

[3.8] Work-around for writing to data variable elements

There are no built-in instructions to write to single elements in data variables. This can be limiting, since data variables are used in the statistics and regression functions. However, you can write to elements indirectly with this procedure:

```
datalist(adata)
prgm
©Convert data variable to matrix
©31dec99/dburkett@infinet.com
©adata: contains data variable name as string

local alist1,alist2,amatrix

©Convert data variable 'adata' to one list for each column
#adata[1]→alist1
#adata[2]→alist2

©Convert lists to matrix
augment(list▶mat(alist1,1),list▶mat(alist2,1))→amatrix

©Modify list or matrix elements as needed, for example:
3→amatrix[2,1]
4→amatrix[2,2]

©Convert lists to data variable
newdata #adata,amatrix

Endprgm
```

The basic idea is to convert the data variable to a matrix, change the elements as needed, then convert the matrix back to a data variable. This program assumes that the data variable has two columns with an equal number of rows. Since the program refers to the data variable, indirection is used (the # operator), and the data variable name is passed as a string. For example, to modify a data variable named *tdata*, call *datalist()* like this:

```
datalist("tdata")
```

Note that the columns of data variables can be accessed with the column number in square brackets:

```
#adata[1]
```

returns the first column of the data variable named in *adata*. The column is returned as a list.

Note that this method will not work in a function, because the *newdata* instruction cannot be used in a function.

It is not necessary to convert the data variable to a matrix. You can just convert the variable to lists, as shown in the example, make the necessary changes to the list elements, then convert the lists back to a data variable like this:

```
NewData #adata,alist1,alist2
```