Method of Least Squares

Mth 351 Oct 10 2001 Maple 6

Bent E. Petersen

We can solve linear least squares problems by using either of Maple's linear algebra packages, but the most convenient approach is to use the stats[fit] package.

```
> restart;
> with(stats[fit,describe]):with(stats[statplots]):with(plots):
Warning, the name changecoords has been redefined
```

Suppose we have some experimental data

```
> Xdata:=[1,2,3,4,5,6,7,8,9,10,11,12];
```

$$Xdata := [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]$$

```
> Ydata:=[3,2,2,3,1,4,5,6,7,5,6,8];
```

$$Ydata := [3, 2, 2, 3, 1, 4, 5, 6, 7, 5, 6, 8]$$

and we suspect a functional relationship of the form $y = f(x)$ where f contains some adjustable parameters. The Maple leastsquare function accepts directly the equation that we are trying to fit to our data.

Here's a little procedure to compute the L2 norm of a vector (or 1-array) of arbitrary dimension

```
> N2:=proc(a)
>    local k;
> sqrt(sum(a[k]^2,k=1..nops(a)));
> end:
```

Alternately we could use the norm function from the linear algebra package

```
> # N2:=v->linalg[norm](v,frobenius);
```

It will also be handy to have a plot of the given data points

```
> plt0:=scatterplot(Xdata,Ydata,color=red,thickness=3):
```

Constant function fit

Consider a constant function

```
> f1:=x->c;
```

$$f1 := x \to c$$

```
> eqn1:=leastsquare[[x,y],y=f1(x),{c}]([Xdata,Ydata]);
```

$$eqn1 := y = \frac{13}{3}$$

Thus or our optimal functional relationship of the given type is given by

```
> F1:=unapply(rhs(eqn1),x);
```

$$F1 := \frac{13}{3}$$

The stats[describe] package defines the function mean()

```
> ym:=mean(Ydata);
```

$$ym := \frac{13}{3}$$

We see as expected the best least squares fit is the mean of the Ydata.

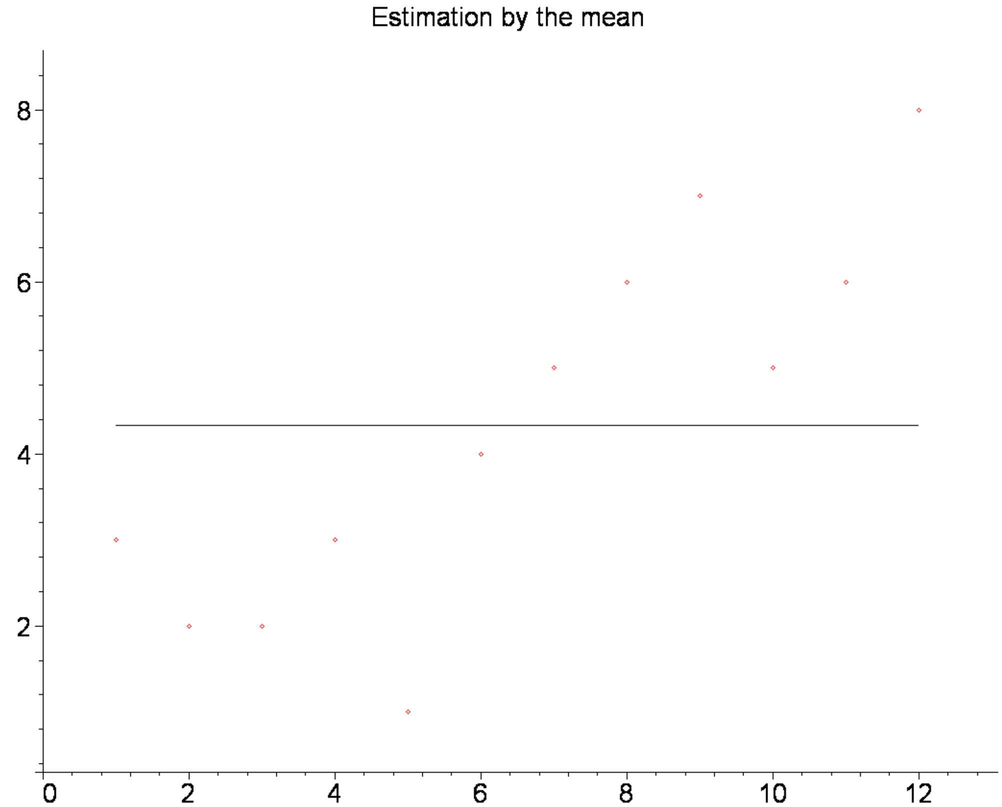
The deviation or total residual is the square root of the sum of the squares of the residuals

```
> ND1:=N2(Ydata-map(F1,Xdata)); evalf(%);
```

$$ND1 := \frac{1}{3}\sqrt{474}$$

$$7.257180353$$

```
> plt1:=plot(F1,1..12,color=black):
> display(plt0,plt1,title = "Estimation by the mean");
```



The mean is a pretty rough estimator. However it is useful to consider since the quantity ND1 is used to scale the total residual in other cases, to obtain a dimensionless measure of goodness of fit.

