PolyFit

$$PolyFit(x, y, n) := \left| \begin{array}{l} [\, m := \text{length}(x) \;\; r := [1..m] \;\; c := [1..(n+1)] \;\; A := \text{matrix}(m, n+1) + 1 \,] \\ A_{r\,c} := \left( x_r + \left( (x_r = 0) \wedge (c = 1) \right) \right)^{c-1} \\ \text{eval}\left( \left( A^{\mathrm{T}} \cdot A \right)^{-1} \cdot \left( A^{\mathrm{T}} \cdot y \right) \right) \end{array} \right.$$

You can try to substitute the last line with someone of the following, from dislin plugin, which are more robust and general

`dn_LinAlgLLS_COF(A, y)`

`dn_LinAlgLLS_SVD(A, y)`

`dn_LinAlgLLS_QRorLQ(A, y)`

In matlab: (A\y).'

A is the Vandermonde matrix. The code $(x_r = 0) \wedge (c = 1)$ is for avoid the SMath error 0^0, Uncertainty.

$$PolyVal(p, x) := \left| \text{eval}\left( \sum_{k=1}^{\text{length}(p)} p_k \cdot \overrightarrow{x^{k-1}} \right) \right.$$

Must to use Horner, but this is more elegant.

$$R_{Pearson}(X, Y) := \left| \begin{array}{l} [\, \Xi \;\; \Psi \,] := [\, X \;\; Y \,] - \frac{1}{\text{length}(X)} \cdot \left[ \sum X \;\; \sum Y \right] \\ \frac{\sum \overrightarrow{\Xi \cdot \Psi}}{\sqrt{\left( \sum \overrightarrow{\Xi^2} \right) \cdot \left( \sum \overrightarrow{\Psi^2} \right)}} \end{array} \right.$$

---

Shorthands:

$$plot(M) := \left| \begin{array}{l} C := PolyFit(X, Y, n) \\ \left\{ \begin{array}{l} PolyVal(C, x) \\ M \\ \text{augment}(X, Y, \text{"o"}) \\ \text{augment}(X, PolyVal(C, X), \text{"x"}) \end{array} \right. \end{array} \right.$$

$$R2(X, Y, n) := \left| R_{Pearson}(Y, PolyVal(PolyFit(X, Y, n), X))^2 \right.$$

$$MCols(M) := \left| \begin{array}{l} [\, n := \text{cols}(M) \;\; R := \text{matrix}(1, n) \;\; c := [1..n] \,] \\ \text{eval}\left( R_c := \text{col}(M, c) \right) \end{array} \right.$$

---

PolyFit

## Lineal Polyfit

$n := 1$

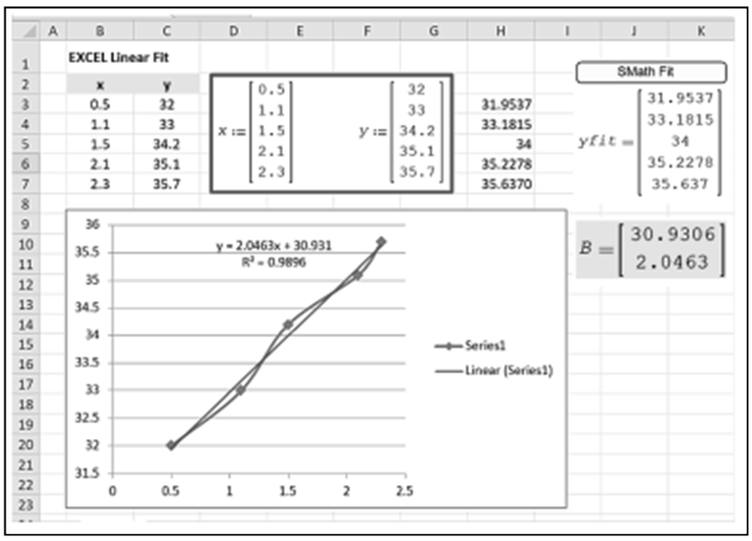
$$Data := \begin{bmatrix} 0.5 & 32 \\ 1.1 & 33 \\ 1.5 & 34.2 \\ 2.1 & 35.1 \\ 2.3 & 35.7 \end{bmatrix}$$

$[X \; Y] := MCols(Data)$

$$C := PolyFit(X, Y, n) = \begin{bmatrix} 30.9306 \\ 2.0463 \end{bmatrix}$$

$R2(X, Y, n) = 0.9895656$

$$PolyVal(C, X) = \begin{bmatrix} 31.9537 \\ 33.1815 \\ 34 \\ 35.2278 \\ 35.637 \end{bmatrix}$$




