

Curve fitting and least squares

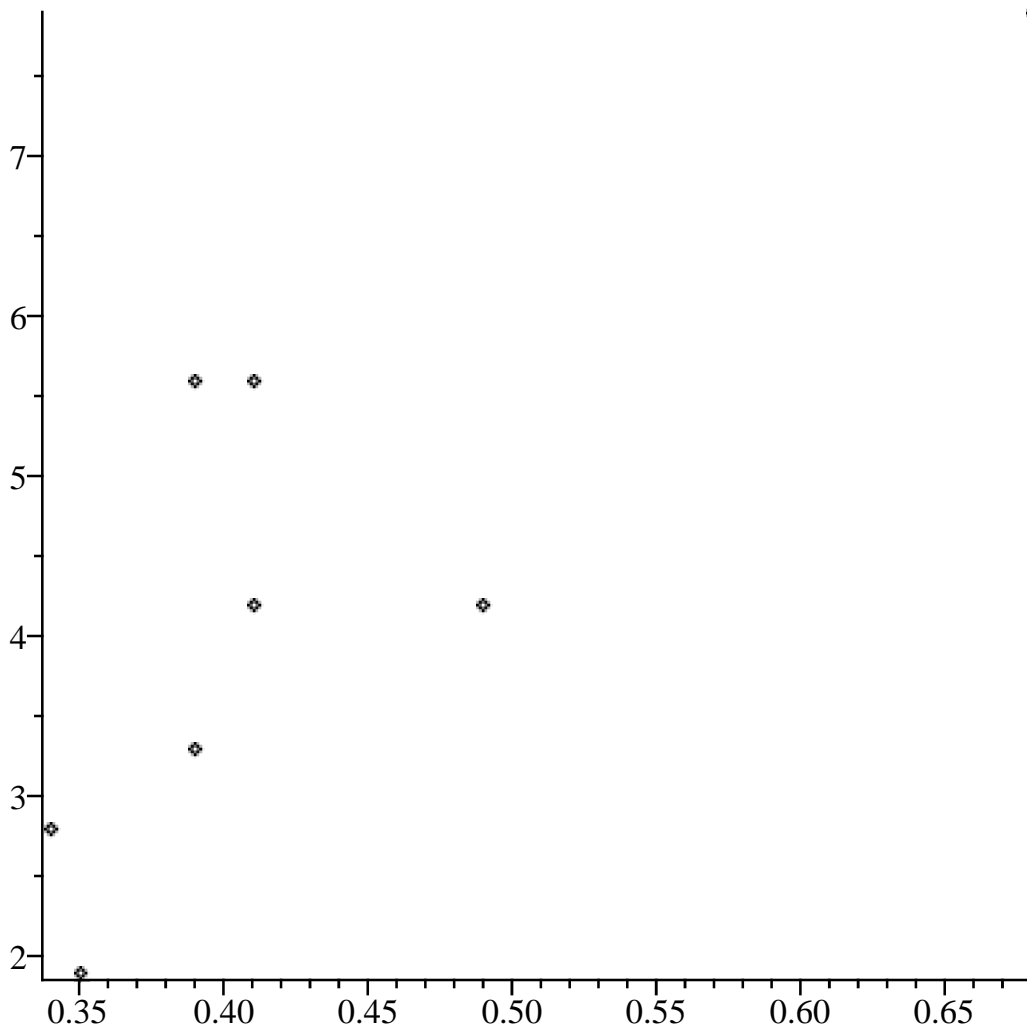
In modeling, we must often start with observed data, and attempt to describe it with some underlying function. Our observed data is never exact, so we must find a curve which fits it well, in some sense.

Example: Manganese (Mn) is important for the health of newborn infants. A paper in the American Journal of Clinical Nutrition gave the following data on Mn intake and serum Mn levels. The first line is the Mn intake per day, in micrograms per kg mass of the infant. The second is the concentration of Mn in the infants' blood, in micrograms per liter.

