I. Linear Regression:

Suppose the data consists of paired points \( \{x_i\}, \{y_i\} \). The regression refers to finding the coefficient \( c_i \) of the following equation and to minimize the sum of squares of the deviations of the calculated and observed \( y \).

\[
y = \sum_i c_i f_i(x)
\]

where \( c_i \) and \( f_i(x) \) are the coefficient and arbitrary functions of \( x \), respectively.

"Fitting data to a linear function" of the form \( y = ax + b \) where \( a \) is the slope and \( b \) the intercept.

Example 1:

You need first to define the number of points \( N \) and then the range variable \( i \) (index):

\[
N := 10 \quad i := 0..N
\]

Data:

\[
\begin{array}{c|c}
  x_i & y_i \\
  \hline
  -5 & -11 \\
  -4 & -8 \\
  -3 & -6.5 \\
  -2 & -4 \\
  -1 & -1.3 \\
    0 & 0 \\
    1 & 1.1 \\
    2 & 4 \\
    3 & 6.2 \\
    4 & 7 \\
    5 & 8.1 \\
\end{array}
\]

\[
a := \text{slope}(x,y) \quad a = 1.927
\]

\[
b := \text{intercept}(x,y) \quad b = -0.4
\]

\[
\text{fit}(z) := a \cdot z + b
\]

\[
z := -6,-4.5..6
\]
II. Least Squares Fits:

Suppose the dataset is \{(x_i, y_i)\} with \(i = 1, 2, 3, \ldots, n\). And you need to find the best polynomial fit to these data points in the form of \(f(z)\):

\[
f(z) = \sum_{j=0}^{d} (c_j z^j)
\]

equivalent to

\[
f(z) = c_0 + c_1 z + c_2 z^2 + c_3 z^3
\]

with \(d = 3\)

The best fit of \(y(x)\) to the data is found by minimizing the squared deviation function:

\[
M = \sum_{i=1}^{n} \left[ y_i - \sum_{j=0}^{d} c_j (x_i^j) \right]^2
\]

equivalent to

\[
M = (y_1 - c_0 + c_1 x_1)^2 + (y_2 - c_0 + c_1 x_2)^2
\]

with \(n = 2\) and \(d = 1\)

Define a function \(X\) such as \(X_{i,j} = (x_i^j)\) and a matrix \(M\) such as \(M = X^T \cdot X\)

Then the values of \(c_j\) that minimize \(M\) are given by

\[
c = M^{-1} \cdot X \cdot y
\]

Example 2:

\[
x := \begin{pmatrix} -2.5 & -1.0 & -0.2 & 0.5 & 2.2 & 3.9 & 5.6 \end{pmatrix}^T \quad y := \begin{pmatrix} 5.2 & 2.9 & 1.1 & 1.6 & 5.75 & 18 & 29 \end{pmatrix}^T
\]

\[
x = \begin{pmatrix} 0.5 \\ 2.2 \\ 3.9 \\ 5.6 \end{pmatrix} \quad y = \begin{pmatrix} 1.6 \\ 5.75 \\ 18 \\ 29 \end{pmatrix}
\]

Using the functions defined above

\[
X_{i,j} = (x_i^j) \quad \text{and} \quad M := X^T \cdot X
\]

\[
M = \begin{pmatrix} 7 & 8.5 & 58.95 \\ 8.5 & 58.95 & 229.075 \\ 58.95 & 229.075 & 1.278 \times 10^3 \end{pmatrix}
\]

\[
M^{-1} = \begin{pmatrix} 0.247 & 0.029 & -0.017 \\ 0.029 & 0.059 & -0.012 \\ -0.017 & -0.012 & 3.679 \times 10^{-3} \end{pmatrix}
\]

\[
X_{i,j} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}
\]

\[
X = \begin{pmatrix} 1 & -2.5 & 6.25 \\ 1 & -1 & 1 \\ 1 & -0.2 & 0.04 \\ 1 & 0.5 & 0.25 \\ 1 & 2.2 & 4.84 \\ 1 & 3.9 & 15.21 \\ 1 & 5.6 & 31.36 \end{pmatrix}
\]
Recall: example: 3x3 matrix

\[
\begin{pmatrix}
1 & 0 & 3 \\
9 & 0 & 6 \\
7 & 8 & 9
\end{pmatrix}
\]

\[
A^T = \begin{pmatrix}
1 & 9 & 7 \\
0 & 0 & 8 \\
3 & 6 & 9
\end{pmatrix}
\]

\[
A^T A = \begin{pmatrix}
1 & 0 & 3 \\
9 & 0 & 6 \\
7 & 8 & 9
\end{pmatrix}
\]

\[
A^{-1} = \begin{pmatrix}
-0.286 & 0.143 & 0 \\
-0.232 & -0.071 & 0.125 \\
0.429 & -0.048 & 0
\end{pmatrix}
\]

\[
|A| = 168 \quad \text{nonsingular (invertible) determinant is not zero}
\]

