>

# Feb 15, 2024

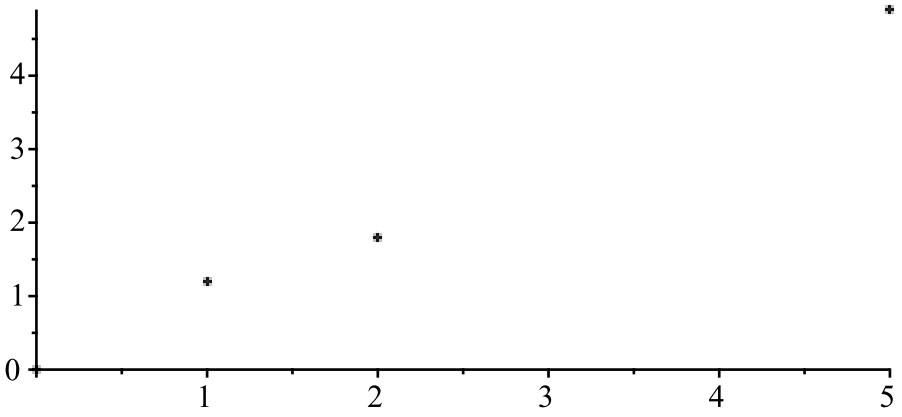
Fitting curves to data continues.

> $points := [[0, 0], [1, 1.2], [2, 1.8], [5, 4.9]];$

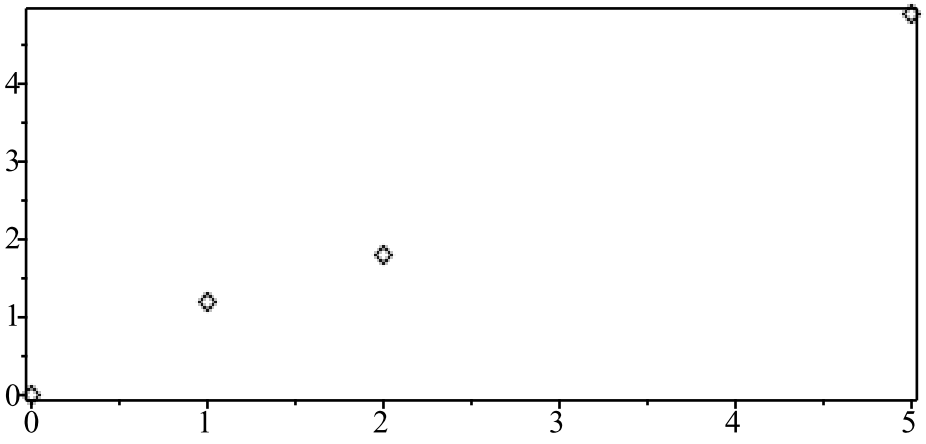
$$points := [[0, 0], [1, 1.2], [2, 1.8], [5, 4.9]] \quad (1)$$

> $with(plots):$

> pointplot(points);



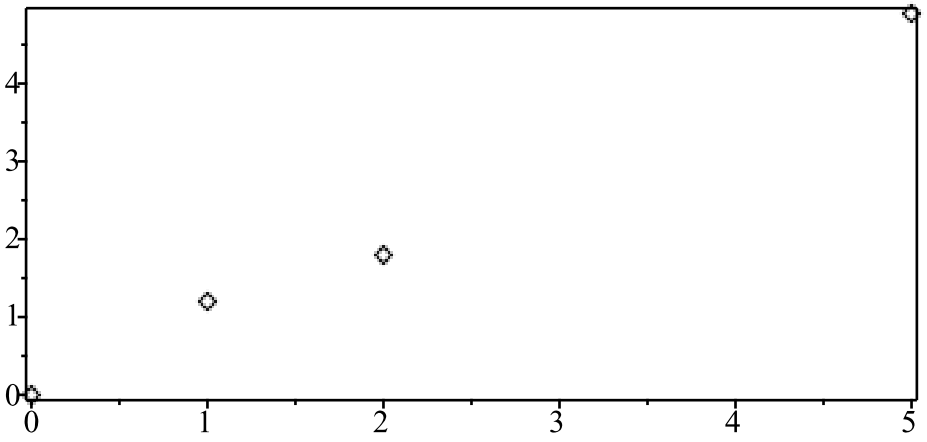

> pointplot(points,symbolsize=19,size=[.7,.5],axes=box);