0.34	0.35	0.39	0.39	0.41	0.41	0.49	0.68
2.8	1.9	3.3	5.6	4.2	5.6	4.2	7.9

It's not easy to see a trend here. We can look at the data graphically using a scatter plot. This is in the "Statistics" package of Maple, so we have to load it, using "with":

```
> with(Statistics) :  
I'm entering the two rows as vectors X and Y.  
> X := [0.34, 0.35, 0.39, 0.39, 0.41, 0.41, 0.49, 0.68];  
      X := [0.34, 0.35, 0.39, 0.39, 0.41, 0.41, 0.49, 0.68] (1)  
> Y := [2.8, 1.9, 3.3, 5.6, 4.2, 5.6, 4.2, 7.9];  
      Y := [2.8, 1.9, 3.3, 5.6, 4.2, 5.6, 4.2, 7.9] (2)  
> ScatterPlot(X, Y);
```



Now suppose that we have reason to think that serum Mn is related to intake Mn in the form $y = mx + b$. What we have to do is find the values of m and b which best fit the data. Let's call the data points (x_i, y_i) , $i = 1, \dots, 8$. The usual method is to find m and b so that that sum of the squares of the difference between the predicted y values and observed y values is minimized. In other words, we want to find m and b so that $E(m, b) = \sum_{i=1}^8 (y_i - mx_i - b)^2$ is a minimum (our example has 8 points, but this works for any number of data points, so let's sum to n). The result is a *linear least squares regression*. There are other possible approaches. We could try minimizing the sum of the distances measured in the y direction, for

example. Then we'd be minimizing $\sum_{i=1}^8 |y_i - mx_i - b|$. The problem with this approach is that the resulting function of m and b is not differentiable, so we can't use calculus to find the minimum. Another idea is to find the line which minimizes the sum of the distances or the squares of the distances from the points to the line (rather than just the distances measured parallel to the y axis. This would be a mess to try to implement. Back to least squares. We look for critical points, by setting the partials equal to zero:

$$\frac{\partial E}{\partial m} = -2 \sum_{i=1}^n x_i (y_i - mx_i - b) = 0$$

$$\frac{\partial E}{\partial b} = -2 \sum_{i=1}^n (y_i - mx_i - b) = 0$$

which is a linear system in the two unknowns m and b . If we rewrite the system as

$$m \sum_{i=1}^n x_i^2 + b \sum_{i=1}^n x_i = \sum_{i=1}^n x_i y_i,$$

$$m \sum_{i=1}^n x_i + n \cdot b = \sum_{i=1}^n y_i,$$

it's not too hard to write down the solution. You can do it explicitly using Cramer's rule:

$$m = \frac{\det \begin{bmatrix} \sum_{i=1}^n x_i y_i & \sum_{i=1}^n x_i \\ \sum_{i=1}^n y_i & n \end{bmatrix}}{\det \begin{bmatrix} \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & n \end{bmatrix}}$$

with a similar formula for b . Of course, this is automated in Maple. It's in the CurveFitting package. Here's how:

```
> with(CurveFitting);
[BSpline, BSplineCurve, Interactive, InteractiveChangeSlider, LeastSquares,
  PolynomialInterpolation, RationalInterpolation, Spline, ThieleInterpolation]
```

```
> LeastSquares(X, Y, x);
-1.610082791 + 13.9828503843879376 x
```

`>`

The syntax is "LeastSquares(Xdata,Ydata,variable)". Use Maple's Help for more information. It would be helpful to plot this on top of the scatter plot. The way to put two or more plots on top of each other is to include the "plots" package, set the plots to be equal to variables (putting a colon at the end, to suppress pages of garbage), and then use the command "display":

`> with(plots);`

Warning, the name `changecoords` has been redefined

Warning, the previous binding of the name `Interactive` has been removed and it now has an assigned value

