

[3.29] Convert data variables to matrices

You can use the built-in *NewData* command to convert matrices to data variables, but there is no complimentary command to convert data variables to matrices. This function does it:

```
datamat(mσ)
Func
©("name") convert data variable to matrix
©22dec01/dburkett@infinet.com

local mθ,k,l,d,d1

list←mat(#mσ[1],1)→mθ          © Convert first data column to matrix
rowdim(mθ)→d                  © Find row dimension

2→k                            © Loop through remaining data columns
loop
#mσ[k]→l                       © Convert column to a list
dim(l)→d1                       © Find row dimension
if d1=0 then                     © Done when row dimension is zero
  exit
elseif d1≠d then                 © Error if row lengths not equal;
  return "datamat() err"        © return error string
else                             © Augment current column to matrix
  augment(mθ,list←mat(l,1))→mθ
endif
k+1→k                            © Process next column
endloop

mθ                              © Return matrix

EndFunc
```

This conversion is useful if you want to use functions on data variables, because data variables cannot be passed as function arguments. Use *datamat()* to convert the data variable to a matrix, then execute the desired function.

To convert a data variable *mydata* to a matrix, use

```
datamat("mydata")
```

Note that the name of the data variable is passed as a string.

The code is straightforward, but is somewhat complicated because we cannot find the number of columns in a data variable. However, I use the fact that a data variable column can be extracted to a list with

```
datavar[k]→list
```

and the dimension of *list* is zero if column *k* does not exist. This condition exits the loop.

The only error checking ensures that all the data variable columns have the same number of rows, since this must be true for matrices. If the row dimensions are not equal, *datamat()* returns the string "*datamat() err*" instead of the matrix. The calling program or function can use *getType()* on the result to determine if an error occurred.