Linear Function (Line) Fit

```
> f2:=x->m*x+b;
```

$$f2 := x \rightarrow m\,x + b$$

```
> eqn2:=leastsquare[[x,y],y=f2(x),{m,b}]([Xdata,Ydata]);
```

$$eqn2 := y = \frac{74}{143}\,x + \frac{32}{33}$$

```
> F2:=unapply(rhs(eqn2),x);
```

$$F2 := x \rightarrow \frac{74}{143}\,x + \frac{32}{33}$$

```
> ND2:=N2(Ydata-map(F2,Xdata)); evalf(%);
```

$$ND2 := \frac{1}{429}\sqrt{2645214}$$

$$3.791168734$$

The dimensionless (unadjusted) measure of goodness of fit is given by

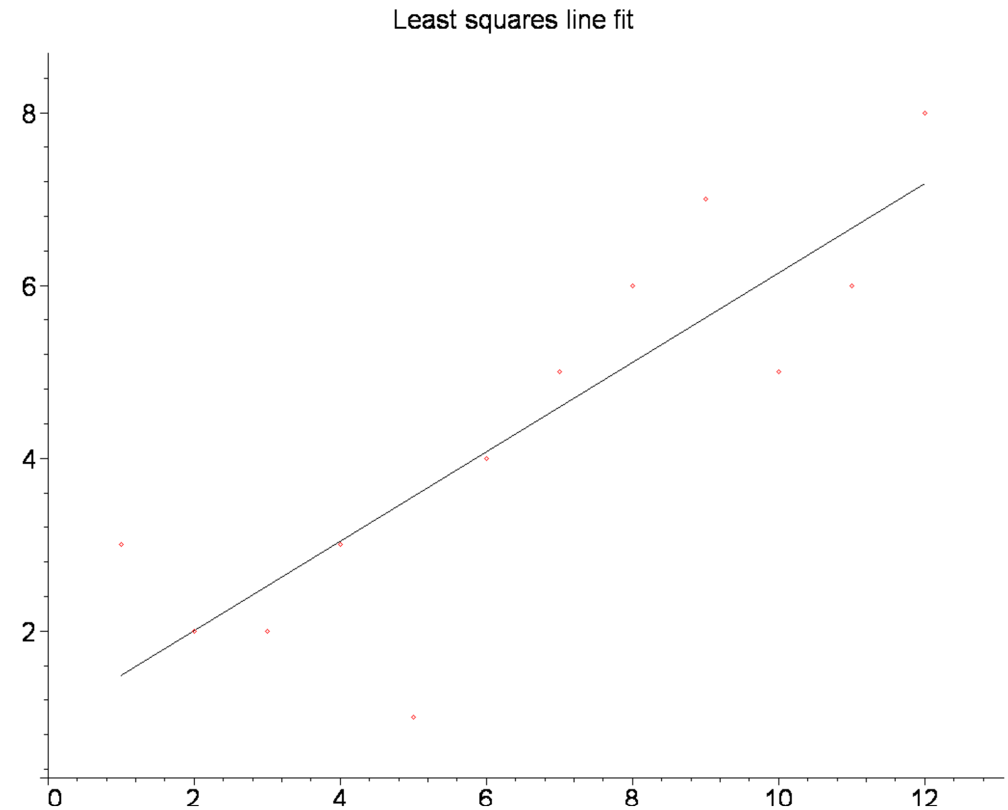
```
> gf2:=1-(ND2/ND1)^2; evalf(%);
```

$$gf2 := \frac{8214}{11297}$$

$$.7270956891$$

Close to 1 is good, close to 0 is bad. Our result here is noncommital.

```
> plt2:=plot(F2,1..12,color=black):
> display([plt0,plt2],title = "Least squares line fit");
```



