INTRODUCTION

SAS® users frequently find themselves wishing they could "rotate" their SAS data sets transforming observations into variables and vice versa. Restructuring a data set to best suit your needs can be a formidable task involving a great deal of somewhat laborious programming. Fortunately, this does not have to be the case. Data sets can be restructured with minimal programming. Methods of doing so include using DATA step processing or the TRANSPOSE procedure. Examples of how these methods can be applied to sample data are presented.

SAMPLE DATA

The data used in the examples presented here result from a (hypothetical) clinical trial. The objective of this trial was to compare the effects of 2 drugs on each of several symptoms associated with the common cold. There are 3 data sets, TABLE, LIST1, and LIST2, shown at the end of this paper. They will often be referred to by these names.

Data set TABLE contains the variables PATNO, SDAY, DRUG, STHROAT, COUGH, HEADACHE, BODYACHE, NAUSEA, and TEMP corresponding, respectively, to patient number, day of the study, study drug, and the symptoms sore throat, cough, headache, body aches, nausea, and temperature. It consists of 149 observations and 9 variables. The organization of this data set may be thought of as a table with combined information regarding patient number, study day, and drug as rows and each symptom as a column.

This structure may be ideal if a data display showing each patient's symptoms on each study day is to be produced. However, to expeditiously apply SAS procedures to these data, it would be preferable if only one symptom were in each observation. In this case, a data set structured as LIST1 or LIST2 is more desirable. These data sets contain exactly the same information as data set TABLE. They contain the variables PATNO, SDAY, and DRUG, along with the variables SYMPTOM, which identifies the symptom in each observation, and RESPONSE, the corresponding value for that symptom.

Data sets LIST1 and LIST2 differ only with regard to the variable SYMPTOM. In LIST1 this variable contains numeric codes to identify the symptoms while in LIST2 this variable contains character values. Both of these data sets contain 894 observations and 5 variables.

The examples presented here will demonstrate how data set TABLE can be restructured so that data set LIST1 or LIST2 is output, and vice versa. In other words, the variables in TABLE will be transformed to observations as LIST1 or LIST2 (table to list). The observations in LIST1 and LIST2 will be transformed to variables and output as TABLE (list to table).

DATA STEP PROCESSING

One of the primary uses of the DATA step is to rearrange or manipulate a data set so that it can be used with a SAS procedure. Many variables may require manipulation in order to restructure the data set. This can be accomplished in the DATA step with several programming statements which process each of the variables in the same manner. However, this approach can become extremely tedious, particularly when many such statements are necessary.

Since the same process is to be applied to each of several variables, this is an ideal situation for array processing. Arrays are used to group variables together so that the same methods can be applied to each variable. Array processing involves grouping variables into an array with an ARRAY statement, repeating the same process for each variable in the array using an iterative DO loop, and indicating which variable is to be processed.
The ARRAY statement names the array, specifies the number of elements in the array, and lists the variables in the array. The syntax is:

```
ARRAY array-name{number-of-elements} list-of-variables;
```

where `array-name` is a SAS name provided by the programmer to temporarily (for the current DATA step only) identify a group of variables. Enclosed in braces is the number of elements in the array `{number-of-elements}`. Each variable to be included in the array is listed on this statement `{list-of-variables}`.

The iterative DO loop begins with a DO statement of the form

```
DO index-variable = 1 TO number-of-elements;
```

and ends with an END statement. The value of the `index-variable` changes each time the DO loop iterates.

To specify the particular variable to be processed, the array name is used along with a subscript to identify the appropriate element in the array. Within the DO loop, a statement containing `array-name{index-variable}` is used to process each variable in the `list-of-variables`. With the first iteration of the DO loop the index variable is equal to 1 and the first variable is processed, the second iteration will process the second variable in the list, and so on.

**Restructuring Variables to Observations**

**Problem:** Transform TABLE into UST1.

**Solution 1:**

```
DATA LIST1;
SET TABLE;
SYMPTOM = 1;
RESPONSE = STHROAT;
OUTPUT;
SYMPTOM = 2;
RESPONSE = COUGH;
OUTPUT;
KEEP PATNO SDAY DRUG SYMPTOM RESPONSE;
```

The same pattern of action is being applied to each of the symptom variables on data set TABLE. That is, the variable SYMPTOM is assigned a value which can be used to identify the symptom, the variable RESPONSE is assigned the value of the input data set symptom variable, and these two new variables are output. Only the variables needed are kept in the new data set, LIST1.