We can save our favorite options, so we don't have to keep retyping them:

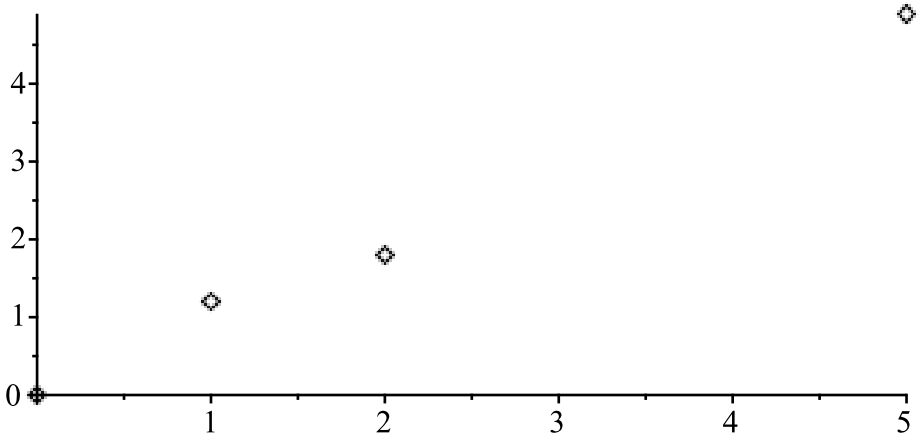
> plots[setoptions](symbolsize=19,size=[.7,.5],axes=box)

> pointplot(points)




But we can override these options if we want.

```
> pointplot(points,axes=normal)
```

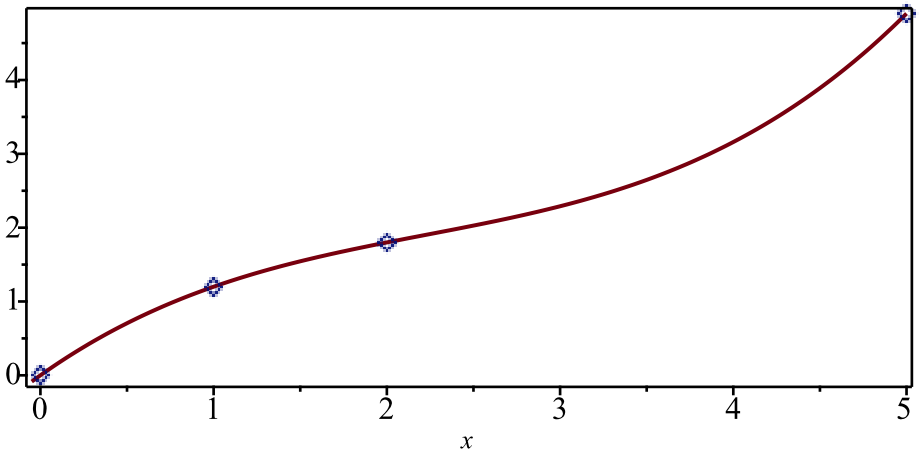



Back to fitting curves

```
> with(CurveFitting):
  p:=PolynomialInterpolation(points,x);
```

$$p := 0.08166666667\,x^3 - 0.5450000000\,x^2 + 1.663333333\,x \quad \textbf{(2)}$$

```
> plot([p,points],x=-.05..5,style=[line,point])
```

