

Code Generator to Check Truncated Values in Derived Datasets

Proma Mukherjee

DISCLAIMER

The opinions expressed in this presentation are solely those of the presenter and not of the employer of the presenter or any third party.

AGENDA

- › Introduction
- › Manual Check
- › Programmatic Check
- › SAS Code
- › Output
- › Scope of Macro
- › Summary

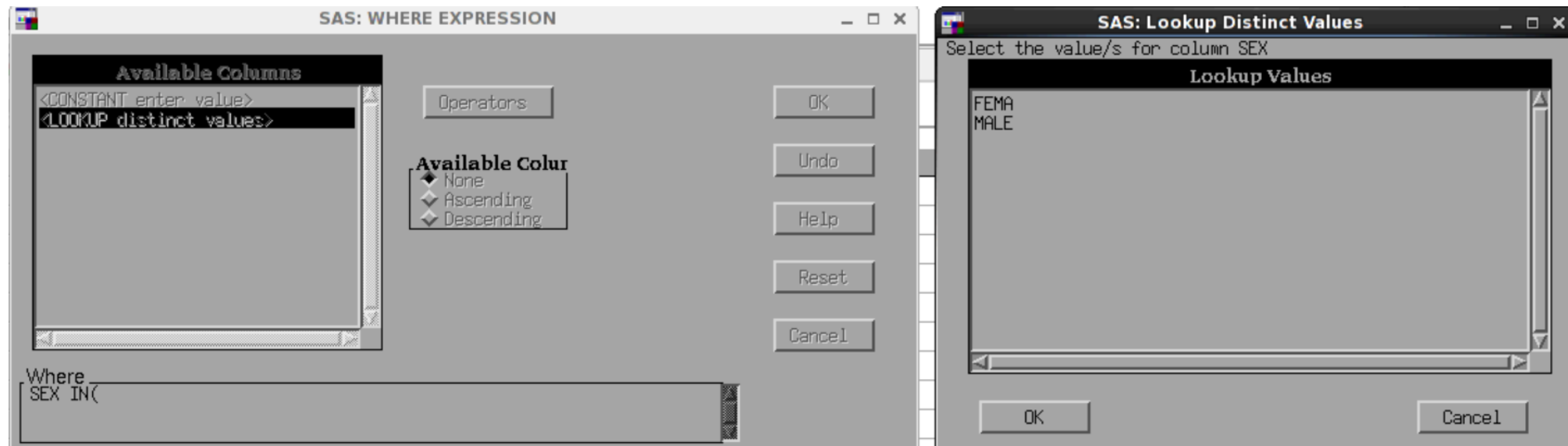
INTRODUCTION

- › While doing dataset programming in SAS®, the log displays notes and messages such as:
 - › Missing values
 - › Uninitialized variables
 - › Repeat of by values in merge statement
 - › Overwritten variables, etc.
- › What the SAS log currently misses out on:
 - › Character variable truncations

MANUAL CHECK

Two methods of validation checks for truncations:

1. Visual inspection of the entire dataset.
2. Filtering the dataset to visually check the longest value of the character variables where one might expect to find truncation.



PROGRAMMATIC CHECK

- › The truncation check macro checks all the character variables irrespective of their length.
- › The maximum length and the corresponding value of each character variable is displayed as a single observation in the resulting 'truncate check' dataset.
 - › If there are multiple observations with the same maximum length, only the first one is displayed.
- › This macro script can be added at the end of the dataset code or can be run in a separate window in the same SAS session.
- › Use the first intermediate datasets instead of the source datasets in the macro call.

SAS CODE

SCRIPT	COMMENT
<pre>%macro truncate_chk (ds =); proc contents data = &ds. out = trchk_&ds.2 (keep = name type memname) noprint; run; data trchk_&ds.2_ (keep = type name2); set trchk_&ds.2 (where = (type = 2)); length name2 \$40.; name2 = compress(name) '_L'; run;</pre>	› This macro requires one parameter which is the dataset name › trchk_&ds.2 contains the variable names and type (for character/numeric variables) › trchk_&ds.2_ contains all the character variables concatenated with '<u>L</u>' in a new variable name2
<pre>proc transpose data = trchk_&ds.2_ out = trchk_&ds.3 (drop = _name_ _label_); id name2; var type; run;</pre>	› Transposed into a one-line dataset trchk_&ds.3 containing numeric variables labelled as '<u>var_name_L</u>' for easier comparison

SAS CODE (CONTD.)

