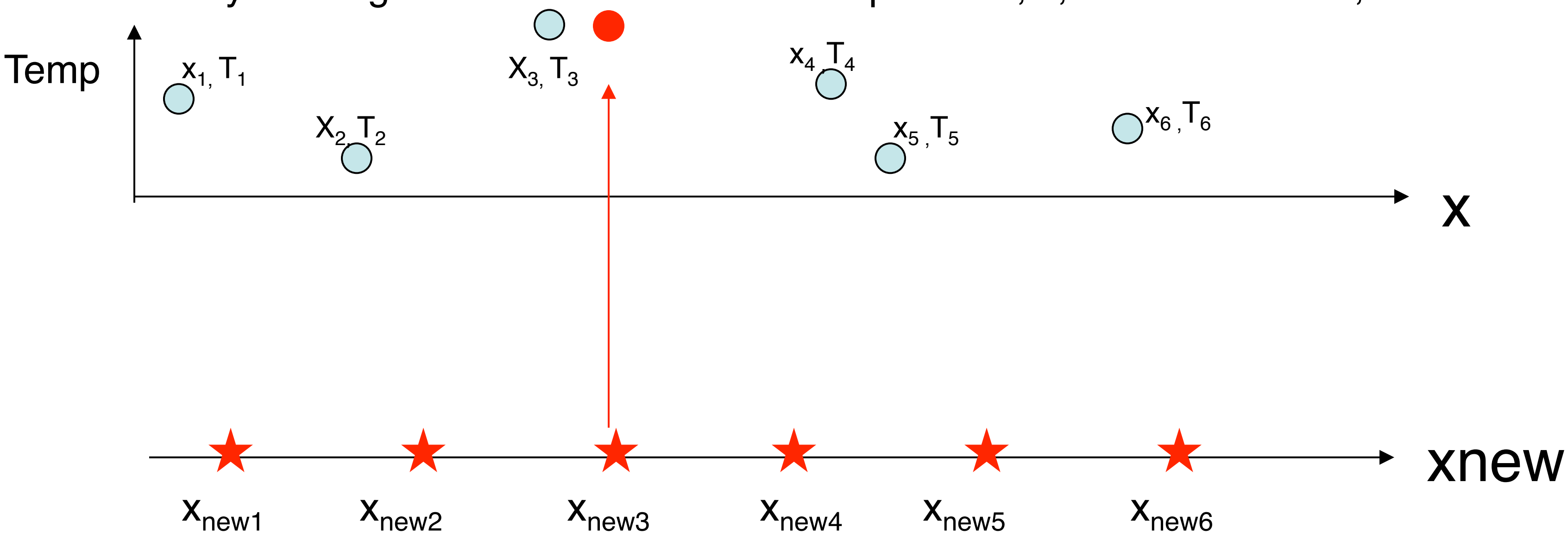
Our new x-axis has an even spacing: $\Delta = x_{new(j+1)} - x_{new(j)}$

Next, we want to estimate our quantity of interest, y, (e.g. temperature) at our new points, x_{new}.

There are many ways to do this……….

- worksheet

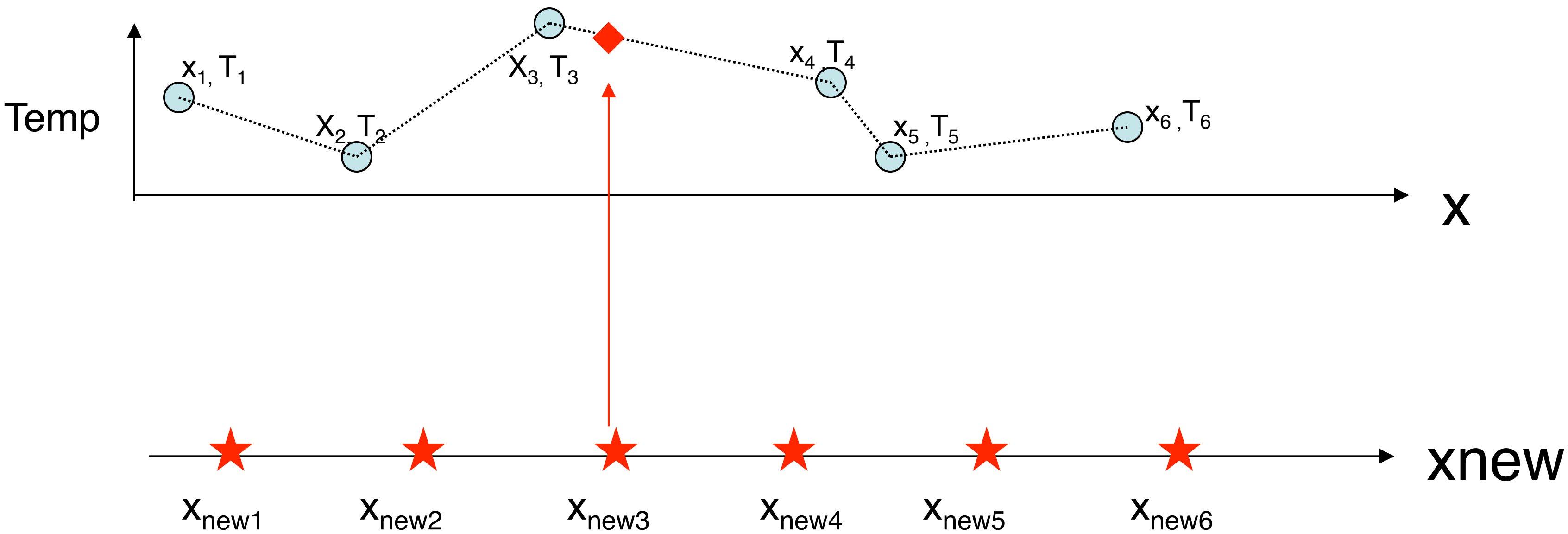
Let's say our original measurements are temperature, T, versus distance, x