Special Line Fit

The leastsquare function is quite versatile. For example, if we have reason to expect the slope should be 1/2 in the example above then we can restrict our least squares candidates to lines with slope 1/2.

```
> f3:=x->(1/2)*x+b;
```

$$f3 := x \to \frac{1}{2}x + b$$

```
> eqn3:=leastsquare[[x,y],y=f3(x),{b}]([Xdata,Ydata]);
```

$$eqn3 := y = \frac{1}{2}x + \frac{13}{12}$$

```
> F3:=unapply(rhs(eqn3),x);
```

$$F3 := x \to \frac{1}{2}x + \frac{13}{12}$$

```
> ND3:=N2(Ydata-map(F3,Xdata)); evalf(%);
```

$$ND3 := \frac{1}{6}\sqrt{519}$$

$$3.796928584$$

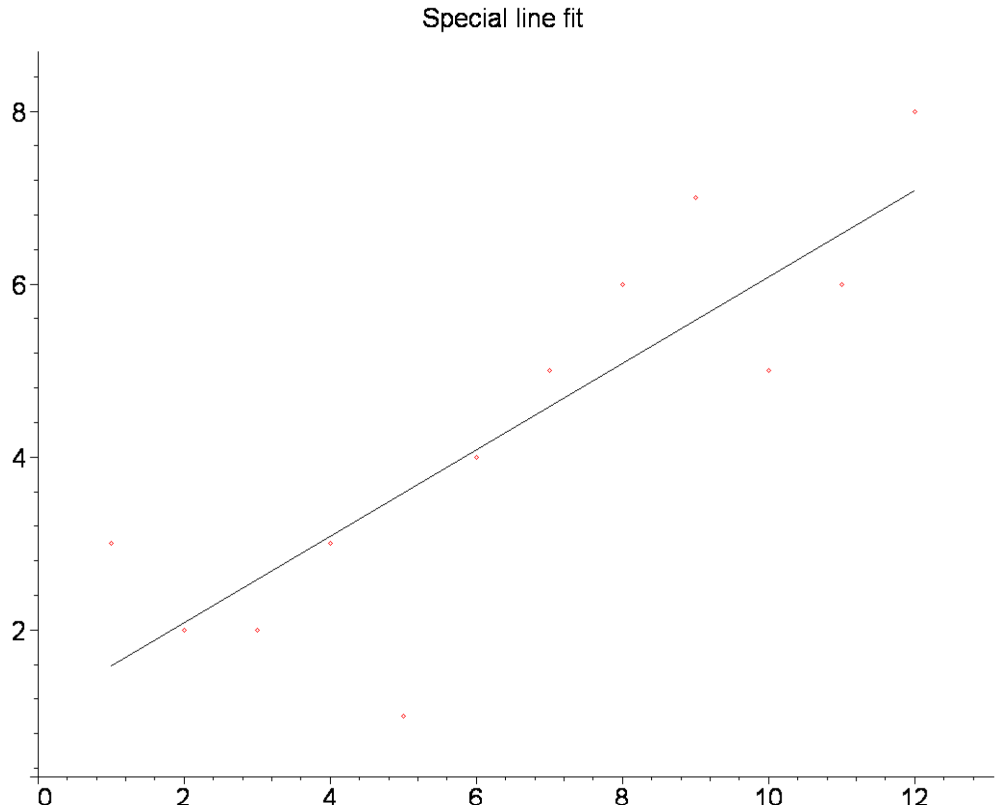
```
> gf3:=1-(ND3/ND1)^2; evalf(%);
```

$$gf3 := \frac{459}{632}$$

$$.7262658228$$

As expected, the fit is not as good as before, since we have restricted the class of candidates.

```
> plt3:=plot(F3,1..12,color=black):
> display([plt3,plt0],title="Special line fit");
```



Quadratic Fit

```
> f4:=x->a*x^2+b*x+c;
```

$$f4 := x \rightarrow a\,x^2 + b\,x + c$$

```
> eqn4:=leastsquare[[x,y],y=f4(x),{a,b,c}]([Xdata,Ydata]);
```

$$eqn4 := y = \frac{31}{1001}x^2 + \frac{115}{1001}x + \frac{21}{11}$$

```
> F4:=unapply(rhs(eqn4),x);
```

$$F4 := x \rightarrow \frac{31}{1001} x^2 + \frac{115}{1001} x + \frac{21}{11}$$

```
> ND4:=N2(Ydata-map(F4,Xdata)); evalf(%);
```