Amarasekera Excel screenshot


## Quadratic Polyfit

$n := 2$

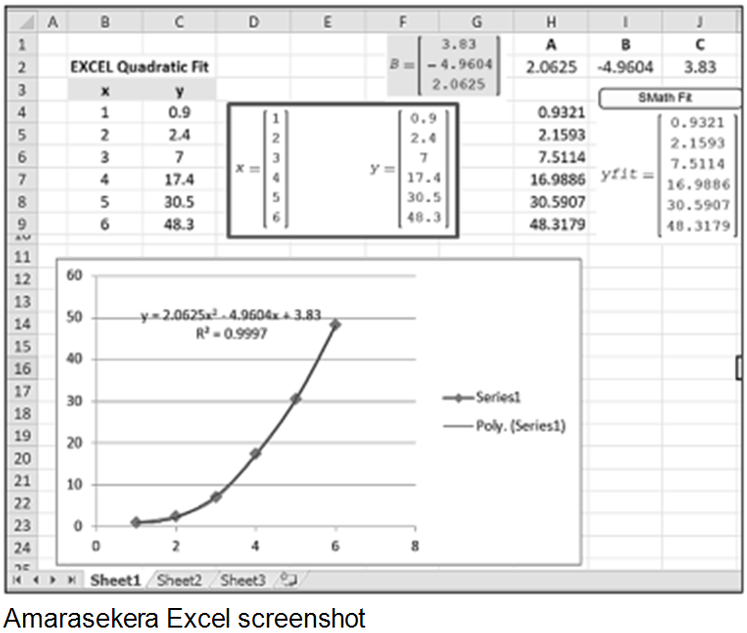
$$Data := \begin{bmatrix} 1 & 0.9 \\ 2 & 2.4 \\ 3 & 7 \\ 4 & 17.4 \\ 5 & 30.5 \\ 6 & 48.3 \end{bmatrix}$$

$[X \; Y] := MCols(Data)$

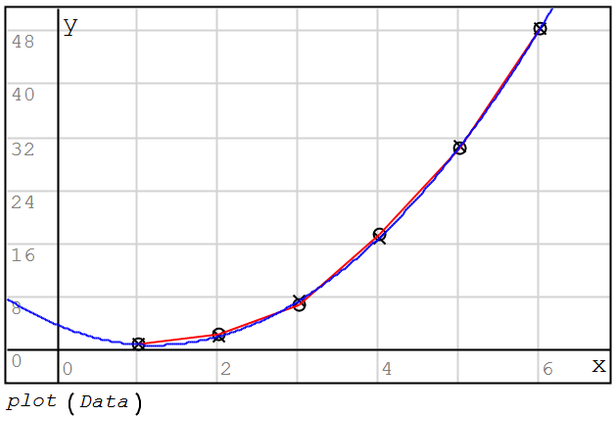
$$C := PolyFit(X, Y, n) = \begin{bmatrix} 3.83 \\ -4.9604 \\ 2.0625 \end{bmatrix}$$

$R2(X, Y, n) = 0.9997121$

$$PolyVal(C, X) = \begin{bmatrix} 0.9321 \\ 2.1593 \\ 7.5114 \\ 16.9886 \\ 30.5907 \\ 48.3179 \end{bmatrix}$$