SCRIPT	COMMENT
<pre>data trchk_&ds.4 (keep = memname combo); set trchk_&ds.2 (where = (type = 2)); by memname; length combo \$999; retain combo; if first.memname then combo = compress(name); else combo = compress(catx(' ', combo, name)); if last.memname; run; proc sql noprint; select combo into :cmb from trchk_&ds.4; create table trchk_&ds.5 as select &cmb from &ds.; quit;</pre>	> trchk_&ds.4 contains a single observation with the dataset name and a new variable combo > combo contains all the character variable names from the input dataset concatenated together and separated by commas > combo is used to include all the character variables from the input dataset in trchk_&ds.5

SAS CODE (CONTD.)

SCRIPT	COMMENT
<pre>data trchk_&ds.6; merge trchk_&ds.3 trchk_&ds.5; run;</pre>	› trchk_&ds.6 contains all the character variables from trchk_&ds.5 and the corresponding numeric variables (ending with '_L') from trchk_&ds.3
<pre>proc sql noprint; select count(*) into :cnt from trchk_&ds.2 where type = 2; quit; %let x = &cnt; proc sql noprint; select name into :nm1 - :nm&x from trchk_&ds.2 where type = 2; select name2 into :l1 - :l&x from trchk_&ds.2_; quit;</pre>	› x/cnt is the count of character variables from the input dataset › All the character variable names are stored in the macro variables nm1 till nm&x › All the numeric variable names (ending with '_L') are stored in the macro variables l1 till l&x

SAS CODE (CONTD.)

SCRIPT	COMMENT
<pre>%do i = 1 %to &x; data trchk_&ds._&i (keep = n &&l&i &&nm&i); set trchk_&ds.6; n = 1; &&l&i = lengthn(&&nm&i); run;</pre>	> <code>trchk_&ds._&i</code> contains two variables: character variable and its corresponding numeric variable > The numeric variable captures the length of the character variable
<pre>proc sort data = trchk_&ds._&i; by n descending &&l&i; run; data trchk_&ds._&i._; set trchk_&ds._&i; by n descending &&l&i; if first.n; run; %end;</pre>	> <code>trchk_&ds._&i._</code> is a one-line dataset containing the first instance of the longest character value

SAS CODE (CONTD.)

SCRIPT	COMMENT
<pre>data tr_chk_&ds.; merge %do j = 1 %to &x; trchk_&ds._&j._ (drop = n) %end;; run; proc datasets; delete trchk_&ds;; quit; %mend truncate_chk; %truncate_chk (ds = ae); /*source dataset*/ %truncate_chk (ds = adae); /*derived dataset*/</pre>	➤ All the individual datasets are merged to create a one-line dataset <code>tr_chk_&ds</code> containing the longest value of all the character variables and their corresponding length

OUTPUT

SAS: VIEWTABLE: Work.Tr_chk_ae								
File Edit View Tools Data Solutions Help								
	USUBJID	USUBJID_L	AEACN	AEACN_L	AEBODSYS	AEBODSYS_L	AELLT	AELLT_L
1	123456.987654	13	DOSE NOT CHANGED	16	Gastrointestinal disorders	26	Feeling queasy	14

SAS: VIEWTABLE: Work.Tr_chk_adae								
File Edit View Tools Data Solutions Help								
	USUBJID	USUBJID_L	AEACN	AEACN_L	AEBODSYS	AEBODSYS_L	AELLT	AELLT_L
1	123456.987654	13	DOSE NOT CHANGED	16	Gastrointestinal disorders	26	Feeling queas	13

SCOPE OF MACRO

- › The variables in the source/derived datasets are not compared programmatically because:
 - › Different variables in a source dataset can have the same value as the derived character variable.
 - › Multiple source datasets can have the same value.
 - › The character variables may not be a direct copy but manipulated via concatenation, case change, etc.

SUMMARY

- › The truncate check macro can be used to simplify and speed up the QC process of checking truncated values in the dataset codes.
- › This macro script makes it easier to spot character variable truncations by generating single record datasets with the character variables and their maximum length.

CONTACT

Please forward your questions and comments to

Proma Mukherjee

Parexel International

Bengaluru

India

proma.mukherjee@parexel.com



Thank you