Note that in the example only 6 symptom variables are actually being transformed. Consider, however, how cumbersome this data step could become with a much larger number of variables to transform.

The solution which follows shows how array processing can be used to obtain the same result.

**Solution 2:**

```
DATA LIST1;
SET TABLE;
ARRAY SYMPT{6} STHROAT COUGH HEADACHE BODYACHE NAUSEA TEMP;
DO i = 1 TO 6;
SYMPTOM = i;
RESPONSE = SYMPT{i};
OUTPUT;
END;
KEEP PATNO SDAY DRUG SYMPTOM RESPONSE;
```

The name of the array is SYMPT which contains the 6 symptom variables, as listed on the ARRAY statement. The DO loop will iterate 6 times. On the first iteration the index variable I is equal to 1. The value 1 will be assigned to the variable SYMPTOM. When \( i = 1 \) `SYMPT{i}` corresponds to the first element of the variable list in the ARRAY statement, STHROAT. Thus, the value of STHROAT for the current observation is assigned to the new variable RESPONSE. During the second iteration `SYMPT{1}` will refer to the second variable in the variable list, COUGH, and so on.

Decodes for the variable SYMPTOM can be provided using PROC FORMAT. Alternatively, a character variable containing the name of the symptom variable from the input data set could have been produced. This was not done here in the interest keeping the example uncomplicated to focus on the problem at hand.

**Restructuring Observations to Variables**

**Problem:** Transform UST1 into TABLE.

Data set LIST1 must first be sorted by PATNO and SDAY. The symptom variables STHROAT, COUGH,
HEADACHE, BODYACHE, NAUSEA, and TEMP are to be created. A RETAIN statement will be required to retain the value of each of these symptom variables. When values for all of these variables have been assigned, the observation should be output. The variables SYMPTOM and RESPONSE, no longer required, are dropped from the new data set.

Solution 1 shows an approach to this problem without using an ARRAY statement.

Solution 1:

```
DATA TABLE;
SET LIST1;
BY PATNO SDAY;
RETAIN STHROAT COUGH HEADACHE BODYACHE NAUSEA TEMP;
IF SYMPTOM = 1 THEN STHROAT = RESPONSE;
IF SYMPTOM = 2 THEN COUGH = RESPONSE;
IF LAST.SDAY THEN OUTPUT;
DROP SYMPTOM RESPONSE;
```

Six statements, one for each new variable being created, were necessary to assign the value of the variable RESPONSE to the appropriate symptom variable. Again, imagine how unwieldy this data step could become if a very large number of variables were being transformed.

If an array statement were used, these 6 statements required in Solution 1 could be replaced by the ARRAY statement and one assignment statement as shown in Solution 2.

Solution 2:

```
DATA TABLE;
SET LIST1;
BY PATNO SDAY;
RETAIN STHROAT COUGH HEADACHE BODYACHE NAUSEA TEMP;
ARRAY SYMPT{6} STHROAT COUGH HEADACHE BODYACHE NAUSEA TEMP;
SYMPT{SYMPTOM} = RESPONSE;
IF LAST.SDAY THEN OUTPUT;
DROP SYMPTOM RESPONSE;
```

In the statement `SYMPT{SYMPTOM} = RESPONSE;` the variable SYMPTOM from the input data set, LIST1, is also the index variable. If the value of the variable SYMPTOM is 1 in the current observation of LIST1, `SYMPT{SYMPTOM}` corresponds to STHROAT, the first element of the variable list. STHROAT is thus assigned the value of the variable RESPONSE in the current observation of LIST1. Similarly, when the variable SYMPTOM is equal to 2 on the current observation of LIST1, `SYMPT{SYMPTOM}` corresponds to the second element of the variable list, COUGH, which is assigned the value of the variable RESPONSE.

Note that while the same process was used to create each of the 6 new variables, this process only occurred once for each observation. An iterative DO loop would have served no purpose here.

