

CREATING ASCII FIXED LENGTH RECORD FILES FROM SAS®

DATA SETS

Lawrence Altmayer
U.S. Bureau of the Census

INTRODUCTION

Statistical software packages sometimes require ASCII input files with fixed length records. One such package is VARCL, designed for hierarchical linear modelling. The work described here was done to model interviewer effect on factors thought to influence survey response rates. A method was needed for providing the number of cases for each interviewer, in a fixed format, according to VARCL input specifications.

The concepts described here can be used separately or together, to approach two data processing needs users may have. The first need is to carry frequencies from one SAS data set, as a variable into a second data set. The second need is to output a variable from a SAS data set into an ASCII file, in a fixed format. We give two methods for creating the ASCII file.

CARRYING FREQUENCIES

One method for using frequencies from one data set in another, is to use the OUT= option on the TABLES statement in PROC FREQ. This option outputs a data set containing the nonzero frequencies of the levels of the variable in the TABLES statement. The NOPRINT option suppresses printing of the output listing from PROC FREQ. The frequencies of the variable requested in the TABLES statement have the variable name "COUNT" in the output data set BHI.

```
PROC FREQ DATA=LEVEL1;  
TABLES IWER / NOPRINT OUT=BHI;  
RUN;
```

OUTPUT TO AN ASCII FILE - FIRST METHOD

The usual method for outputting a SAS data set to an ASCII file is to use a null data step, where no SAS data set is output. In addition, we use the MOD function, a variable column pointer with @ symbol, and trailing @ symbol.

```
DATA _NULL_;  
SET BHI;  
FILE '[SCM_DATA.VARCL]TEST31.DAT';  
REMAIN=MOD(IWER,6);  
IF REMAIN=0 THEN REMAIN=6;  
COL=1+(REMAIN-1)*5;  
IF REMAIN^=6 THEN  
  PUT @ COL COUNT 5.0 @ ;  
  ELSE PUT @ COL COUNT 5.0 ;  
RUN;
```

In this example, IWER (interviewer) are the ascending integers beginning with 1 from the above output data set BHI. The two statements creating the variable REMAIN produce repeating series of the integers 1,2,3,4,5,6. The COL=... statement produces a resulting series of 1,6,11,16,21,26 for each series of REMAIN=1,2,3,4,5,6. The values of COL will provide the location for values of COUNT using the variable column pointer with @ symbol. The trailing @ symbol in the PUT @ COL... statement indicates we remain on that output line for the next value of COUNT, for REMAIN cases of 1,2,3,4 and 5. For the case of REMAIN=6 (ELSE PUT...), we output the value of COUNT, and then jump to the next line to output the next value of COUNT, and so on.

OUTPUT TO AN ASCII FILE - SECOND METHOD¹

As with the first method, we use a null data step to output the SAS data set to the ASCII file. We again use a variable column pointer with @ symbol, and trailing @ symbol. With this method, a RETAIN statement conditionally carries incremented values of COL into the next iteration of the data step.

This is done for all cases of COL, except the last value on a given line (COL=26).

```
DATA _NULL_;  
  SET BHI;  
  PUT @COL COUNT 5.0 ;  
  COL=-4;  
  END;  
RUN;
```

CONCLUSIONS

The methodology described here can be used to output frequencies from a SAS data set into an ASCII file with fixed format. The two individual components and alternative described use several existing SAS options and features, and are also useful themselves.

AUTHOR

Lawrence Altmayer
U.S. Bureau of the Census
ESMPD, Room 1200-4
Washington, DC 20233-0001
(301)457-2581
laltmaye@census.gov

SAS® is a registered trademark of SAS Institute Inc., Cary, NC

¹ The author would like to thank Ian Whitlock of WESTAT, Inc., for his contribution of this data step method.

```
FILE'[SCM_DATA.VARCL]TEST31.DAT';  
RETAIN COL -4;  
COL+5;  
PUT @COL COUNT 5.0 @ ;  
IF COL=26 THEN DO;
```