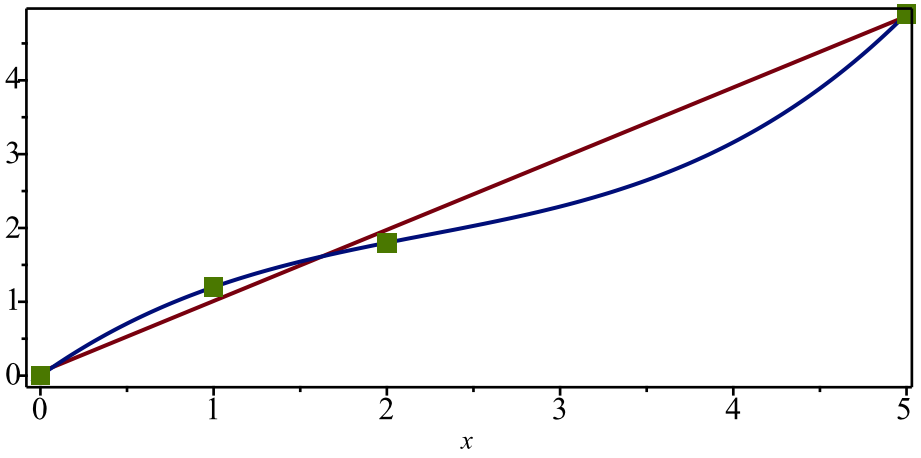



Least-squares fitting minimizes the sum of the squares of the difference between the line (or function) and the y value at each of the points.

```
> lsq:=LeastSquares(points,x)
```

$$lsq := 0.0464285714285716 + 0.964285714285714\,x \quad (3)$$

```
> plot([lsq,p,points],x=-.05..5,style=[line$2,point],symbol=
  solidbox)
```




Let's do this by hand, to minimize the sum of the squares of the error between our data points and our "trial" line.
Assume line is y=m*x+b,  error at $p$ is

```
> err:=p->m*p[1]+b -p[2]
```

$$err := p \mapsto m \cdot p_1 + b - p_2 \quad (4)$$

```
> E:=pts->add(err(pts[i]),i=1..nops(pts))
```

Warning, (in E) `i` is implicitly declared local

$$E := pts \mapsto add\left(err\left(pts_i\right), i = 1\,..nops(pts)\right) \quad (5)$$

```
> E(points);
```

$$4\,b + 8\,m - 7.9 \quad (6)$$

Not what I want, since given any m, there is a b that makes E=0.

```
> err:=p->(m*p[1]+b -p[2])^2
```