Amarasekera Excel screenshot



## Matlab example

$n := 7$

https://la.mathworks.com/help/matlab/ref/polyfit.html?lang=en

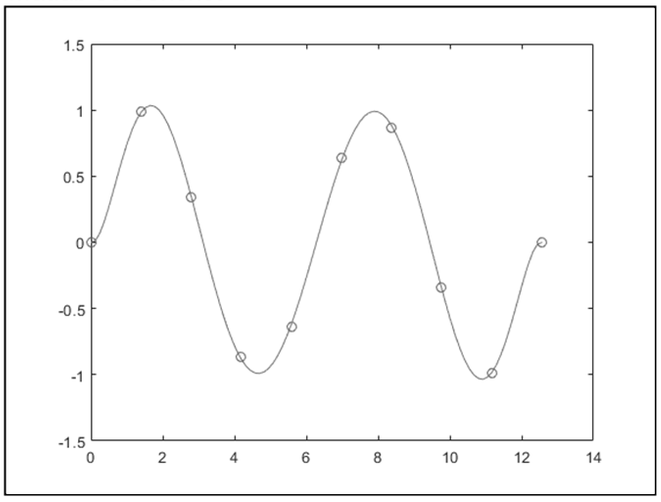
$f(x) := \sin(x)$

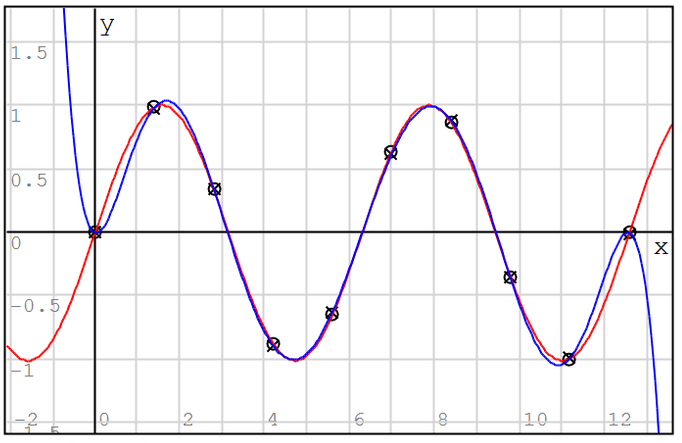
$X := \left[0, \frac{4}{9} \cdot \pi .. (4 \cdot \pi + 0.1)\right]$

$Y := \overrightarrow{f(X)}$

$R2(X, Y, n) = 0.9995795$

Matlab screenshot




$plot(f(x))$

$$PolyFit(X, Y, 7) = \begin{bmatrix} 0.0002 \\ -0.1141 \\ 1.9084 \\ -1.3808 \\ 0.3702 \\ -0.0464 \\ 0.0028 \\ -6.2606 \cdot 10^{-5} \end{bmatrix}$$

```
x = linspace(0,4*pi,10);
y = sin(x);
p = polyfit(x,y,7)
```

```
p = 1×8
   -0.0001    0.0028   -0.0464    0.3702   -1.3808    1.9084   -0.1141    0.0002
```

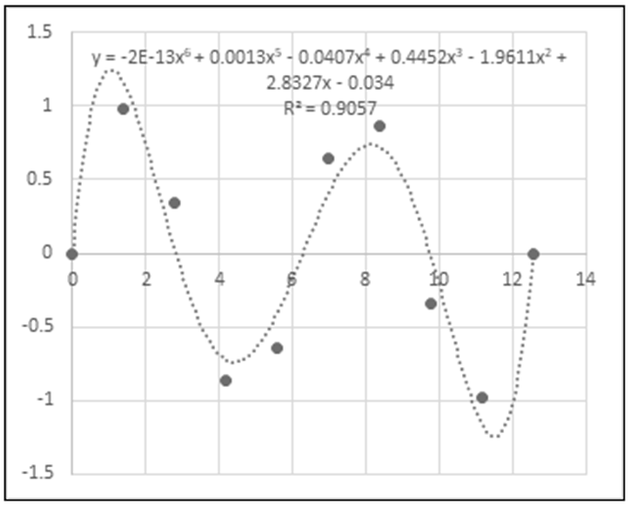
Matlab screenshoot

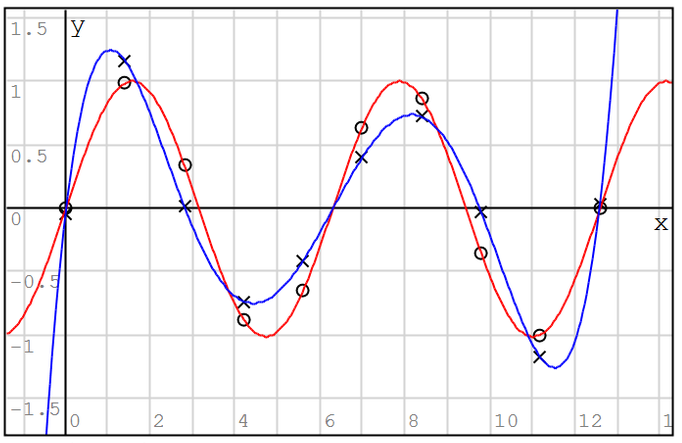
## Matlab example in Excel

Notice that Excel accept only 6 as maximum n $n := 6$

$R2(X, Y, n) = 0.9057328$

Excel screenshot




$plot(f(x))$

The coefficients are the same in SMath, Matlab and Excel

$$PolyFit(X, Y, 6) = \begin{bmatrix} -0.034 \\ 2.8327 \\ -1.9611 \\ 0.4452 \\ -0.0407 \\ 0.0013 \\ -2.2 \cdot 10^{-14} \end{bmatrix}$$

```
x = linspace(0,4*pi,10);
y = sin(x);
p = polyfit(x,y,6)
```

```
p = 1×7
   -0.0000    0.0013   -0.0407    0.4452   -1.9611    2.8327   -0.0340
```

Matlab screenshoot

*Alvaro*