$$ND4 := \frac{1}{1001} \sqrt{13119106}$$

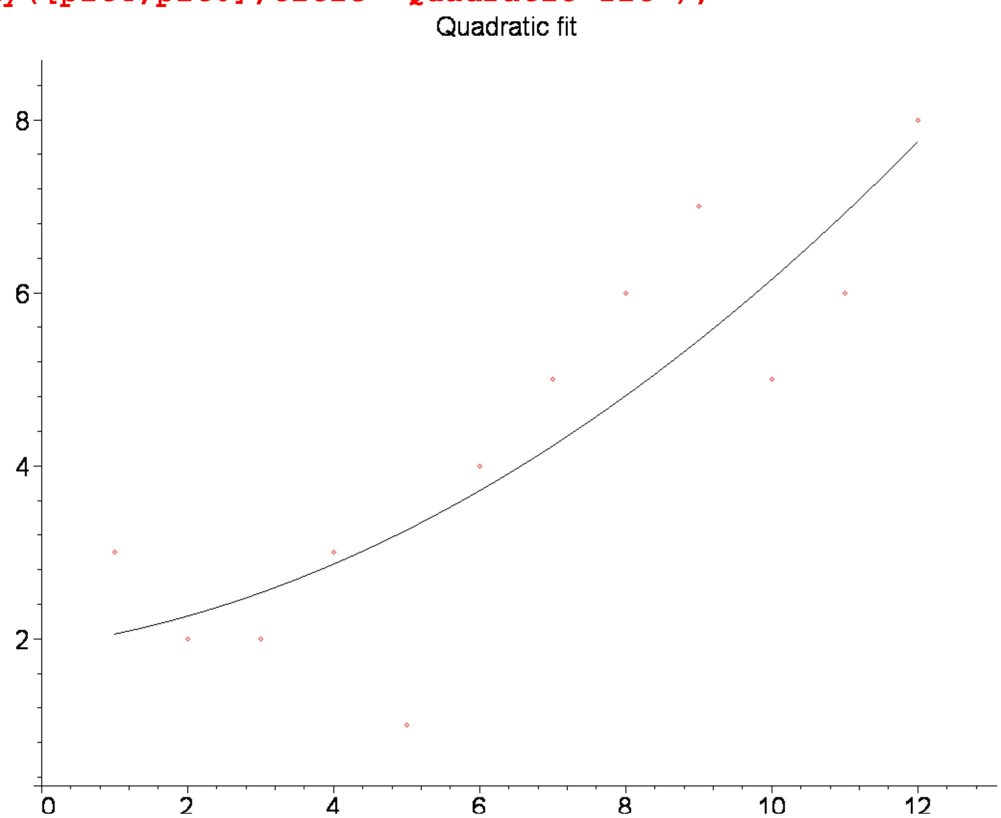
$$3.618412234$$

```
> gf4:=1-(ND4/ND1)^2; evalf(%);
```

$$gf4 := \frac{59420}{79079}$$

$$.7514004982$$

```
> plt4:=plot(F4,1..12,color=black):
> display([plt4,plt0],title="Quadratic fit");
```



Cubic Fit

```
> f5:=x->a*x^3+b*x^2+c*x+d;
```

$$f5 := x \rightarrow a x^3 + b x^2 + c x + d$$

```
> eqn5:=leastsquare[[x,y],y=f5(x),{a,b,c,d}]([Xdata,Ydata]);
```

$$eqn5 := y = -\frac{56}{3861} x^3 + \frac{257}{819} x^2 - \frac{38251}{27027} x + \frac{35}{9}$$

```
> F5:=unapply(rhs(eqn5),x):
> ND5:=N2(Ydata-map(F5,Xdata)); evalf(%);
```