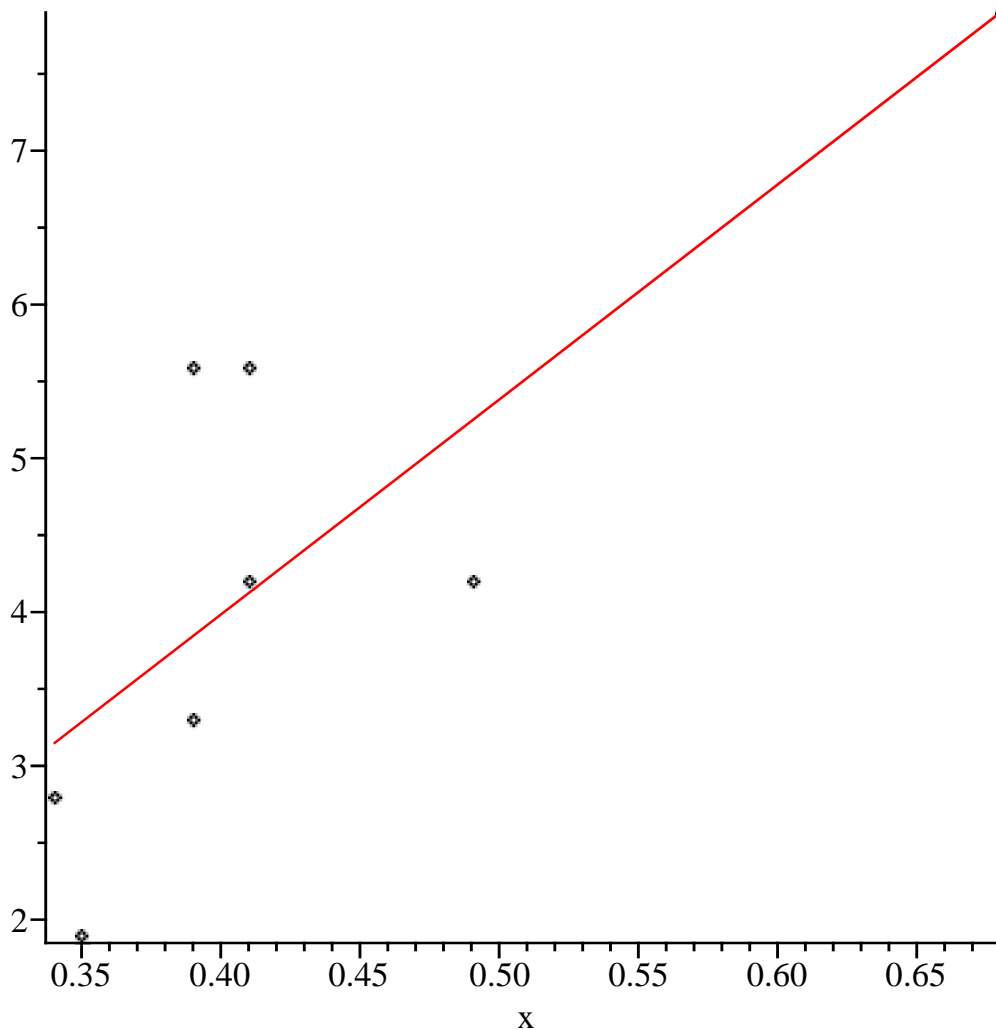
[*Interactive, animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d, conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, cylinderplot, densityplot, display, display3d, fieldplot, fieldplot3d, gradplot, gradplot3d, graphplot3d, implicitplot, implicitplot3d, inequal, interactive, interactiveparams, listcontplot, listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, multiple, odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d, polyhedra_supported, polyhedraplot, replot, rootlocus, semilogplot, setoptions, setoptions3d, spacecurve, sparsematrixplot, sphereplot, surfdata, textplot, textplot3d, tubeplot*]

(5)

`> p1 := ScatterPlot(X, Y) :`

`> p2 := plot(LeastSquares(X, Y, x), x = 0.34 .. 0.68) :`

`> display([p1, p2]);`



It's also possible to fit higher order polynomials to data using least squares. Let's try to fit a quadratic of the form $ax^2 + bx + c$ to the data. Now we'll find the minimum of