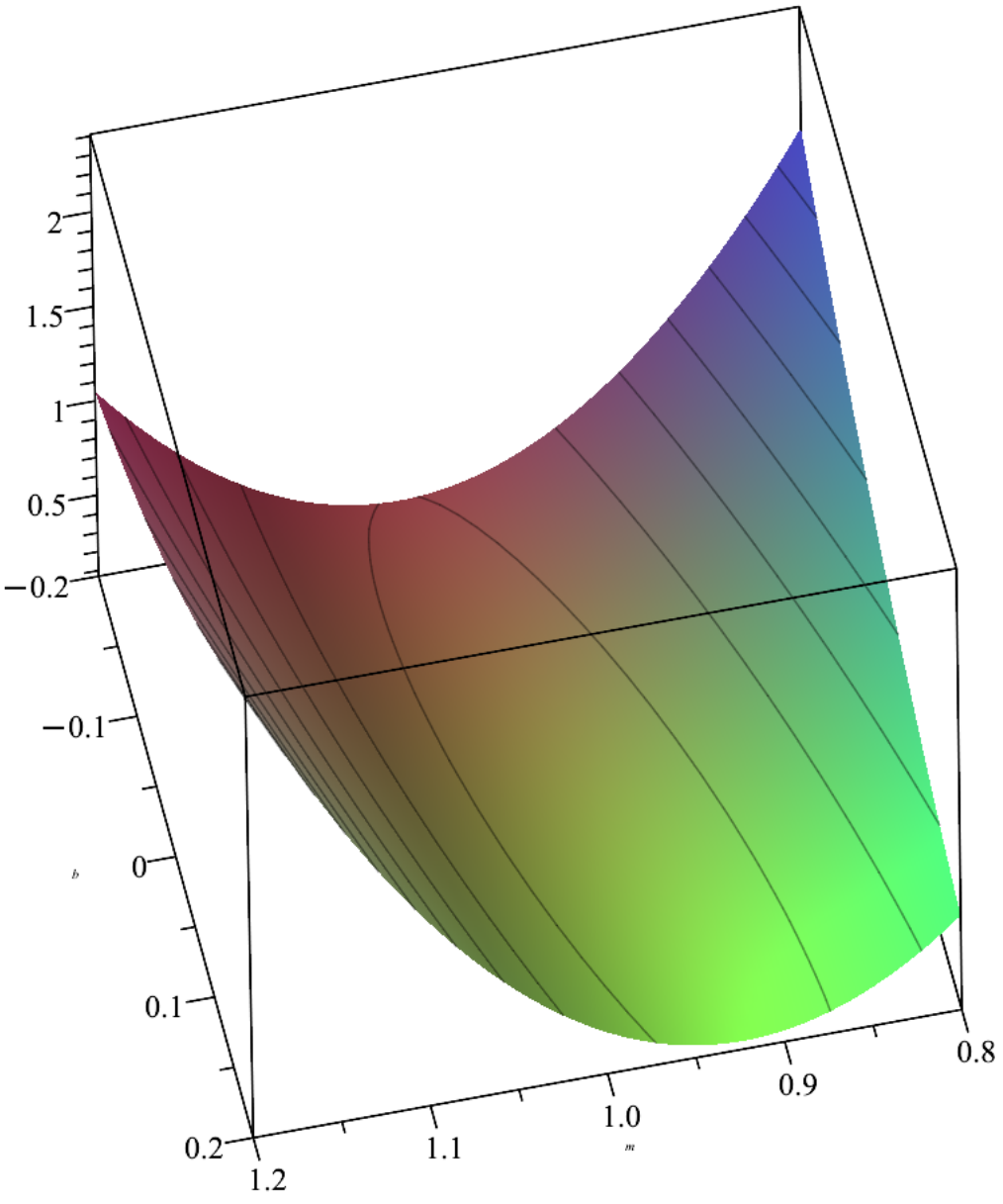
$$err := p \mapsto \left(m \cdot p_1 + b - p_2\right)^2 \quad (7)$$

```
> E(points);
```

$$b^2 + (b + m - 1.2)^2 + (b + 2\,m - 1.8)^2 + (b + 5\,m - 4.9)^2 \quad (8)$$

Note that E is quadratic in b and m, with a unique minimum:

```
> plot3d(E(points),m=0.8..1.2,b=-0.2..0.2)
```




Next, take partials with respect to m and b, set to zero and solve.

```
> dm:=diff(E(points),m);
```

$$dm := 16\,b + 60\,m - 58.6 \quad (9)$$

```
> db:=diff(E(points),b);
```

$$db := 8\,b + 16\,m - 15.8 \quad (10)$$

```
> solve({dm=0,db=0});
```

$$\{b = 0.04642857143,\ m = 0.9642857143\} \quad (11)$$

```
> lsq:=subs(%,m*x+b);
```

$$lsq := 0.9642857143\,x + 0.04642857143 \quad (12)$$

Of course, this is exactly the answer maple got (well ok, up to digits)

```
> maplesanswer:=LeastSquares(points,x)
```

$$maplesanswer := 0.0464285714285716 + 0.964285714285714\,x \tag{13}$$

Let's try another example with nonlinear data.

