

Proc Transpose

Steven L. Wonder, Trilogy Consulting Corporation, Chicago, IL, USA

Abstract

Frequently the need arises in SAS® programming for interchanging columns and rows of data sets. A classic example of this occurs during the processing of normalized data. Modern relational databases store data by primary and foreign keys in a tabular format. However, the presentation and processing of such data often requires re-organization of the data where the keys become columns and the data itself are tabulated within the columns. While Proc Tabulate can yield such a presentation of normalized data, it is not able to output a data set for further analysis. This paper presents a method of processing normalized data that achieves the same result as Proc Tabulate with Proc SQL and Proc Transpose.

Introduction

Proc Transpose changes observations into variables and variables into observations. The application of this procedure is especially useful where data values are stored by identifying or descriptive characteristic, but final results are summarized in columns determined by these identifying characteristics.

This paper describes a non-procedural technique using Proc Transpose and Proc SQL for regeneration of original tabular data from a normalized form.

Procedure Syntax

```
PROC TRANSPOSE <option-list>;
  VAR variable-list;
  ID variable;
  IDLABEL variable;
  COPY variable-list;
  BY variable-list;
```

Table 1 Options

Option	Function
DATA=	Specifies the name of data set to transpose
OUT=	Specifies the name of the resulting data set
PREFIX=	Gives the name of the prefix to use for column names
NAME=	Identifies the name of variable in output data set holding identify of transposed variable
LABEL=	Identifies the name of variable in output data set holding the label of variable transposed
LET	Duplicate values of ID variable in input data set allowed

BY Statement - may be used to obtain separate observations for the variable(s) transposed.

Copy Statement - may be used to copy the value of variable(s) from the input file to the output file.

ID Statement - may be used as a formatted source of name(s) of transposed variable(s) in the output file.

IDLABEL Statement - may be used to provide labels for the transposed variable(s).

VAR Statement - lists the variable(s) to transpose.

Application of Proc Transpose

As an example consider hospital patient data that contains among other things a master identification key along with total charges and six boolean values that indicate the presence or absence of risk pool. The data may be logically considered to be structured as follows

Record Id Nbr...Charges... Major Surgery, Minor Surgery, Cardiac, Endoscopy, Medical, All Patients...

Each Risk Pool flag is a character value of either zero, one, or missing. For a single patient six full columns are used to store information about risk pools even if none are applicable. For a large database of one million or more records, this adds up to a significant amount of disk space. Thus it makes sense to store only those risk pools that are not zero or missing. A normalized table consisting of record key and a flag value was chosen to store risk pool data. Table 1 contains the six risk pools and values assigned to each. Table 2 demonstrates an example of the normalized risk pool data. For Record ID 00001 there are three risk pools: minor surgery, cardiac, and medical.

Table 2 Risk Pool Flags

Flag Name	Flag Value
major surgery	1
minor surgery	2
cardiac	3
endoscopy	4
medical	5
all patients	6

Table 3 Normalized Risk Pool Data

Record Id Key	Value
00001	2
00001	3
00001	5
00002	5
00002	6
00003	1
00003	4
00003	5
00004	3
00005	1
00005	6

For human interpretation of this data, a structure much like that of the original data is useful. To convert normalized data into a "wide tabular" structure is needed. The following SAS® code was used to transform the data.

Program Code

```

data rpd2 (keep=rpflag)
  rpfll (keep=rpflag rpname);
length rpname $ 12;
rpflag=1; rpname='major surgery'; output;
rpflag=2; rpname='minor surgery'; output;
rpflag=3; rpname='cardiac'; output;
rpflag=4; rpname='endoscopy'; output;
rpflag=5; rpname='medical'; output;
rpflag=6; rpname='all patients'; output;
run;

proc sort data=rpd2 nodup;
  by rpflag;
run;

data rpfmt (keep=start end label ffmtname);
  retain ffmtname "rpfmt";
  set rpfll;
  start=rpflag; end=rpflag; label=rpname;
run;

proc format cntlin=rpfmt;
run;

data rpd1 (keep=mrn)
  rpd2 (keep=mrn rpflag rpvalue);
  retain rpvalue 1;
  infile cards;
  input mrn rpflag;
  list;
cards;
00001 2
00001 3
00001 5
00002 5
00002 6
00003 1
00003 4
00003 5
00004 3
00005 1
00005 6
;
run;

proc sort data=rpd1 nodup;
  by mrn;
run;

proc sort data=rpd2;
  by mrn;
run;

/* generate cartesian product, then merge back with actual data */
proc sql;
  create table rp2 as
  select rp1.mrn, rp1.rpflag,
  sum(rp1.rpvalue, rpd2.rpvalue) as value
  from (select rpd1.mrn, rpd2.rpflag, 0 as rpvalue
  from rpd1, rpd2) rp1 left join rpd2
  on rp1.mrn=rpd2.mrn and rp1.rpflag=rpd2.rpflag;
quit;

proc transpose data=rp2 out=rp3;
  by mrn;
  var value;
  id rpflag;
  format rpflag rpfmt.;
run;

proc print data=rp3 noobs;
  title2 "Risk Pools";
run;

```

Results obtained from de-normalizing the risk pool data follow:

MRN	NAME	MAJOR_SU	MINOR_SU	Risk Pools			
				CARDIAC	ENDOC SOP	MEDICAL	ALL PATI
1	VALUE	0	1	1	0	1	0
2	VALUE	0	0	0	0	1	1
3	VALUE	1	0	0	1	1	0
4	VALUE	0	0	1	0	0	0
5	VALUE	1	0	0	0	0	1

Conclusions

Proc Transpose is a highly useful non-procedural way of converting normalized data into a form that is more readily useful to an analyst and possibly more useful for some computations. These techniques primarily rely on non-procedural methods that can be used by any programmer familiar with SQL and Proc Transpose.

Because the processing technique described here produces a full cartesian product, it's best application exists against small- and medium-sized files.

SAS is a registered trademark of SAS Institute Inc. in the USA and other countries. © indicates USA registration.

References

SAS Institute Inc. (1990), SAS Procedures Guide, Third Edition, Cary NC: SAS Institute Inc.

Author Contact

Steven L. Wonder
Trilogy Consulting Corporation
850 South Green Bay Road
Waukegan, IL 60085