**THE TRANSPOSE PROCEDURE**

The TRANSPOSE procedure exists specifically for data restructuring purposes. This procedure transposes a data set changing observations into variables and vice versa. The only output from the TRANSPOSE procedure is a new data set in which the rows (observations) of the original data matrix have become columns (variables) and the columns have become rows.

The transposed variables by default are named COL1, COL2,... (COL for column). An additional variable _NAME_ which contains the name of the transposed variable for each observation is also created.

The only statement required with the TRANSPOSE procedure is the PROC TRANSPOSE statement. Among the options available with this statement are DATA=, OUT=, NAME=, and PREFIX=. The DATA= and OUT= options can be used to name the input and output data sets, respectively. The PREFIX= option is used to specify a prefix other than COL for the transposed variables. If a variable name other than _NAME_ is preferred, that name can be specified with the NAME= option.

Some of the other statements available with this procedure are the BY, ID, and VAR statements. The BY and VAR statements are used in the TRANSPOSE procedure in much the same way as they are used in other procedures. When a BY statement is used, an observation for each variable being transposed is generated for each BY group. The variables to be transposed are listed in the VAR statement. If an ID statement is used, the formatted values of the ID variable become variable names for the transposed variables in the output data set.

If names other than COL1, etc. are preferred for the
new variables, other names can be assigned by using either an ID statement or the PREFIX= option on the PROC TRANSPOSE statement. These have already been discussed. If neither an ID statement nor the PREFIX= option is used, an input variable called _NAME_ can be used to provide names of the transposed variables.

Restructuring Variables to Observations

Problem: Transpose the variables in TABLE to observations.

Solution:

PROC TRANSPOSE DATA=TABLE OUT=EX_TRANS;

A sample of the output data set, EX_TRANS, is shown at the end of this paper. It consists of 9 observations and 150 variables. Recall that data set TABLE has 149 observations and 9 variables. Since the observations have been transformed into variables the new data set was expected to have 9 variables. Note, however, that the 149 observations from the input data set have now become 150 variables. The additional variable is _NAME_ which contains the name of the transposed variables.

Restructuring Observations to Variables

Problem: Transform TABLE into LIST2.

Solution:

PROC TRANSPOSE DATA = TABLE OUT = LIST2
  NAME = SYMPTOM;
  BY PATNO SDAY DRUG;
  VAR STHROAT COUGH HEADACHE BODYACHE NAUSEA TEMP;

The DATA step preceding the PROC TRANSPOSE step assigns the variable name RESPONSE to contain the values of the transposed variables. Otherwise, this variable name would be COL1. The NAME= option specifies that the variable which contains the names of the transposed variables be named SYMPTOM instead of _NAME_. The VAR statement lists the variables to be transposed.

As noted earlier, the only difference between the LIST2 data set generated by the TRANSPOSE procedure and LIST1, as generated by the data step, is that in LIST2 the variable SYMPTOM contains the previous variable name for each symptom while in LIST1 this variable contains a numeric code.

Restructuring Observations to Variables

Problem: Transform LIST2 into TABLE.

Solution:

PROC TRANSPOSE DATA = LIST2
  OUT = TABLE DROP = _NAME_;
  BY PATNO SDAY DRUG;
  ID SYMPTOM;
  VAR RESPONSE;

This transpose procedure creates the symptom variables from the input data set, LIST2, and outputs these variables to the output data set, TABLE. Only one variable, RESPONSE, is being transposed. Therefore, the value of the variable _NAME_ in the output data set is RESPONSE in every observation. Because this variable adds no information, it is dropped as the output data set is created.

As a result of the ID statement the names of the newly created symptom variables are determined by the value of the variable SYMPTOM in the input data set. If this statement were omitted, these new variables would be named COL1, COL2, ... COL6 as discussed earlier.

Data set LIST1 can also be restructured via the TRANSPOSE procedure to produce an output data set identical to TABLE. The FORMAT procedure is first invoked to output a format, SYMPTOMF as...
follows:

PROC FORMAT;
  RESPONSE SYMPTOMF
  1 = 'STHROAT'
  2 = 'COUGH'
  3 = 'HEADACHE'
  4 = 'BODYACHE'
  5 = 'NAUSEA'
  6 = 'TEMP'
;

Having output this format, a TRANSPOSE procedure step similar to that shown when LIST1 was transformed will restructure data set LIST1 into data set TABLE if the FORMAT statement

FORMAT SYMPTOM SYMPTOMF.;

is included.

