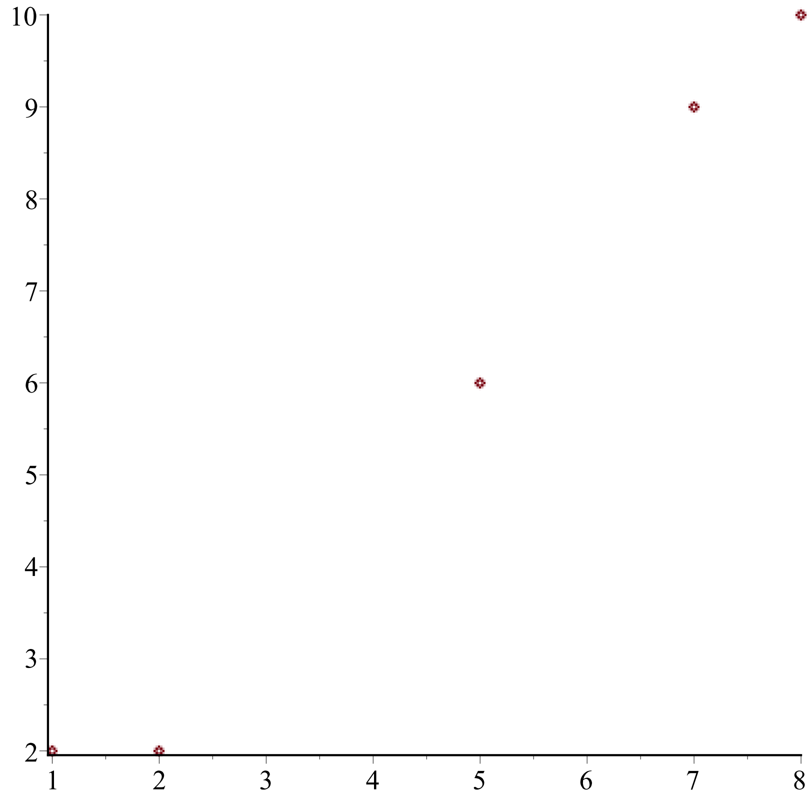
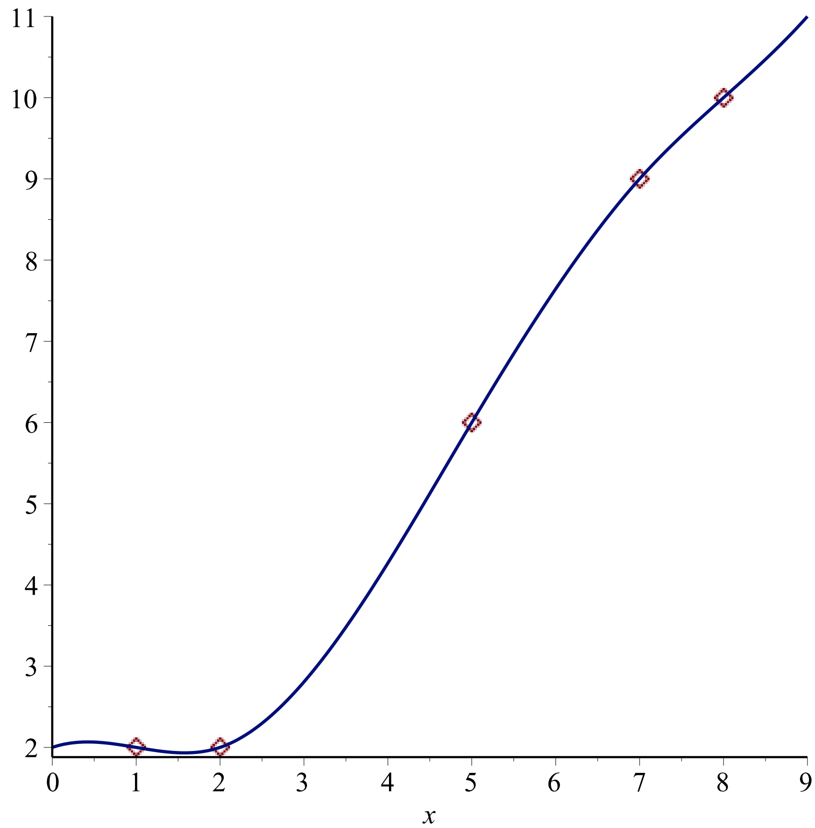
```
> data := [[1, 2], [2, 2], [5, 6], [7, 9], [8, 10]] :
> plot(data, style=point);
```