Back to our problem, the coefficients \(c\) are given by

\[
c := M^{-1} A^T y
\]

\[
c = \begin{pmatrix}
1.681 \\
0.556 \\
0.798
\end{pmatrix}
\]

Polynomial fitting function:

\[
f(z) := \sum_{j=0}^{d} (c_j z^j)
\]

Use of "\textit{linfit}" function (MathCAD built in function)

Here \(y(x) = c_0 + c_1 x + c_2 x^2\). We need to define a function \(Y(x)\) which gives the power of the variable \(x\).

The \textit{linfit}(x,y,Y) function returns a vector containing the parameters used to create a linear combination of the functions in vector \(Y\) that best approximates the data in \(x\) and \(y\) in the least-squares sense. There must be at least as many data points as there are terms in \(Y\). \(Y(x)\) is a vector of functions; each element is one linear functional term in the fit function. In the case of a single linear function, \(Y\) is a scalar.

\[
Y(x) := \begin{pmatrix}
1 \\
x \\
x^2
\end{pmatrix}
\]
\[
\begin{bmatrix}
1.681 \\
0.556 \\
0.798
\end{bmatrix}
\]

\[
c := \text{linfit}(x, y, Y)
\]

\[
\text{linfit}(x, y, Y) := \begin{bmatrix}
0 & 9.1 \\
1 & 7.3 \\
2 & 3.2 \\
3 & 4.6 \\
4 & 4.8 \\
5 & 2.9 \\
6 & 5.7 \\
7 & 7.1 \\
8 & 8.8 \\
9 & 10.2
\end{bmatrix}
\]

III. Polynomial Regression:

_Polynomial function that best fit the set of data points_

The function \texttt{regress(X,Y,k)} returns a vector which \texttt{interp} uses to find the \textit{kth} order polynomial that best fits the \textit{x} and \textit{y} data values.

The function \texttt{interp(s,X,Y,x)} returns interpolated \textit{y} value corresponding to \textit{x}.

\textit{X} is a vector of real data values in ascending order (the \textit{x} values).
\textit{Y} is a vector of real data values (the \textit{y} values).
\textit{s} is a vector generated by the function \texttt{regress}.
\textit{k} is a positive integer specifying the order of the polynomial you want to use to fit the data. Usually you'll want \textit{k < 5}.
\textit{x} is the value of the independent variable at which you want to evaluate the regression curve.

_Notes:_
- \texttt{Regress} is useful when you have a set of measured \textit{y} values corresponding to \textit{x} values and you want to fit a single polynomial of any order to those \textit{y} values.
- You should always use \texttt{interp} after using the \texttt{regress} function.
- Since \texttt{regress} tries to accommodate all your data points using a single polynomial, it will not work well when your data does not behave like a single polynomial.

Example 3: Consider a matrix of X-Y data to be analyzed (X coordinate in first column, Y coordinate in second)
X := data \langle 0 \rangle \\
Y := data \langle 1 \rangle \\
n := rows(data) \quad n = 10

**Enter the degree of polynomial to fit, value of  k:** \quad k := 2

**Define the polynomial function that fit best the set of data points using "regress" as follows:**

\[
z := \text{regress}(X, Y, k)
\]

\[
z = \begin{pmatrix}
3 \\
3 \\
2 \\
8.698 \\
-2.341 \\
0.288
\end{pmatrix}
\]

**Use the function "interp" as the polynomial fitting function:**

\[
\text{fit}_1(x) := \text{interp}(z, X, Y, x)
\]

\[
x := 0, 1 \ldots 10
\]

**interpolation of y value corresponding to x to give a vector vs**

\[
\text{coeffs} := \text{submatrix}(z, 3, \text{length}(z) - 1, 0, 0)
\]

\[
\text{coeffs} = \begin{pmatrix}
8.698 \\
-2.341 \\
0.288
\end{pmatrix}
\]

**The coefficients are given by:**

\[
\text{coeffs}^T = \begin{pmatrix}
8.698 \\
-2.341 \\
0.288
\end{pmatrix}
\]

\[
\text{Fit}(x) := \sum_{i=0}^{k} (\text{coeffs}_i \cdot x_i)
\]