$$ND5 := \frac{1}{3003}\sqrt{96098002}$$

$$3.264388558$$

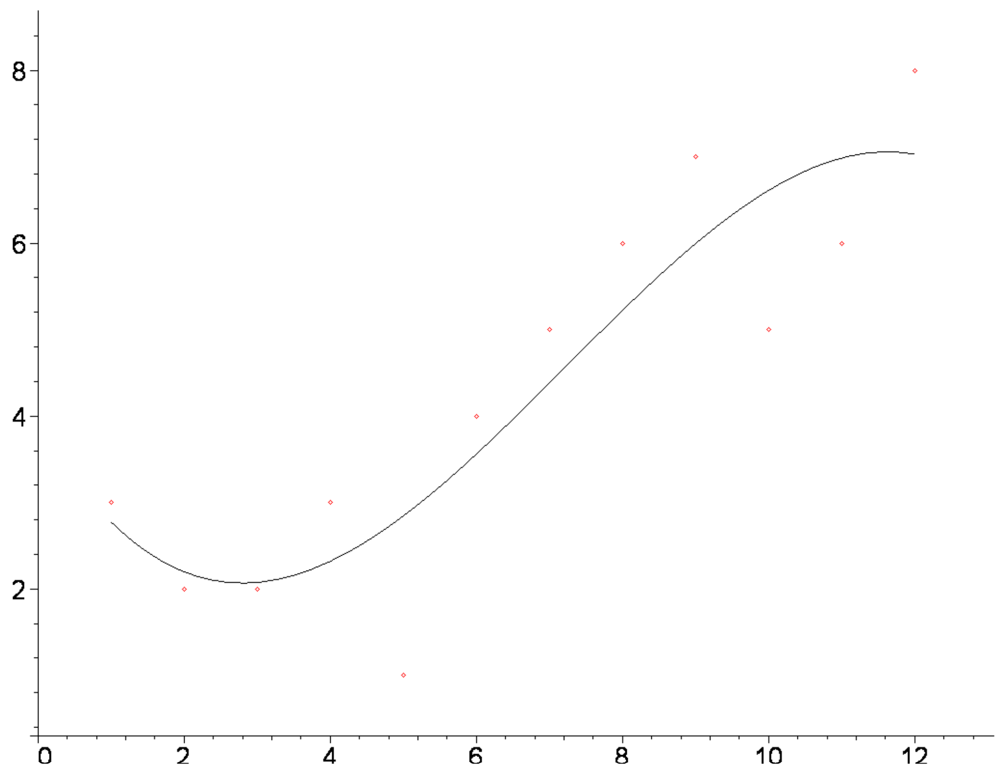
```
> gf5:=1-(ND5/ND1)^2; evalf(%);
```

$$gf5 := \frac{189236}{237237}$$

$$.7976664686$$

```
> plt5:=plot(F5,1..12,color=black):
> display([plt5,plt0],title="Cubic fit");
```

Cubic fit



Degree 4 fit

```
> f6:=x->a*x^4+b*x^3+c*x^2+d*x+e;
```

$$f6 := x \rightarrow a\,x^4 + b\,x^3 + c\,x^2 + d\,x + e$$

```
> eqn6:=leastsquare[[x,y],y=f6(x),{a,b,c,d,e}]([Xdata,Ydata]);
```

$$eqn6 := y = \frac{61}{13728}x^4 - \frac{8033}{61776}x^3 + \frac{53837}{41184}x^2 - \frac{282169}{61776}x + \frac{1319}{198}$$

```
> F6:=unapply(rhs(eqn6),x):
> ND6:=N2(Ydata-map(F6,Xdata)); evalf(%);
```

$$ND6 := \frac{1}{858}\sqrt{6476470}$$

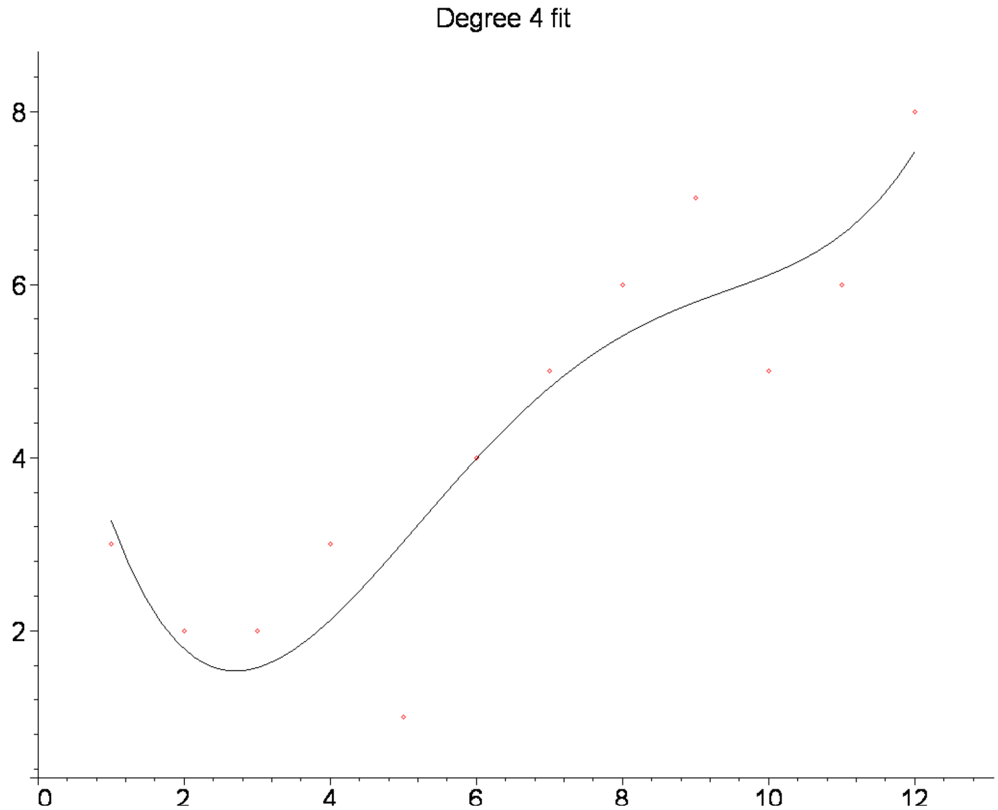
$$2.966073381$$

```
> gf6:=1-(ND6/ND1)^2; evalf(%);
```