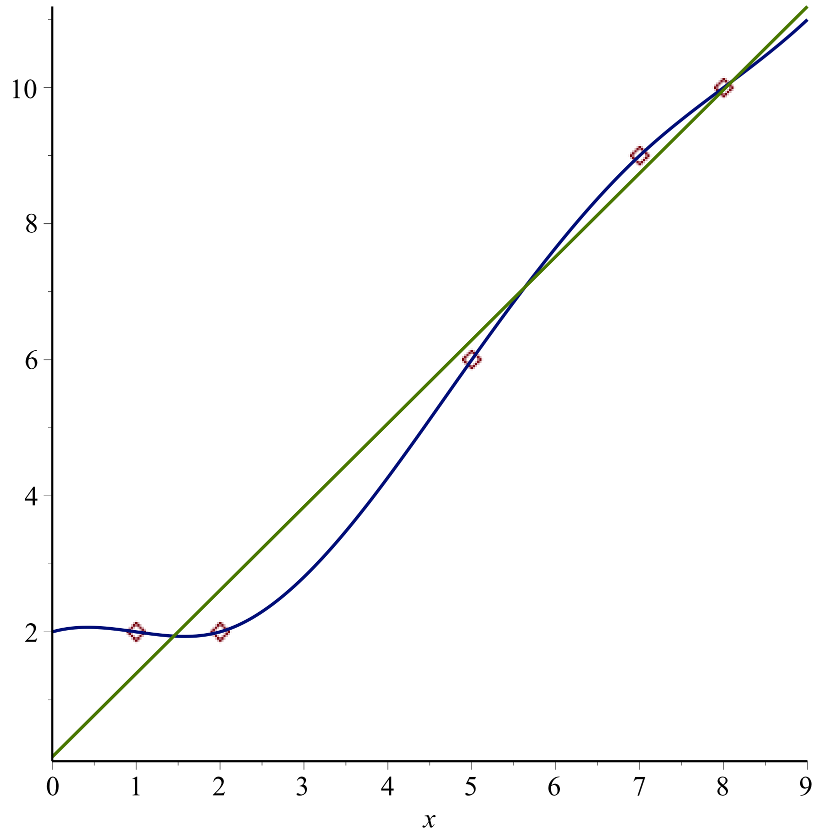

```
> with(CurveFitting) :
> plot([data, Spline(data, x, degree = 3)], x = 0 ..9, style = [point, line], symbolsize = 18);
```




> *LeastSquares*(*data*, *x*);

$$\frac{5}{31} + \frac{38}{31}\,x \qquad (1)$$

> *plot*([*data*, *Spline*(*data*, *x*, *degree* = 3), *LeastSquares*(*data*, *x*)], *x* = 0 ..9, *style* = [*point*, *line*, *line*], *symbolsize* = 18);




>

Least squares minimizes the sum (or average) of the squares of the difference in y values between each data point and the line.

> *data*;

$$[[1, 2], [2, 2], [5, 6], [7, 9], [8, 10]] \quad (2)$$

The distance from the point [5,6] to the line y=x+2 is (5*1+2) - 6 =1

> $ptdist := (m, b, pt) \rightarrow (m \cdot pt[1] + b - pt[2])^2$

$$ptdist := (m, b, pt) \rightarrow \left(m\, pt_1 + b - pt_2\right)^2 \quad (3)$$

> $ptdist(1, 2, [5, 6]);$

$$1 \quad (4)$$

> $nops(data);$

$$5 \quad (5)$$

> $linedist := (m, b, data) \rightarrow sum(ptdist(m, b, data), i = 1 .. nops(data));$

$$linedist := (m, b, data) \rightarrow \sum_{i=1}^{nops(data)} ptdist(m, b, data) \quad (6)$$

