Example of an APL (A Programming Language) "program."

This function does some statistical calculations on a set of vectors representing data points.   In the fine tradition of APL, there are no comments in the program itself.

$\nabla ITER[\Box]\nabla$
[0] $ITER$
[1] $P \leftarrow 1 - \div 1 + *TH[1] + TH[2] \times D$
[2] $MU \leftarrow N \times P$
[3] $V \leftarrow MU \times 1 - P$
[4] $S \leftarrow (+/Y - MU), +/D \times Y - MU$
[5] $IN \leftarrow 2\ 2\ \rho\ (+/V), (2\rho + /V \times D), +/V \times D * 2$
[6] $\Box \leftarrow TH \leftarrow TH + (\boxdiv IN) + . \times S$
∇

What it means:

APL does strictly right-to-left evaluations, with no operator precedence. Any variable could be a scalar, a vector, or a matrix of any dimension. It is assumed that vectors *TH, N, D,* and *Y* were assigned values interactively before function *ITER* is run.

∇ denotes a function definition and termination. In actual use, it switches the user out of, and back into interactive execution mode.

Line [1] does: $p_i = 1 - \dfrac{1}{1 + e^{TH[1] + d_i TH[2]}}$, for each element of vector *d*. Since APL uses $\times$ for multiplication, $*$ is the exponential operator. With no base, *e* is the default. With no dividend for the $\div$ operator, 1 is the default.

Line [2] does: $\mu_i = n_i p_i$ for all *i*.
Line [3] does: $v_i = \mu_i (1 - p_i)$ for all *i*.

Line [4] forms: $S = \begin{bmatrix} \sum_i (y_i - \mu_i) \\ \sum_i d_i (y_i - \mu_i) \end{bmatrix}$. The operator combination '+/' represents "sum over" a vector or a matrix row/column. The ',' operator

forms a vector of the two sums. Note the placement of the parentheses with strict right-to-left evaluation.

Line [5] forms: $IN = \begin{bmatrix} \sum\limits_i v_i & \sum\limits_i d_i v_i \\ \sum\limits_i d_i v_i & \sum\limits_i d_i^2 v_i \end{bmatrix}$. The interesting operator here is the "rank" operator 'ρ'. In the middle part, (2ρ+/V×D) means "form a vector of size 2 of my right-hand side. Since there is only one item there, it gets duplicated twice. With the ',' operator, a vector of 4 elements is created, with the middle two being the same. Finally, 2 2 ρ … forms this one-dimensional vector of 4 elements into a two-dimensional (indicated by the length of the vector '2 2') matrix, in a 2 by 2 grid.

Line [6] does $TH = TH + IN^{-1} \times S$: Recall that TH is a vector with 2 elements, S is a vector with 2 elements, and IN is a 2-by-2 matrix. The operator ⌹ finds the inverse (!) of a matrix. The operator combination '+.×' represents an algebraic dot-product. In this case, it does the matrix-vector multiplication.

Finally, the □ symbol is the output operator, and "assigning" something to this operator has the effect of printing the result. This also is used in the function header to indicate that the function will produce a result.

By the way, all intermediate values used in the function remain assigned after the function has been executed. In fact, this particular function was meant to be iterated (hence the name) interactively, until the results were satisfactory.