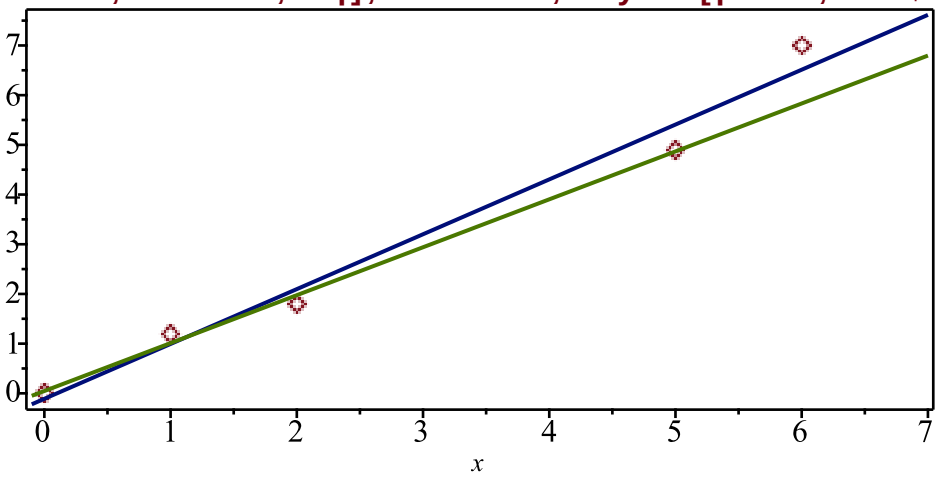
```
> newdata:= [ op(points), [6,7]];
```

$$newdata := [[0, 0], [1, 1.2], [2, 1.8], [5, 4.9], [6, 7]] \tag{14}$$

```
> newline:=LeastSquares(newdata,x);
```

$$newline := -0.110447761194030 + 1.10373134328358\,x \tag{15}$$

```
> plot([newdata,newline,lsq],x=-0.1..7, style=[point,line$2] )
```




maybe this isn't well approximated by a line. Let's do the same stuff with y=a*x^2 + b*x +c

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

$$f := x \mapsto a \cdot x^2 + b \cdot x + c \tag{16}$$

```
> newdata[2];
```

$$[1, 1.2] \tag{17}$$

```
> f(1);
```

$$a + b + c \tag{18}$$

```
> err:=p->(f(p[1])-p[2])^2
```

$$err := p \mapsto \left(f\left(p_1\right) - p_2\right)^2 \tag{19}$$

```
> err(newdata[2])
```

$$(a + b + c - 1.2)^2 \tag{20}$$

```
> E(newdata)
```

$$c^2 + (a + b + c - 1.2)^2 + (4\,a + 2\,b + c - 1.8)^2 + (25\,a + 5\,b + c - 4.9)^2 + (36\,a + 6\,b + c - 7)^2 \tag{21}$$

```
> da:=diff(E(newdata),a);
  db:=diff(E(newdata),b);
  dc:=diff(E(newdata),c);
```

$$da := 3876\,a + 700\,b + 132\,c - 765.8$$

$$db := 700\,a + 132\,b + 28\,c - 142.6$$

$$dc := 10\,c + 132\,a + 28\,b - 29.8 \tag{22}$$

```
> solve({da=0,db=0,dc=0},{a,b,c})
```

$$\{a = 0.08011083744,\ b = 0.6099137931,\ c = 0.2147783251\} \quad \textbf{(23)}$$

```
> parab:=subs(%,f(x));
```

$$parab := 0.08011083744\,x^2 + 0.6099137931\,x + 0.2147783251 \quad \textbf{(24)}$$

```
> LeastSquares(newdata,x,curve=a*x^2+b*x+c)
```

$$0.214778325123154 + 0.609913793103447\,x + 0.0801108374384237\,x^2 \quad \textbf{(25)}$$

BUT... the function we fit has to be linear in the coefficients (so that we have a system of linear equations with a unique solution).

```
> LeastSquares(newdata,x,curve=a*sin(b*x))
Error, (in CurveFitting:-LeastSquares) curve to fit is not linear in the parameters
```