A Remark on Algorithm AS57 – Printing Multidimensional Tables

By Tim Hopkins and David Morse
Computing Laboratory, University of Kent,
Canterbury, Kent CT2 7NF, UK.

Keywords: Table; Printing; Multidimensional.

LANGUAGE

Fortran 77.

DESCRIPTION AND PURPOSE

This routine is a Fortran 77 version of that appearing in Griffiths and Hill (1985) (originally Haberman (1973)). It uses character variables rather than storing character information in integers and reals which was the only way of implementing such an algorithm in standard Fortran 66. Where possible data statements have been replaced by parameter statements. The code has been restructured and variable names have been altered to avoid clashes with Fortran 77 intrinsic function names and the Fortran specifier name $UNIT$.

The algorithm prints one or more $NVAR$-dimensional parallel tables stored in an array $TABLE$ of length $NTAB$. For example, $TABLE$ might contain a table of observations, a table of fitted values and a table of residuals. To be specific, suppose that a three-dimensional table $\{n_{ijk}\}$ has been studied, where $1 \leq i \leq 4$, $1 \leq j \leq 3$, $1 \leq k \leq 2$. From this investigation, a fitted table $\{m_{ijk}\}$ and a residual table $\{r_{ijk}\}$ have been derived. For purposes of display it is desired that the printed table have the format shown in Figure 1. In this case, $TABLE$ is divided into three subtables as follows.

The subtable corresponding to the observations $\{n_{ijk}\}$ begins at $TABLE(1)$, the subtable for the fit $\{m_{ijk}\}$ begins at $TABLE(4 \times 3 \times 2 + 1 = 25)$ and the subtable of residuals $\{r_{ijk}\}$ begins at $TABLE(49)$. Each subtable is arranged in standard Fortran fashion; that is, $TABLE(1) = n_{111}$, $TABLE(2) = n_{211}$, $TABLE(3) = n_{311}$, etc. (Note: that standard Fortran 77 would allow $TABLE$ in the calling routine to be a four-dimensional array dimensioned (4,3,2,3) where the last dimension denotes the number of tables.)

To permit proper printing of the table, the algorithm requires descriptive information concerning the structure of the table, and information which indicates how the table should be displayed. The basic structural information is provided by an integer array DIM of length $NVAR$ and an integer array LOC of length COL. The array DIM gives the number of categories in each of the

		Var 1.	Cat. 11	Cat. 12	Cat. 13	Cat. 14
Var 3.	Var 2.					
Cat. 31	Cat. 21	Res.	n_{111}	n_{112}	n_{113}	n_{114}
		Fit.	m_{111}	m_{112}	m_{113}	m_{114}
		Obs.	r_{111}	r_{112}	r_{113}	r_{114}
Cat. 31	Cat. 22	Res.	n_{121}	n_{122}	n_{123}	n_{124}
		Fit.	m_{121}	m_{122}	m_{123}	m_{124}
		Obs.	r_{121}	r_{122}	r_{123}	r_{124}
Cat. 31	Cat. 23	Res.	n_{131}	n_{132}	n_{133}	n_{134}
		Fit.	m_{131}	m_{132}	m_{133}	m_{134}
		Obs.	r_{131}	r_{132}	r_{133}	r_{134}
Cat. 32	Cat. 21	Res.	n_{211}	n_{212}	n_{213}	n_{214}
		Fit.	m_{211}	m_{212}	m_{213}	m_{214}
		Obs.	r_{211}	r_{212}	r_{213}	r_{214}
Cat. 32	Cat. 22	Res.	n_{221}	n_{222}	n_{223}	n_{224}
		Fit.	m_{221}	m_{222}	m_{223}	m_{224}
		Obs.	r_{221}	r_{222}	r_{223}	r_{224}
Cat. 32	Cat. 23	Res.	n_{231}	n_{232}	n_{233}	n_{234}
		Fit.	m_{231}	m_{232}	m_{233}	m_{234}
		Obs.	r_{231}	r_{232}	r_{233}	r_{234}

Figure 1: Sample test.

table variables. Thus $DIM(1) = 4$, $DIM(2) = 3$ and $DIM(3) = 2$ in the example. The array LOC gives the location in $TABLE$ of each of the COL parallel tables. In the example, $COL = 3$, $LOC(1) = 1$, $LOC(2) = 25$ and $LOC(3) = 49$.

