

2 PROC TRANSPOSEs = 1 DATA Step DOW-Loop

Nancy Brucken, i3 Statprobe, Ann Arbor, MI

ABSTRACT

A limitation of PROC TRANSPOSE is that multiple variables in the VAR statements lead to one record per variable in the output dataset. Sometimes, this behavior is desired, but often we need a single record per BY statement as output. This paper demonstrates the use of the DOW-Loop to transpose the data properly in a single pass through the input dataset.

PROBLEM

Suppose we have a SAS® dataset of summarized laboratory data, containing one record per parameter, category and treatment group. We need to report that data in a table with one record per parameter and category, and the treatment group counts displayed in separate columns. The input dataset looks like this:

ANALYTEC	LISTTYPE	LISTNUM	RXGRP	PATCT	DENOM
CALCIUM	Abnormal Baseline Values	1	1	6	260
CALCIUM	Abnormal Baseline Values	1	2	4	227
CALCIUM	Normal Baseline and Abnormal EOT Values	2	1	5	260
CALCIUM	Normal Baseline and Abnormal EOT Values	2	2	6	227

We need to transpose this data to a reporting dataset looking like this:

ANALYTEC	LISTTYPE	LISTNUM	RXGRP	PATCT1	PATCT2	DENOM1	DENOM2
CALCIUM	Abnormal Baseline Values	1	1	6	4	260	227
CALCIUM	Normal Baseline and Abnormal EOT Values	2	2	5	6	260	227

One possible solution is to use PROC TRANSPOSE to transpose the dataset by parameter and category. However, the following code, transposing both variables in the same step, yields the dataset shown below, which is not quite what we are looking for.

```

PROC TRANSPOSE DATA=LABS OUT=LABTRAN;
  BY ANALYTEC LISTTYPE LISTNUM;
  VAR PATCT DENOM;
  ID RXGRP;
RUN;

```

yields:

ANALYTEC	LISTTYPE	LISTNUM	NAME	_1	_2
CALCIUM	Abnormal Baseline Values	1	PATCT	6	4
CALCIUM	Abnormal Baseline Values	1	DENOM	260	227
CALCIUM	Normal Baseline and Abnormal EOT Values	2	PATCT	5	6
CALCIUM	Normal Baseline and Abnormal EOT Values	2	DENOM	260	227

We can then split this dataset apart so that the PATCT and DENOM records end up stored in separate datasets, and merge it back together by parameter and category, in order to reach our target dataset. However, PROC TRANSPOSE requires one pass through the entire dataset, splitting the resulting dataset apart requires another pass, and the subsequent merge makes still another pass through the dataset. For small datasets, the three passes through the data may not take much time or system resources. However, for larger datasets, each pass may take a significant amount of execution time and/or resources.

ALTERNATIVE SOLUTION

An alternative solution involves a single pass through the dataset using a technique commonly known as the Dorfman-Whitlock DO-Loop (DOW-Loop). This powerful technique moves the DATA step SET statement inside of a explicitly-coded DO-loop, thus giving the programmer complete control over retention of variable values and the population of the Program Data Vector (PDV).

The code to generate the desired reporting dataset directly from the input dataset looks like:

```

data labtran (drop=i patct denom rxgrp);
  array patcts (*) patct1-patct2;
  array denoms (*) denom1-denom2;

  *** Initialize arrays;
  do i=1 to dim(patcts);
    patcts(i) = 0;
    denoms(i) = 0;
  end;

  do until (last.listnum or eof);
    set labs end=eof;
    by analytec listnum;

    *** Populate arrays;
    patcts(rxgrp) = patct;
    denoms(rxgrp) = denom;
  end;

  output;
run;

```

The ARRAY statements define two arrays, one in which to store the patient subgroup counts for each treatment group, and one in which to store the total number of patients in each treatment group. The array initialization loop is executed every time control returns to the top of the DATA step, after the last record for a category has been output.

The DOW-Loop itself begins with the DO UNTIL statement, and takes control from the traditional implicit DATA step loop. Because the SET statement is inside of the DOW-loop, and the loop is not exited until after the last record for the category has been processed, the array elements populated inside of the loop are retained while all of the records for the category are read in. Thus, we can store values in PATCT1, PATCT2, DENOM1 and DENOM2 for all of the records in the category during the execution of the DOW-loop, without having them reinitialized at the top of the DATA step. The single record for the category, containing the filled arrays, is output at the completion of the DOW-loop. Control then returns to the top of the DATA step, the arrays are reinitialized, and the records for the next category are read in.

CONCLUSION

The DOW-Loop is an extremely powerful programming technique, which gives the programmer more control over the input and retention of variables. It can reduce the number of passes through a dataset required by a program, and fewer passes through the dataset generally result in faster and more efficient programs.

REFERENCES

Dorfman, Paul. (2008) "The DOW-Loop Unrolled". PharmaSUG 2008 Conference Proceedings, housed at <http://www.lexjansen.com>.

Chakravarthy, Venky, (2005). "RETAIN or NOT? Is LAG Far Behind?". PharmaSUG 2005 Conference Proceedings, housed at <http://www.lexjansen.com>.

For more information on applications of the DOW-Loop, visit the SAS-L listserv archives at <http://www.listserv.uga.edu/archives/sas-l.html>, and search for postings by Ian Whitlock and Paul Dorfman, among others.

ACKNOWLEDGMENTS

Thank you to Paul Slagle and Nancy Fish for reviewing this paper for me, to all of the people who gave me the inspiration for using this technique through their postings on SAS-L, and to Don Henderson, who, so far as anyone can determine, was probably the first person to try it.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Nancy Brucken
i3 Statprobe
5430 Data Court, Suite 200
Ann Arbor, MI 48108

Work Phone: (734) 757-9568
E-mail: Nancy.Brucken@i3statprobe.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.