> $linedist(2, 1, data);$

$$5\,([0,2]+1)^2 \quad \textbf{(7)}$$

> $linedist := (m, b, data) \rightarrow sum(ptdist(m, b, data[i]), i = 1..nops(data));$

$$linedist := (m, b, data) \rightarrow \sum_{i=1}^{nops(data)} ptdist(m, b, data_i) \quad \textbf{(8)}$$

> $linedist(2, 1, data);$

$$120 \quad \textbf{(9)}$$

> $linedist := (m, b, data) \rightarrow \frac{sum(ptdist(m, b, data[i]), i = 1..nops(data))}{nops(data)};$

$$linedist := (m, b, data) \rightarrow \frac{\sum_{i=1}^{nops(data)} ptdist(m, b, data_i)}{nops(data)} \quad \textbf{(10)}$$

> $linedist(2, 1, data);$

$$24 \quad \textbf{(11)}$$

> $linedist(1.5, 1, data);$

$$5.150000000 \quad \textbf{(12)}$$

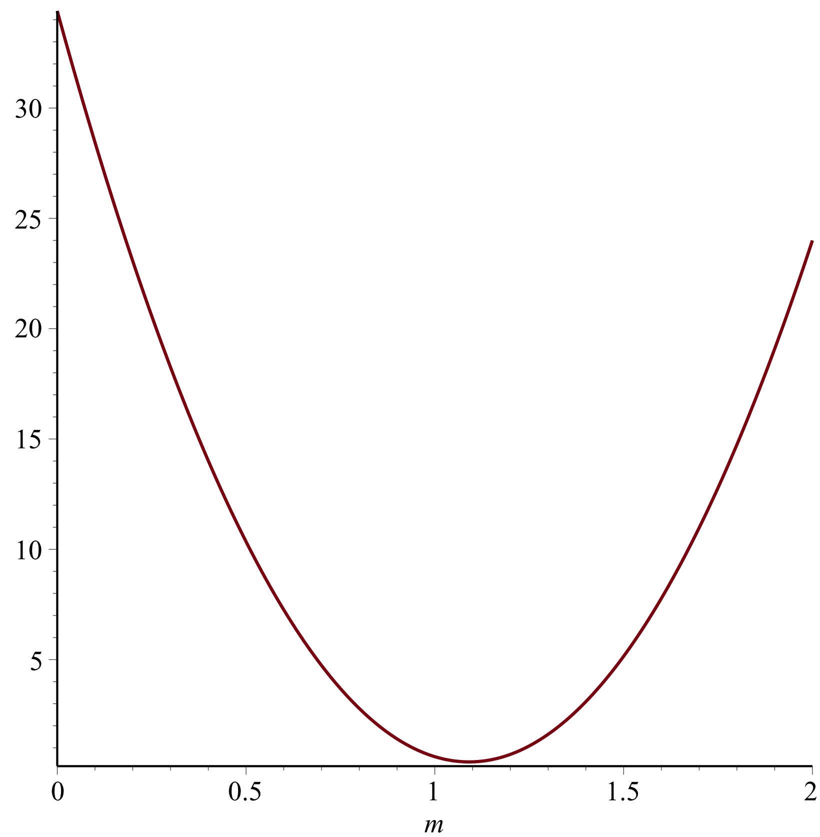
> $linedist(1.0, 1, data);$

$$0.6000000000 \quad \textbf{(13)}$$

> $linedist(m, 1, data);$

$$\frac{1}{5}(m-1)^2 + \frac{1}{5}(2m-1)^2 + \frac{1}{5}(5m-5)^2 + \frac{1}{5}(7m-8)^2 + \frac{1}{5}(8m-9)^2 \quad \textbf{(14)}$$

> $plot(linedist(m, 1, data), m = 0..2);$




> $diff(linedist(m, 1, data), m);$

$$\frac{286}{5}\,m - \frac{312}{5} \tag{15}$$

> $solve(\% = 0);$

$$\frac{12}{11} \tag{16}$$

> $diff\left(linedist\left(\frac{12}{11}, b, data\right), b\right);$

$$-\frac{86}{55} + 2\,b \tag{17}$$

> $solve(\% = 0);$

$$\frac{43}{55} \tag{18}$$

> $\{diff(linedist(m, b, data), m) = 0, diff(linedist(m, b, data), b) = 0\}$

$$\left\{\frac{46}{5}\,m + 2\,b - \frac{58}{5} = 0, \frac{286}{5}\,m + \frac{46}{5}\,b - \frac{358}{5} = 0\right\} \tag{19}$$