Label information is required for the table variables, the categories of each variable and the names of the parallel tables. In addition, the title for the complete display is needed. The character arrays $TITLE$, $VARNAM$, $CATNAM$ and $COLLAB$ provide these data. All label arrays are declared to be assumed size character arrays; the variables $TITLEN$ and $LABLEN$ control how much of each label is actually printed as well as the field width used for output of the labels. Thus the first $TITLEN$ characters of each of the $NT2$ lines of the title are output and the maximum length of any label is $LABLEN$ characters. The integer constant GAP (see Adjustable constants below) controls the amount of space inserted between a label and the following data values. Of the $NVAR$ labels of $VARNAM$, the last $VERT$ of them are printed on the left-hand side of the page, while $NVAR - VERT$ labels are printed along the top. In the example, $VERT = 2$.

The body of the table is printed by use of F-conversion. The format for an entry from parallel table J is $Fa.b$, where a is $LABLEN + GAP$ and b is $DEC(J)$. The integer array DEC has length COL.

When many tables are to be printed, it is sometimes useful to print several small tables on the same page. To permit this practice, the user may use the value of the line counter $LINE$ as returned by a prior call to $TABWRT$ along with the argument $RESTOR$. If $RESTOR$ is set $.FALSE.$, the table will be printed, space permitting, within the remainder of the page. If $RESTOR$ is set $.TRUE.$ the new table will be printed at the top of a new page.

STRUCTURE

```
      SUBROUTINE TABWRT(TITLE,TITLEN,NT2,TABLE,NTAB,DIM,NVAR,
     +     LOC,COL,DEC,VARNAM,LABLEN,CATNAM,MAXCAT,COLLAB,
     +     VERT,RESTOR,LINE,SKIP,PAGE,WIDTH,UNIT,IFAULT)
```

Formal parameters

$TITLE$	Character*(*) array ($NT2$)	input:	the title of the table.
$TITLEN$	Integer	input:	the number of characters per line of the title.
$NT2$	Integer	input:	the number of lines in the title.
$TABLE$	Real array ($NTAB$)	input:	the tables to be printed.
$NTAB$	Integer	input:	the number of elements in $TABLE$.
DIM	Integer array ($NVAR$)	input:	the number of categories in each variable of the table.
$NVAR$	Integer	input:	the number of variables (dimensions) in the table.
LOC	Integer array (COL)	input:	the locations in $TABLE$ of the subtables to be printed.
COL	Integer	input:	the number of subtables to be printed.
DEC	Integer array (COL)	input:	the number of places to the right of the decimal point for each subtable.
$VARNAM$	Character*(*) array ($NVAR$)	input:	the variable names.

$LABLEN$	Integer	input:	the maximum number of characters of each label to be printed.
$CATNAM$	Character*(*) array ($MAXCAT$)	input:	the names of the variable categories.
$MAXCAT$	Integer	input:	the maximum number of categories per variable.
$COLLAB$	Character*(*) array (COL)	input:	the table names.
$VERT$	Integer	input:	the number of labels to be printed on the left side of the page.
$RESTOR$	Logical	input:	If $.TRUE.$ the table will be printed on a new page, if $.FALSE.$ then an attempt will be made to print the table on the same page (see *Description and Purpose* for details).
$LINE$	Integer	input: and output:	the line counter.
$SKIP$	Integer	input:	the number of lines between tables on the same page.
$PAGE$	Integer	input:	the number of lines per page.
$WIDTH$	Integer	input:	the number of characters per line (excluding carriage control character).

$UNIT$	Integer	input:	the number designating the output device.
$IFAULT$	Integer	output:	1 if $NVAR$, COL, or DIM is incorrectly specified; 2 if table will not fit on the defined page; 3 if $TITLEN > WIDTH$ or $LABLEN > WIDTH - GAP$; 4 if $RESTORE$ is $.FALSE.$ and $LINE < 0$; 0 otherwise.

Adjustable constants
The following values are defined in $PARAMETER$ statements:

$MAXVAR$	Integer	constant:	maximum number of variables permitted in a table.
GAP	Integer	constant:	number of characters between label and start of data values.

RESTRICTIONS

None.

PRECISION

It does not seem worth making this algorithm adjustable for double precision. If double precision data are to be used, values should first be copied into a single precision array.

REFERENCES

Griffiths, P. and Hill, I. D., editors (1985) *Applied statistics algorithms.* Chichester: Ellis Horwood.
Haberman, S. J. (1973) Printing Multidimensional Tables. *Appl. Statist*, **22**, 118–126.