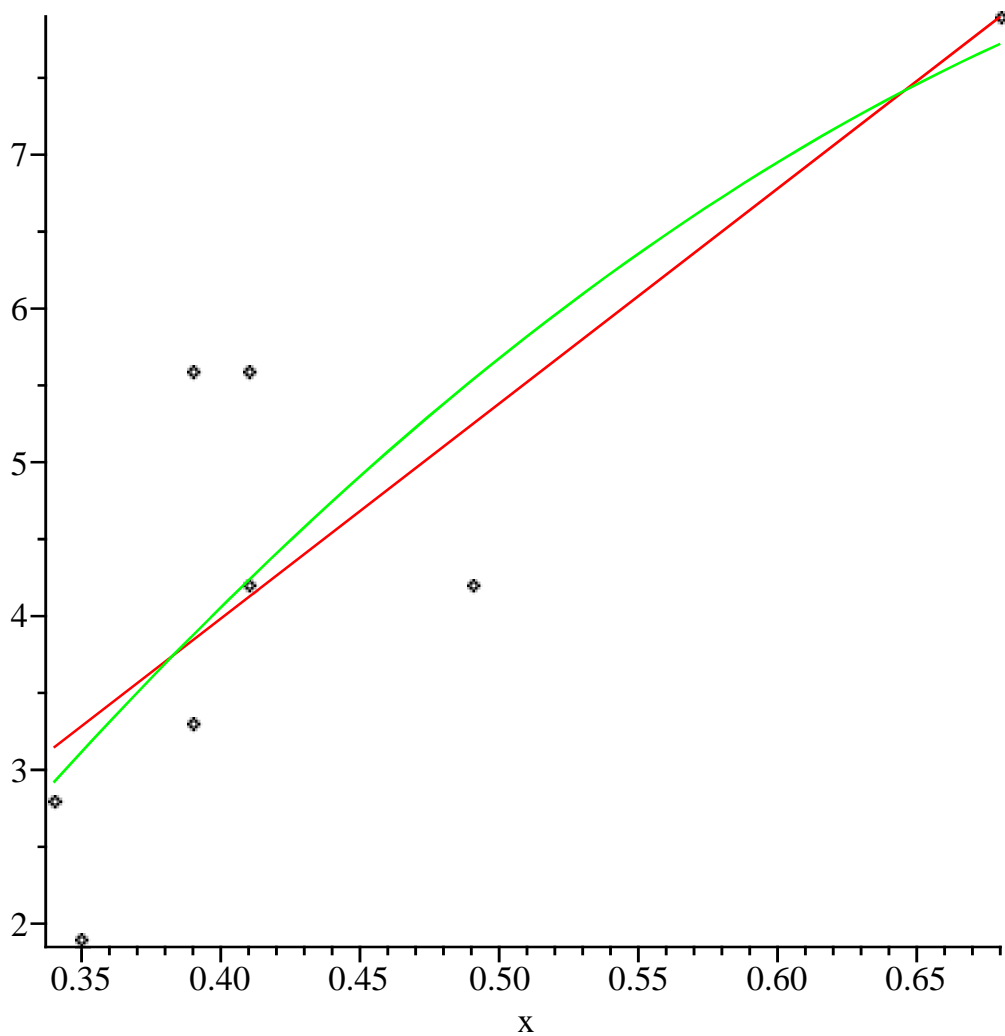
$$E(a, b, c) = \sum_{i=1}^n (y_i - ax_i^2 - bx_i - c)^2$$

by setting the partials with respect to a , b , and c to zero. This will be a linear system in these unknowns. Of course, we can do this in Maple as well:

```
> LeastSquares(X, Y, x, curve = a · x2 + b · x + c);
-5.878270305 + 31.7413311315182512 x - 17.2668205407223994 x2 (6)
```

Just for fun, let's plot this as well:

```
> p3 := plot(LeastSquares(X, Y, x, curve = a · x2 + b · x + c), x = 0.34 .. 0.68, color = green) :
> display([p1, p2, p3]);
```



We can put different forms of curves into the LeastSquares command, but they must be linear in the parameters (not necessarily the variable, as we saw above).