> $solve(\%);$

(20)

$$\left\{b = \frac{5}{31}, m = \frac{38}{31}\right\} \quad (20)$$

> *evalf*(%);

$$\{b = 0.1612903226, m = 1.225806452\} \quad (21)$$

> *LeastSquares*(*data*, *x*);

$$\frac{5}{31} + \frac{38}{31} x \quad (22)$$

>

Process: Write a distance between a point and my target function, depending on parameters. Minimize sum of squares by taking the partials wrt each of the variable and solve for them = 0.

> *data2* := [[1, 2], [2, 2], [5, 6], [7, 9], [8, 10], [9, 10.2], [11, 11]] :

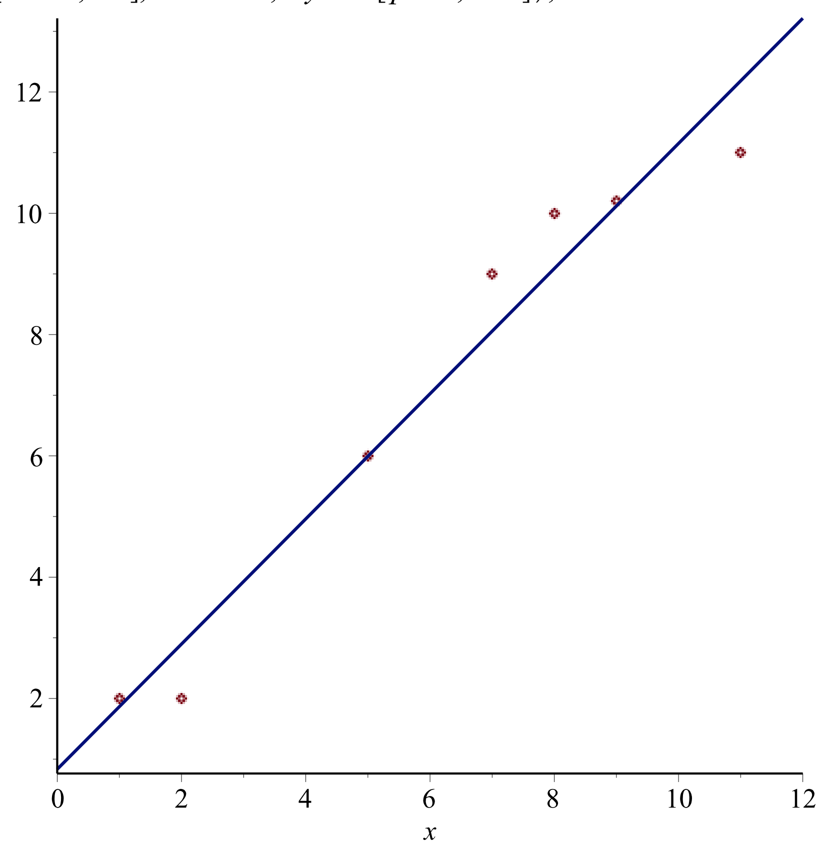
> *sol* := *solve*({*diff*(*linedist*(*m*, *b*, *data2*), *m*) = 0, *diff*(*linedist*(*m*, *b*, *data2*), *b*) = 0});

$$sol := \{b = 0.8332155530, m = 1.031802119\} \quad (23)$$

> *lin* := *subs*(*sol*, *m*·*x* + *b*);

$$lin := 1.031802119\, x + 0.8332155530 \quad (24)$$

> *plot*([*data2*, *lin*], *x* = 0 ..12, *style* = [*point*, *line*]);




target function is a*x^3 + b*x^2 + c*x + d;

```
> targ := (a, b, c, d, x) -> a*x^3 + b*x^2 + c*x + d;
```

$$targ := (a, b, c, d, x) \rightarrow a\,x^3 + b\,x^2 + c\,x + d \qquad (25)$$

```
> ptdist := (a, b, c, d, pt) -> (targ(a, b, c, d, pt[1]) - pt[2])^2
```

