Macros

How can I create sample data to share with others? e.g. on a SAS forum

SAS fora are a useful way to glean some 'crowd wisdom' when struggling with a coding dilemma. Sometimes this can be answered in the abstract, but at other times it would be helpful if a sample set of user data could be shared.

This utility macro allows any existing dataset to be incorporated within a DATA STEP which anyone could then run to re-create the data.

The macro retrieves metadata about the target dataset, and then builds a DATA STEP with a format statement based on any existing formats, or standard numeric / character for the remainder. An INPUT statement is also generated to read inline data which is acheived by using the DATALINES syntax. Values are written into the code as standard numeric / character values, to which the formats are then applied to mirror the original dataset.

The two operational parameters for the macro are the dataset name (one- or two-level) and the number of observations to be processed.

The macro writes code to an external file (in the WORK library), and also to the LOG window, from where it can be shared.

```
%macro data2dstep ( help
                   ,dsn  =
                   ,obs  = 10
                  ) ;
options nomprint nomlogic nosymbolgen nonotes ;
%global fmt_stat inp_stat var_stat ;
%let fmt_stat = ;
%let inp_stat = ;
%let var_stat = ;
%local help dsn obs lib dset workpath ;

%if &help = ? or %lowcase(&help) = help %then
%do ;
  options notes ;
  %put NOTE: This macro programme has one HELP paramater , and two ope
rational parameters. ;
  %put NOTE- HELP - specify the first paramter as ? | help | HELP and
useage notes are displayed in the LOG. ;
  %put NOTE- DSN  - specify the dataset to be converted as either a on
e- or two-level name. ;
  %put NOTE- OBS  - specify the number of observations to process.  Th
e default value is 10. ;
  %goto endmac ;
%end ;

%*** Validate DSN *** ;
%let dsn = %upcase(&dsn) ;
```

```
%if not %sysfunc(exist(&dsn)) %then
%do ;
  options notes ;
  %put ERROR: The dataset &dsn does not exist.  Processing will stop.
;
  %goto endmac ;
%end ;

%*** Split two-level name *** ;
%if %index(&dsn,.) %then
%do ;
  %let lib  = %scan(&dsn,1,.) ;
  %let dset = %scan(&dsn,2,.) ;
%end ;
%else
%do ;
  %*** Ascertain if USER library is assigned, otherwise assume WORK fo
r one-level name *** ;
  proc sql noprint ;
    select distinct libname into :lib separated by ''
    from dictionary.libnames
    where libname = 'USER'
    ;
  quit ;
  %if &lib = %then %let lib = WORK ;
  %let dset = &dsn ;
%end ;

data _null_ ;
  set sashelp.vcolumn (keep = libname memname name type length format
informat) ;
    where libname = "&lib" and memname = "&dset" ;
    *** Assign FORMATs for each variable for FORMAT statement *** ;
    if format = '' then format = ifc( type = 'char'
                                     ,cats('$',put(length,3.),'.')
                                     ,'best32.'
                                    ) ;
    else format = lowcase(format) ;
    *** Assign INFORMATs for each variable for INPUT statement *** ;
    if informat = '' then informat = ifc( type = 'char'
                                         ,cats('$',put(length,3.),'.')
                                         ,'best32.'
                                        ) ;
    else informat = lowcase(informat) ;
    *** Assign FORMATs for each variable for PUT statement *** ;
    putfmt = ifc( type = 'char'
                 ,cats('$',put(length,3.),'.')
                 ,'best32.'
                ) ;
```

```
    *** Build statements *** ;
    call symputx('fmt_stat',catx('0A202020202020202020'x,symget('fmt
_stat'),strip(name)!!'09'x!!format       )) ;
    call symputx('inp_stat',catx('0A202020202020202020'x,symget('inp
_stat').strip(name)!!'09'x!!':'!!informat)) ;
    call symputx('var_stat',catx('0A202020202020202020'x,symget('var
_stat'),strip(name)!!'09'x!!':'!!putfmt  )) ;
run ;

%let dset = %lowcase(&dset) ;

%*** Write code lines to external file *** ;
%let workpath = %sysfunc(pathname(work)) ;

data _null_ ;
  file "&workpath\&dset..sas" dsd ;
  set &dsn (obs = &obs) end = eof ;
  if _n_ = 1 then
  do ;
    put @1  "data &dset ;          " ;
    put @3  'format                ' ;
    put @11 "&fmt_stat             " ;
    put @11 ';                     ' ;
    put @3  'infile datalines dsd ; ' ;
    put @3  'input                 ' ;
    put @11 "&inp_stat             " ;
    put @11 ';                     ' ;
    put @3  'datalines ;           ' ;
  end ;
  put &var_stat ;
  if eof then
  do ;
    put @1 ';    ' ;
    put @1 'run ;' ;
  end ;
run ;

options notes ;
%put NOTE: The DATA STEP code has been written to the external file: &
workpath\&dset..sas ;
options nonotes ;

options nodate  ps = 999 ;
%*** Write code lines to the LOG *** ;
data _null_ ;
  infile "&workpath\&dset..sas" ;
  input ;
  put _infile_ ;
run ;
```

```
%endmac:
options notes ;
%mend data2dstep ;
```

The macro can be called:

```
proc datasets lib = sashelp nolist ;
  copy out = work ;
  select class ;
quit ;

%data2dstep(dsn = class)
```

Which produces output code like:

```
data class ;
  format
         Name  $8.
  Sex  $1.
  Age  best32.
  Height      best32.
  Weight      best32.
  ;
  infile datalines dsd ;
  input
  Name  :$8.
  Sex  :$1.
  Age  :best32.
  Height      :best32.
  Weight      :best32.
  ;
  datalines ;
Alfred,M,14,69,112.5
Alice,F,13,56.5,84
Barbara,F,13,65.3,98
Carol,F,14,62.8,102.5
Henry,M,14,63.5,102.5
James,M,12,57.3,83
Jane,F,12,59.8,84.5
Janet,F,15,62.5,112.5
Jeffrey,M,13,62.5,84
John,M,12,59,99.5
```

```
;
run ;
```

Unique solution ID: #1049
Author: Alan D Rudland
Last update: 2019-12-18 15:23