CONCLUSION

Transforming a data set to make programming more expedient does not have to be a time-consuming, monotonous chore. Use of arrays in the DATA step or the TRANSPOSE procedure can accomplish this task with relatively few programming statements. This paper provided some examples of how data sets can be restructured using these techniques. It is not, by any means, all encompassing. A great deal of additional information concerning arrays and the TRANSPOSE procedure is available in the appropriate SAS manuals.

SAS is a registered trademark of SAS Institute Inc, Cary NC, USA.
### Sample Data Sets

#### DATA SET TABLE

<table>
<thead>
<tr>
<th>OBS</th>
<th>PATNO</th>
<th>SDAY</th>
<th>DRUG</th>
<th>STHROAT</th>
<th>COUGH</th>
<th>HEADACHE</th>
<th>BODYACHE</th>
<th>NAUSEA</th>
<th>TEMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>2</td>
<td>4</td>
<td>7</td>
<td>0</td>
<td>38.0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>0</td>
<td>37.8</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>37.6</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>37.0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>37.0</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>37.0</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>36.8</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>36.4</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>36.7</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>37.0</td>
</tr>
</tbody>
</table>

#### DATA SET LIST1

<table>
<thead>
<tr>
<th>OBS</th>
<th>PATNO</th>
<th>SDAY</th>
<th>DRUG</th>
<th>SYMPTOM</th>
<th>RESPONSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>6.0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2.0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>4.0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>7.0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>0.0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>38.0</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>5.0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1.0</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3.0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>6.0</td>
</tr>
</tbody>
</table>

#### DATA SET LIST2

<table>
<thead>
<tr>
<th>OBS</th>
<th>PATNO</th>
<th>SDAY</th>
<th>DRUG</th>
<th>SYMPTOM</th>
<th>RESPONSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>STHROAT</td>
<td>6.0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>COUGH</td>
<td>2.0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>HEADACHE</td>
<td>4.0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>BODYACHE</td>
<td>7.0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>NAUSEA</td>
<td>0.0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>TEMP</td>
<td>38.0</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>STHROAT</td>
<td>5.0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>COUGH</td>
<td>1.0</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>HEADACHE</td>
<td>3.0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>BODYACHE</td>
<td>6.0</td>
</tr>
</tbody>
</table>

#### DATA SET EX TRANS

<table>
<thead>
<tr>
<th>OBS <em>NAME</em></th>
<th>COL1</th>
<th>COL2</th>
<th>COL3</th>
<th>COL4</th>
<th>COL5</th>
<th>COL6</th>
<th>COL7</th>
<th>COL8</th>
<th>COL9</th>
<th>COL10</th>
<th>COL11</th>
<th>COL12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 PATNO</td>
<td>1</td>
<td>1.0</td>
<td>1.0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>2</td>
<td>3.0</td>
</tr>
<tr>
<td>2 SDAY</td>
<td>1</td>
<td>2.0</td>
<td>3.0</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>2.0</td>
<td>3.0</td>
<td>4.0</td>
<td>5</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3 DRUG</td>
<td>1</td>
<td>1.0</td>
<td>1.0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1</td>
<td>2.0</td>
<td>2</td>
</tr>
<tr>
<td>4 STHROAT</td>
<td>6</td>
<td>5.0</td>
<td>1.0</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>1.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0</td>
<td>8.0</td>
<td>5</td>
</tr>
<tr>
<td>5 COUGH</td>
<td>2</td>
<td>1.0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>6 HEADACHE</td>
<td>4</td>
<td>3.0</td>
<td>1.0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>7 BODYACHE</td>
<td>7</td>
<td>6.0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td>6.0</td>
<td>2</td>
</tr>
<tr>
<td>8 NAUSEA</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>9 TEMP</td>
<td>38</td>
<td>37.8</td>
<td>37.6</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>38.8</td>
<td>36.4</td>
<td>36.7</td>
<td>37</td>
<td>37.1</td>
<td>37</td>
</tr>
</tbody>
</table>