Transposing Data Sets

The TRANSPOSE procedure is used to transpose data sets from one form to another. The TRANSPOSE procedure can transpose variables and observations, or transpose variables and observations within BY groups. This section discusses some applications of the TRANSPOSE procedure relevant to time series data sets. Refer to the SAS Procedures Guide for more information on PROC TRANSPOSE.

Transposing from Interleaved to Standard Time Series Form

The following statements transpose part of the interleaved form output data set FOREOUT, produced by PROC FORECAST in a previous example, to a standard form time series data set. To reduce the volume of output produced by the example, a WHERE statement is used to subset the input data set.

Observations with _TYPE_=ACTUAL are stored in the new variable ACTUAL; observations with _TYPE_=FORECAST are stored in the new variable FORECAST; and so forth. Note that the method used in this example only works for a single variable.

```
title "Original Data Set";
proc print data=foreout;
   where date > '1may1991'd & date < '1oct1991'd;
run;

proc transpose data=foreout out=trans(drop=_name_ _label_);
   var cpi;
   id _type_;
   by date;
   where date > '1may1991'd & date < '1oct1991'd;
run;

title "Transposed Data Set";
proc print data=trans;
run;
```

The TRANSPOSE procedure adds the variables _NAME_ and _LABEL_ to the output data set. These variables contain the names and labels of the variables that were transposed. In this example, there is only one transposed variable, so _NAME_ has the value CPI for all observations. Thus, _NAME_ and _LABEL_ are of no interest and are dropped from the output data set using the DROP= data set option. (If none of the variables transposed have a label, PROC TRANSPOSE does not output the _LABEL_ variable and the DROP=_LABEL_ option produces a warning message. You can ignore this message, or you can prevent the message by omitting _LABEL_ from the DROP= list.)

The original and transposed data sets are shown in Figure 2.23. (The observation numbers shown for the original data set reflect the operation of the WHERE statement.)

Original Data Set

Obs	date	_TYPE_	_LEAD_	cpi
37	JUN1991	ACTUAL	0	136.000
38	JUN1991	FORECAST	0	136.146
39	JUN1991	RESIDUAL	0	-0.146
40	JUL1991	ACTUAL	0	136.200
41	JUL1991	FORECAST	0	136.566
42	JUL1991	RESIDUAL	0	-0.366
43	AUG1991	FORECAST	1	136.856
44	AUG1991	L95	1	135.723
45	AUG1991	U95	1	137.990
46	SEP1991	FORECAST	2	137.443
47	SEP1991	L95	2	136.126
48	SEP1991	U95	2	138.761

Transposed Data Set

Obs	date	ACTUAL	FORECAST	RESIDUAL	L95	U95
1	JUN1991	136.0	136.146	-0.14616	.	.
2	JUL1991	136.2	136.566	-0.36635	.	.
3	AUG1991	.	136.856	.	135.723	137.990
4	SEP1991	.	137.443	.	136.126	138.761

Figure 2.23: Original and Transposed Data Sets

Transposing Cross-sectional Dimensions

The following statements transpose the variable CPI in the CPICITY data set shown in a previous example from time series cross-sectional form to a standard form time series data set. (Only a subset of the data shown in the previous example is used here.) Note that the method shown in this example only works for a single variable.

```
   title "Original Data Set";
   proc print data=cpicity;
   run;

   proc sort data=cpicity out=temp;
```

```
      by date city;
   run;

   proc transpose data=temp out=citycpi(drop=_name_ _label_);
      var cpi;
      id city;
      by date;
   run;

   title "Transposed Data Set";
   proc print data=citycpi;
   run;
```

The names of the variables in the transposed data sets are taken from the city names in the ID variable CITY. The original and the transposed data sets are shown in Figure 2.24.

Original Data Set

Obs	city	date	cpi
1	Chicago	JAN90	128.1
2	Chicago	FEB90	129.2
3	Chicago	MAR90	129.5
4	Chicago	APR90	130.4
5	Chicago	MAY90	130.4
6	Chicago	JUN90	131.7
7	Chicago	JUL90	132.0
8	Los Angeles	JAN90	132.1
9	Los Angeles	FEB90	133.6
10	Los Angeles	MAR90	134.5
11	Los Angeles	APR90	134.2
12	Los Angeles	MAY90	134.6
13	Los Angeles	JUN90	135.0
14	Los Angeles	JUL90	135.6
15	New York	JAN90	135.1
16	New York	FEB90	135.3
17	New York	MAR90	136.6
18	New York	APR90	137.3
19	New York	MAY90	137.2
20	New York	JUN90	137.1

Obs	city	date	cpi
21	New York	JUL90	138.4

Transposed Data Set

Obs	date	Chicago	Los_Angeles	New_York
1	JAN90	128.1	132.1	135.1
2	FEB90	129.2	133.6	135.3
3	MAR90	129.5	134.5	136.6
4	APR90	130.4	134.2	137.3
5	MAY90	130.4	134.6	137.2
6	JUN90	131.7	135.0	137.1
7	JUL90	132.0	135.6	138.4

Figure 2.24: Original and Transposed Data Sets

The following statements transpose the CITYCPI data set back to the original form of the CPICITY data set. The variable _NAME_ is added to the data set to tell PROC TRANSPOSE the name of the variable in which to store the observations in the transposed data set. (If the (DROP=_NAME_ _LABEL_) option were omitted from the first PROC TRANSPOSE step, this would not be necessary. PROC TRANSPOSE assumes ID _NAME_ by default.)

The NAME=CITY option in the PROC TRANSPOSE statement causes PROC TRANSPOSE to store the names of the transposed variables in the variable CITY. Because PROC TRANSPOSE recodes the values of the CITY variable to create valid SAS variable names in the transposed data set, the values of the variable CITY in the retransposed data set are not the same as the original. The retransposed data set is shown in Figure 2.25.

```
data temp;
   set citycpi;
   _name_ = 'CPI';
run;

proc transpose data=temp out=retrans name=city;
   by date;
run;

proc sort data=retrans;
   by city date;
run;

title "Retransposed Data Set";
proc print data=retrans;
run;
```

```
               Retransposed Data Set

 Obs    date      city           CPI

   1   JAN90   Chicago        128.1
   2   FEB90   Chicago        129.2
   3   MAR90   Chicago        129.5
   4   APR90   Chicago        130.4
   5   MAY90   Chicago        130.4
   6   JUN90   Chicago        131.7
   7   JUL90   Chicago        132.0
   8   JAN90   Los_Angeles    132.1
   9   FEB90   Los_Angeles    133.6
  10   MAR90   Los_Angeles    134.5
  11   APR90   Los_Angeles    134.2
  12   MAY90   Los_Angeles    134.6
  13   JUN90   Los_Angeles    135.0
  14   JUL90   Los_Angeles    135.6
  15   JAN90   New_York       135.1
  16   FEB90   New_York       135.3
  17   MAR90   New_York       136.6
  18   APR90   New_York       137.3
  19   MAY90   New_York       137.2
  20   JUN90   New_York       137.1
  21   JUL90   New_York       138.4
```

Figure 2.25: Data Set Transposed Back to Original Form

Chapter Contents Previous Next Top