$$gf6 := \frac{112919}{135564}$$

$$.8329571273$$

```
> plt6:=plot(F6,1..12,color=black):
> display([plt6,plt0],title="Degree 4 fit");
```



Degree 5 Fit

```
> f7:=x->a*x^5+b*x^4+c*x^3+d*x^2+e*x+f;
```

$$f7 := x \rightarrow a\,x^5 + b\,x^4 + c\,x^3 + d\,x^2 + e\,x + f$$

```
> eqn7:=leastsquare[[x,y],y=f7(x),{a,b,c,d,e,f}]([Xdata,Ydata]);
```

$$eqn7 := y = \frac{23}{10608}x^5 - \frac{321}{4862}x^4 + \frac{82091}{116688}x^3 - \frac{29543}{9724}x^2 + \frac{47885}{9724}x + \frac{3}{11}$$

```
> F7:=unapply(rhs(eqn7),x):
> ND7:=N2(Ydata-map(F7,Xdata)); evalf(%);
```

$$ND7 := \frac{1}{2431}\sqrt{32344455}$$

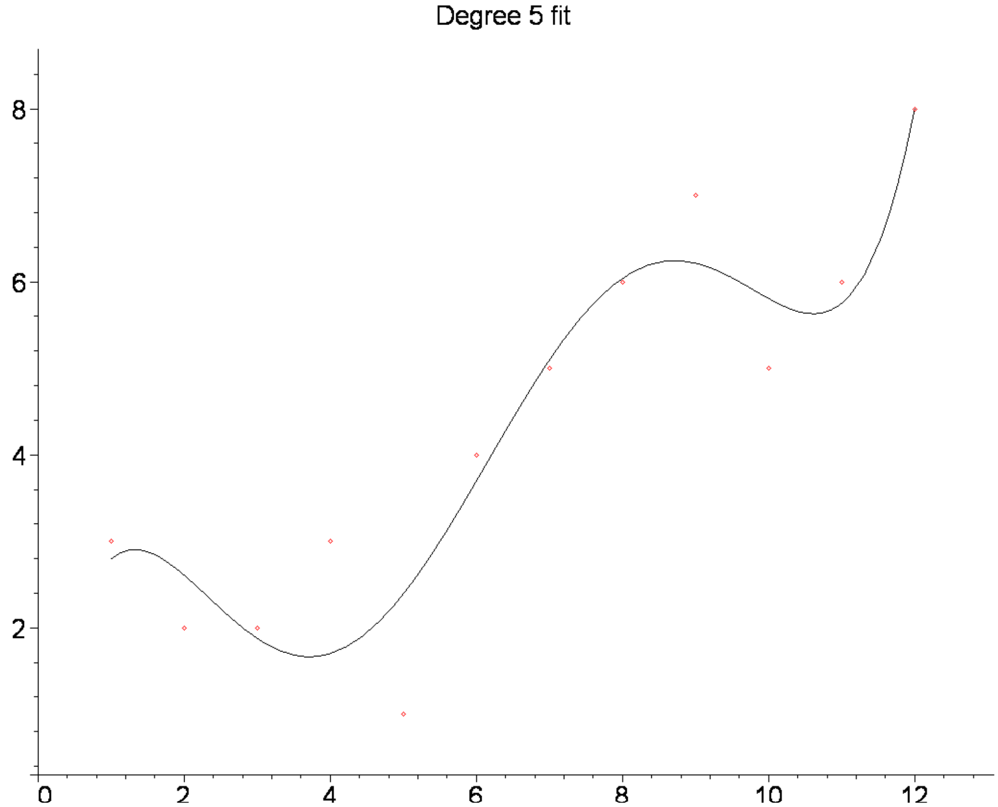
$$2.339456423$$

```
> gf7:=1-(ND7/ND1)^2; evalf(%);
```

$$gf7 := \frac{344183}{384098}$$

$$.8960812084$$

```
> plt7:=plot(F7,1..12,color=black):
> display([plt7,plt0],title="Degree 5 fit");
```