We could estimate the temperature at each of our new points, $x_{new(j)}$ by assigning

1. The value of temp at the nearest point $x_{(i)}$ in the original time series,
   e.g., x_3 is closest to x_{new3}, so we could put Temperature(x_{new3}) = T_3

   Name: nearest neighbor interpolation
   Advantage: fast
   Disadvantage: produces a "step-like" function, *i.e.*, discontinuous



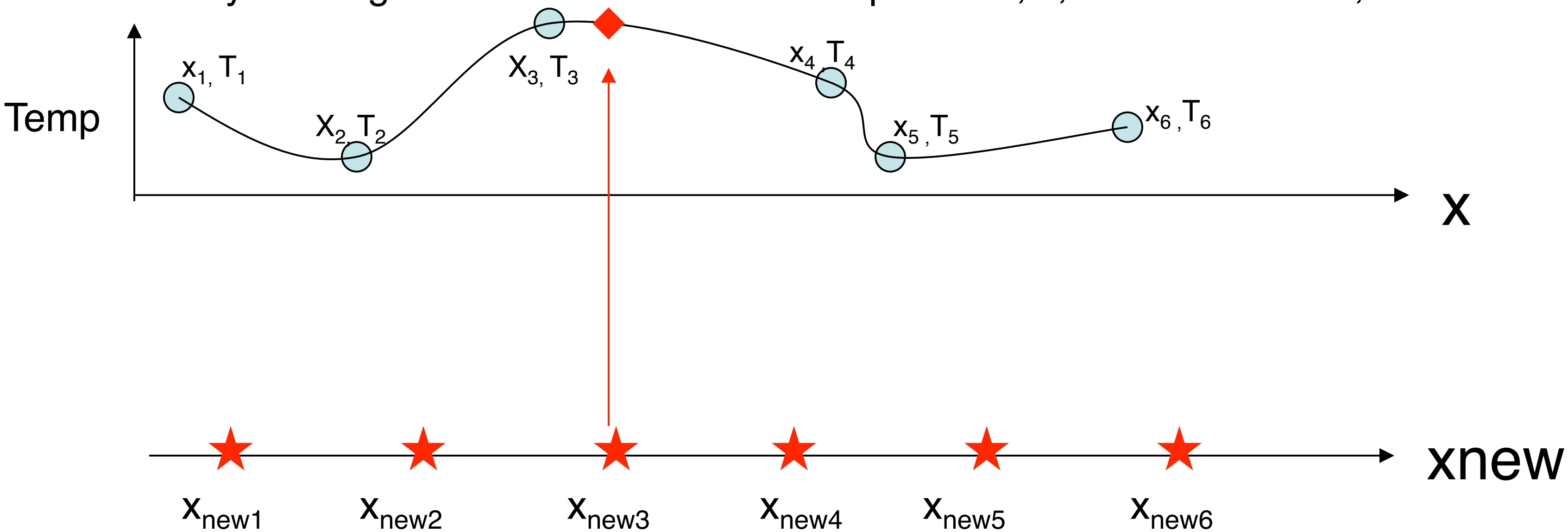
2. We could average the values of temperature at the two points x_3 and x_4

   Not so great, but …..

3. We could add a percentage of the difference T_4-T_3 to the value T_3, and the % would be based on the fractional distance $xnew_3$ is toward x_4 from x_3

   Name: linear interpolation
   Advantage: (1) still pretty fast, (2) produces continuous function
   Disadvantage: "corners" at data points, discontinuous first derivative

Let's say our original measurements are temperature, T, versus distance, x



4. We might want our estimates to be more smoothly varying and fit a polynomial. A better version of this approach involves the use of functions called **splines**.

Name: cubic splines
Advantage: smooth function, continuous second derivative
Disadvantage: slower (not big deal), can produce "overshoots" in large data gaps

Run demoweek11_tues