$$ptdist := (a, b, c, d, pt) \rightarrow \left(targ\left(a, b, c, d, pt_1\right) - pt_2\right)^2 \qquad (26)$$

```
> ptdist(1, 2, 3, 4, [5, 6]);
```

$$35344 \qquad (27)$$

```
> cubdist := (a, b, c, d, data) -> sum(ptdist(a, b, c, d, data[i]), i = 1..nops(data)) / nops(data);
```

$$cubdist := (a, b, c, d, data) \rightarrow \frac{\sum_{i=1}^{nops(data)} ptdist\left(a, b, c, d, data_i\right)}{nops(data)} \qquad (28)$$

```
> cubdist(1, 0, 2, -1, data2);
```

$$3.919779486\ 10^5 \qquad (29)$$

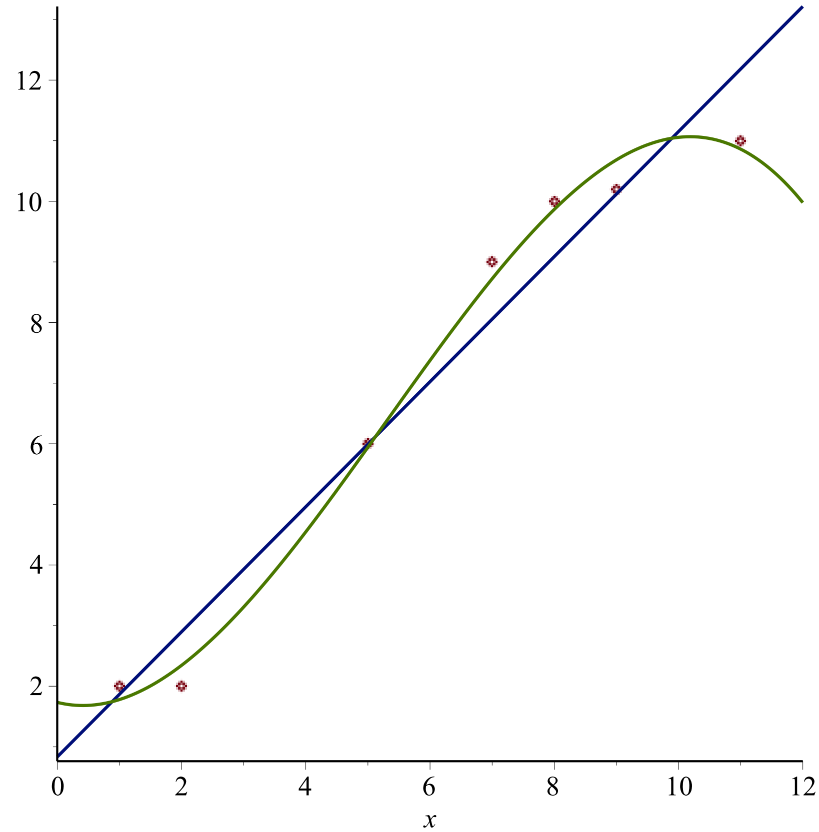
```
> solve(
    { diff(cubdist(a, b, c, d, data2), a) = 0,
      diff(cubdist(a, b, c, d, data2), b) = 0,
      diff(cubdist(a, b, c, d, data2), c) = 0,
      diff(cubdist(a, b, c, d, data2), d) = 0 });
```

$$\{a = -0.02009567630, b = 0.3195404736, c = -0.2535906383, d = 1.733521584\} \qquad (30)$$

```
> f3 := subs(%, targ(a, b, c, d, x));
```

$$f3 := -0.02009567630\,x^3 + 0.3195404736\,x^2 - 0.2535906383\,x + 1.733521584 \qquad (31)$$

```
> plot([data2, lin, f3], x = 0..12, style = [point, line, line]);
```




> $LeastSquares(data2, x, curve = targ(a, b, c, d, x));$

$$1.73352099111052 - 0.253590116002082\, x + 0.319540375116072\, x^2 - 0.0200956711593076\, x^3 \quad (32)$$

> $targ(a, b, c, d, x);$

$$a\, x^3 + b\, x^2 + c\, x + d \quad (33)$$

> $LeastSquares(data2, x, curve = \mathrm{A} \cdot \exp(\mathrm{B} \cdot \mathrm{x}) + c)$