Degree 10 Fit

```
> f12:=x->a10*x^10+a9*x^9+a8*x^8+a7*x^7+a6*x^6+a5*x^5+a4*x^4+a3*x^3+
  a2*x^2+a1*x+a0;
```

$$f12 := x \rightarrow a10\, x^{10} + a9\, x^9 + a8\, x^8 + a7\, x^7 + a6\, x^6 + a5\, x^5 + a4\, x^4 + a3\, x^3 + a2\, x^2 + a1\, x + a0$$

```
> eqn12:=leastsquare[[x,y],y=f12(x),{a10,a9,a8,a7,a6,a5,a4,a3,a2,a1,
  a0}]([Xdata,Ydata]);
```

$$eqn12 := y = -\frac{157}{3628800} x^{10} + \frac{14263}{5080320} x^9 - \frac{133939}{1693440} x^8 + \frac{2915323}{2298240} x^7 - \frac{293270623}{22982400} x^6 + \frac{6533315417}{78140160} x^5 - \frac{84072994307}{234420480} x^4 + \frac{50526430079}{51279480} x^3 - \frac{124616339339}{75969600} x^2 + \frac{278166527}{188955} x - \frac{1055}{2}$$

```
> F12:=unapply(rhs(eqn12),x):
> ND12:=N2(Ydata-map(F12,Xdata)); evalf(%);
```

$$ND12 := \frac{109}{58786}\sqrt{176358}$$

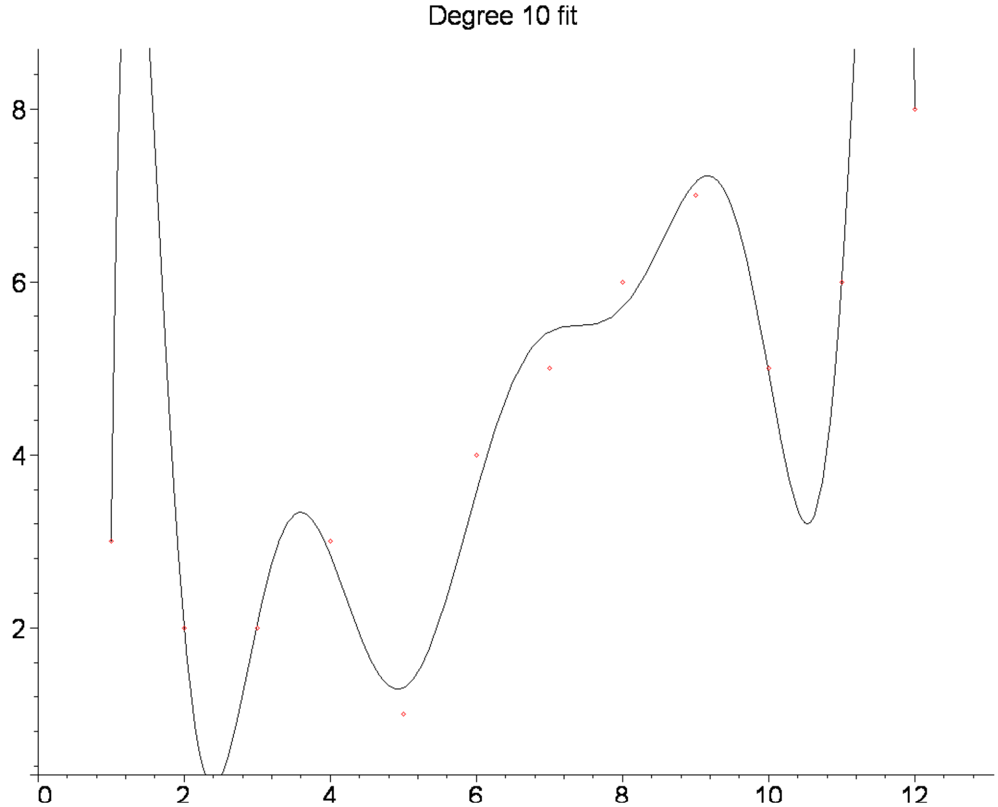
$$.7786641323$$

```
> gf12:=1-(ND12/ND1)^2; evalf(%);
```

$$gf12 := \frac{9181259}{9288188}$$

$$.9884876361$$

```
> plt12:=plot(F12,1..12,color=black):
> display([plt12,plt0],title="Degree 10 fit");
```




```
>
```

