Paper 121-29

Tricky Aspects of the Data Step

Philip Mason, Wood Street Consultants, Oxfordshire, England

ABSTRACT

This workshop aims to cover a range of features in the data step which are not used very much or are often misunderstood, such as:

- Useful options on statements such as libname, filename, infile & file
- Things you can do with PUT & INPUT
- Interfacing with macro language from data step
- Doing ODS from data step
- Unusual functions & formats
- Basic GUIs with window & display

INTRODUCTION

This workshop aims to cover a range of features in the data step which are not used very much or are often misunderstood. For each section we will look through there will be an example of some code to demonstrate the features we are looking at.

USES OF DATA _NULL_

- Data steps can be used to produce reports, as a last resort. Only use a data step when a procedure is unable to produce what is required
- Data step reports are more complicated to produce, but extremely flexible
- Reports can be produced using ODS, or not
- Reports can be sent directly to window, external file, printer, web site, etc.

OTHER USES OF DATA _NULL_

- Produce macro variables to be used elsewhere based on data values
- Create files for export to other applications (e.g. CSV)
- To generate SAS code to run via use of CALL EXECUTE
- generating commands to run

USEFUL STATEMENTS WITH DATA _NULL_ REPORTS

FILE
Directs output to a location

FILENAME
Defines a location

ODS
Defines type of file to be written

PUT
Writes values to location

BY
Defines grouping of data being read

RETAIN
keeps values of variables between iterations

SPECIAL VARIABLES

_N_
number of current data step iteration

ERROR
error flag

FIRST.*BY_VAR*
first occurrence of BY variable

LAST.*BY_VAR*
last occurrence of BY variable

FILE STATEMENT SYNTAX

FILE-SPECIFICATION CAN BE

- External file, e.g. 'c:/myfile.txt'
- Fileref, e.g. MYLIB
- Member of fileref, e.g. MYLIB(file1)
- LOG … the SAS log
- PRINT … the SAS output destination

FILE STATEMENT OPTIONS

COLUMN=VAR
sets var to current column number

DELIMITER=','|VAR
sets delimiter to a character (e.g. comma) or variable value

DROPOVER
drop data too long for line

DSD
write data items with delimiters

FILENAME=VAR
var is set to physical file name in use

FILEVAR=VAR
var determines the physical file to be written to

FLOWOVER
anything that doesn't fit on this line is written to the next

[NO]FOOTNOTES
prints footnotes

HEADER=LABEL
run statements at label for each new page

LINE=VAR
contains current logical line number

LINESIZE=
columns per line

LINESLEFT=VAR
contains lines left on page

MOD
write after whatever is in file

ODS=
Defines ODS sub-options (see later)

OLD
wipes file before writing

PAGESIZE=
lines per page

STOPOVER
stops data step if trying to write beyond end of line

[NO]TITLES
prints titles

FILE=VAR
contains current output buffer and is Read/Write

FILE EXAMPLES

FILENAME SYNTAX

fileref should be a SAS name
device-type has many values

- Disk
- Dummy
- Pipe
- Printer
- Temp
- Catalog
- Socket
- FTP
- URL

FILENAME EXAMPLES

ODS STATEMENT

Can write to various ODS destinations.

PUT SYNTAX EXAMPLES

- `Put x y z ;`
- `Put ‘hello’ ’09’x ;`
- `Put 132*’_’ ;`
- `Put #3 @44 cost ;`
- `Put var 1-5 ;`
- `Put cost dollar12.2 ;`
- `Put (a b) (1. ‘,’ $3.) ;`
- `Put _infile_ ;`
- `Put _all_ ;`
- `Put _ods_ ;`
- `Put a b c @ ;`
- `Put d e @@ ;`
- `Put @10 name ;`
- `Put @pos name ;`
- `Put @(3*pos) name ;`
- `Put a +3 b ;`
- `Put a +gap b ;`
- `Put a +(2*gap) b ;`
- `Put #2 text ;`
- `Put #line text ;`
- `Put #(line*3) text ;`
- `Put line1 / line2 ;`
- `Put @1 title overprint`
  `@1 ‘_____’ ;`
- `Put _blankpage_ ;`
- `Put _page_ ;`
- `Put name= phone= ;`
- `Put my_big_array(*) ;`

BY SYNTAX & OPTIONS

ASCENDING IS DEFAULT

Descending should be specified for each variable

GROUPFORMAT INDICATES THAT FORMATTED VALUES, RATHER THAN STORED VALUES,

Groupformat should be used with first.var & last.var.

NOTSORTED INDICATES THAT DATA IS GROUPED TOGETHER, BUT NOT SORTED

Ideal for dealing with data such as months, where you want to keep it in the correct sequence: Jan, Feb, Mar, Apr, …

BY EXAMPLES

- Simple data _null_ report
- Specifying precise column positions
- Doing Custom headers
- Doing custom footers
- Do a data _null_ report of the derived dataset from a previous table shell exercise
- Get it as close as possible in layout to the table shell
- Save the LOG & OUTPUT to external files

CONCLUSION

The data step is a powerful tool that enables you to read, manipulate and write almost any kind of data in any way. Learning more advanced features of the data step will greatly extend what can be done with it and is well worth the investment of time and effort.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. You can find an updated version of the paper at my web site (see below). Contact the author at:

Philip Mason
Wood Street Consultants
16 Wood Street
Wallingford, Oxfordshire, OX10 0AY, ENGLAND
Work Phone: +44 1491 824905
Email: how@woodstreet.org.uk
Web: www.woodstreet.org.uk