Error, (in CurveFitting:-LeastSquares) curve to fit is not linear in the parameters

> $LeastSquares(data2, x, curve = \mathrm{A} \cdot \exp(\mathrm{x}) + c + B \cdot x)$

$$0.2681969853 + 1.191499623\, x - 0.00004032305909\, e^x \quad (34)$$

> $LeastSquares(data2, x, curve = \mathrm{A} \cdot \exp(\mathrm{B} \cdot \mathrm{x}))$

Error, (in CurveFitting:-LeastSquares) curve to fit is not linear in the parameters

> $\ln(A \cdot \exp(B \cdot x)) = \ln(A) + B \cdot x$

$$\ln\left(A\, e^{B\, x}\right) = \ln(A) + B\, x \quad (35)$$

want to take log of y values in data.

> $data2;$

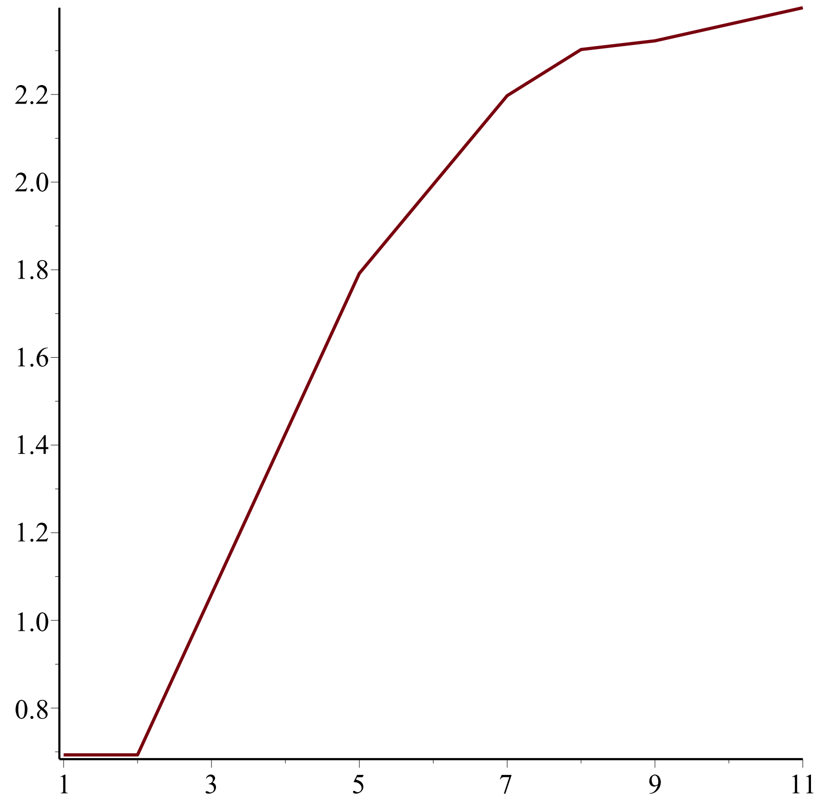
(36)

$[[1, 2], [2, 2], [5, 6], [7, 9], [8, 10], [9, 10.2], [11, 11]]$ **(36)**

> *data3* := evalf([[data2[i][1], ln(data2[i][2])]$ i = 1 ..nops(data2)]);

$data3 := [[1., 0.6931471806], [2., 0.6931471806], [5., 1.791759469], [7., 2.197224578], [8., 2.302585093], [9., 2.322387720], [11., 2.397895273]]$ **(37)**

> *plot*(*data3*);




> *loggy* := *unapply*(*LeastSquares*(data3, x), *x*); *loggy*(*x*);

$loggy := x \rightarrow 0.558823121250176 + 0.19735778245229688\, x$

$0.558823121250176 + 0.197357782452297\, x$ **(38)**

ln(A) + B*x, want Aexp(Bx)

> *subs*({*A* = *loggy*(0), *B* = *loggy*(1) − *loggy*(0)}, *A*·exp(*B*·*x*));

$0.558823121250176\, e^{0.197357782452297\, x}$ **(39)**

>