This example is typical. Even though our goodness of fit is about 0.988 the graph intuitively looks worse. The fit is of course great near the data points, but polynomials of high degree tend to oscillate a lot and therefore we get large (unjustified?) swings between the data points. The philosophy, more or less, is a large goodness of fit is meaningless unless it is achieved by a polynomial of low degree, or more generally, by a function with few adjustable parameters. Is it meaningful in the later case? Well, maybe, maybe not.

An Exponential Fit

Maple can handle any kind of linear least squares fit (but not nonlinear)

```
> f20:=x->a+b*exp(x/15);
```

$$f20 := x \to a + b\,\mathbf{e}^{(1/15\,x)}$$

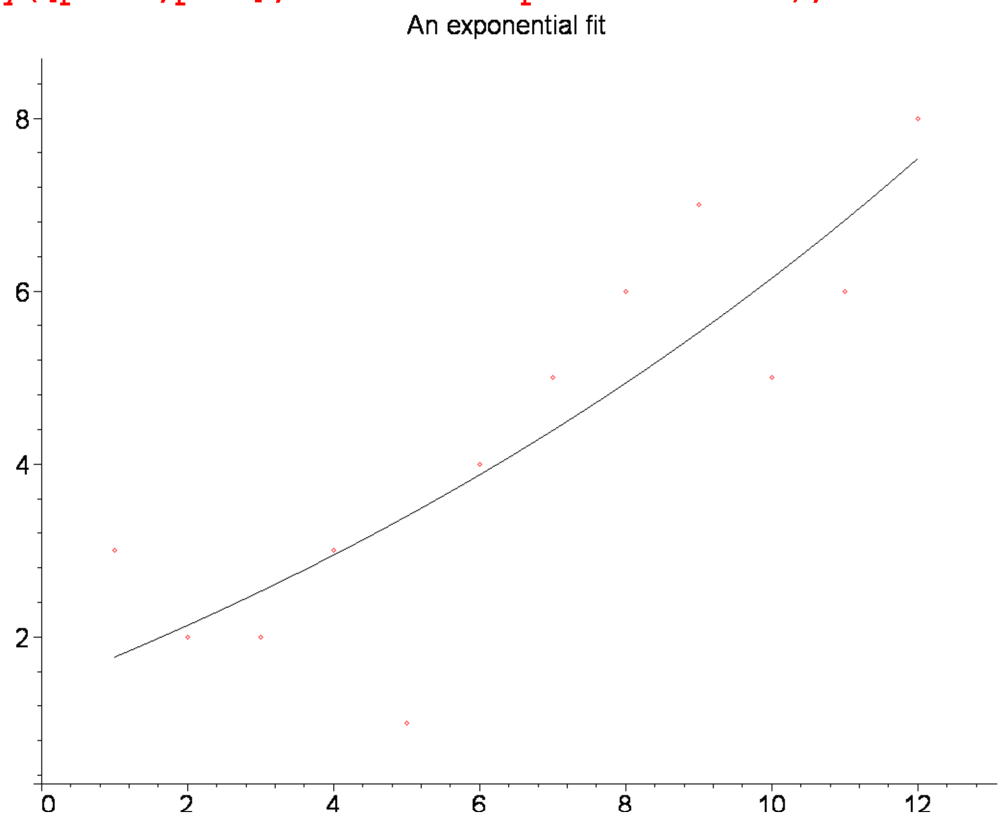
```
> eqn20:=leastsquare[[x,y],y=f20(x),{a,b}]([Xdata,Ydata]):
> F20:=unapply(rhs(eqn20),x):
> ND20:=N2(Ydata-map(F20,Xdata)): evalf(%);
```

$$3.670066155$$

```
> gf20:=1-(ND20/ND1)^2: evalf(%);
```

$$.7442521727$$

```
> plt20:=plot(F20,1..12,color=black):
> display([plt20,plt0],title="An exponential fit");
```



Would you care to use any of these models to predict the "future" (larger values of x)? What is required to make prediciton anything but a meaningless exercise?

```
>
```