

Chapter 10

Useful SAS

Procedures

Arthur Li

Restructuring Datasets Using the TRANSPOSE Procedure

```
PROC TRANSPOSE <DATA=input-data-set>  
                <DELIMITER=delimiter>  
                <LABEL=label>  
                <LET>  
                <NAME=name>  
                <OUT=output-data-set>  
                <PREFIX=prefix>  
                <SUFFIX=suffix>;  
  
BY <DESCENDING> variable-1  
    <...<DESCENDING> variable-n>;  
COPY variable(s);  
ID variable;  
IDLABEL variable;  
VAR variable(s);  
RUN;
```

Transposing an Entire Data Set

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
data dat1;  
    input s_name $ s_id $ e1-e3;  
    label e1 = English1  
          e2 = English2  
          e3 = English3;  
    datalines;  
John A01 89 90 92  
Mary A02 92 . 81  
;
```

❖ Read in data

- ❑ S_Name & S_ID: character variables
- ❑ E1 – E3: numeric variables
- ❑ E1 – E3: have variable labels

Transposing an Entire Data Set

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc transpose data=dat1  
               out=dat1_out1;  
run;  
  
proc print data=dat1_out1;  
run;
```

- ❖ Only the PROC TRANSPOSE statement is used
- ❖ OUT=: specifies the name of the transposed data set

Transposing an Entire Data Set

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc transpose data=dat1
               out=dat1_out1;
run;

proc print data=dat1_out1;
run;
```

Obs	_NAME_	_LABEL_	COL1	COL2
1	e1	English1	89	92
2	e2	English2	90	.
3	e3	English3	92	81

Transposing an Entire Data Set

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc transpose data=dat1  
               out=dat1_out1;  
run;  
  
proc print data=dat1_out1;  
run;
```

By default, without specifying the names of the transposing variables, all the numeric variables from the input data set are transposed.

COL1	COL2
89	92
90	.
92	81

Transposing an Entire Data Set

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc transpose data=dat1  
               out=dat1_out1;  
run;  
  
proc print data=dat1_out1;  
run;
```

**Default variable
names**

Obs	_NAME_	_LABEL_	COL1	COL2
1	e1	English1	89	92
2	e2	English2	90	.
3	e3	English3	92	81

Transposing an Entire Data Set

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc transpose data=dat1
               out=dat1_out1;
run;

proc print data=dat1_out1;
run;
```

Obs	_NAME_	_LABEL_	COL1	COL2
1	e1	English1	89	92
2	e2	English2	90	.
3	e3	English3	92	81

Transposing an Entire Data Set

Contain labels

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc transpose data=dat1
               out=dat1_out1;
run;

proc print data=dat1_out1;
run;
```

Obs	_NAME_	_LABEL_	COL1	COL2
1	e1	English1	89	92
2	e2	English2	90	.
3	e3	English3	92	81

Transposing an Entire Data Set

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc transpose data=dat1
               out=dat1_out1;
run;

proc print data=dat1_out1;
run;
```

Obs	_NAME_	_LABEL_	COL1	COL2
1	e1	English1	89	92
2	e2	English2	90	.
3	e3	English3	92	81

How to control the names of these variables?

Transposing an Entire Data Set

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc transpose data=dat1
                out=dat1_out1
                name=varname
                label=labelname
                prefix=score_;
run;
```

Obs	varname	labelname	score_1	score_2
1	e1	English1	89	92
2	e2	English2	90	.
3	e3	English3	92	81

Transposing an Entire Data Set

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc transpose data=dat1
               out=dat1_out1
               name=varname
               label=labelname
               prefix=score_;
run;
```

Obs	varname	labelname	score_1	score_2
1	e1	English1	89	92
2	e2	English2	90	.
3	e3	English3	92	81

Transposing an Entire Data Set

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc transpose data=dat1
               out=dat1_out1
               name=varname
               label=labelname
               prefix=score_;
run;
```

Obs	varname	labelname	score_1	score_2
1	e1	English1	89	92
2	e2	English2	90	.
3	e3	English3	92	81

**SUFFIX= option:
attach a suffix**

Transposing an Entire Data Set

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc transpose data=dat1
                out=dat1_out1
                name=varname
                label=labelname
                prefix=score_;
```

run

John's
score

Mary's
score

Obs	varname	labelname	score_1	score_2
			89	92
			90	.
			92	81

How can you change
the variable names to
score_John and
score_Mary?

Transposing an Entire Data Set

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc transpose data=dat1
                out=dat1_out2
                label=labelname
                name=varname
                prefix=score_;

    var e1-e3;
    id s_name;
run;
```

Obs	varname	labelname	score_ John	score_ Mary
1	e1	English1	89	92
2	e2	English2	90	.
3	e3	English3	92	81

Transposing an Entire Data Set

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc transpose data=dat1
                out=dat1_out2
                label=labelname
                name=varname
                prefix=score_;

    var e1-e3;
    id s_name;
run;
```

Obs	varname	labelname	score_ John	score_ Mary
1	e1	English1	89	92
2	e2	English2	90	.
3	e3	English3	92	81

Transposing an Entire Data Set

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc transpose data=dat1
                out=dat1_out2
                label=labelname
                name=varname
                delim=_;

    var e1-e3;
    id s_name s_id;
    idlabel s_id;

run;
```

Obs	varname	labelname	John_A01	Mary_A02
1	e1	English1	89	92
2	e2	English2	90	.
3	e3	English3	92	81

Transposing an Entire Data Set

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc transpose data=dat1
                out=dat1_out2
                label=labelname
                name=varname
                delim=_;

    var e1-e3;
    id s_name s_id;
    idlabel s_id;
run;
```

Obs	varname	labelname	John_A01	Mary_A02
1	e1	English1	89	92
2	e2	English2	90	.
3	e3	English3	92	81

Transposing an Entire Data Set

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc contents data=dat1_out2;  
run;
```

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Label
3	John_A01	Num	8	A01
4	Mary_A02	Num	8	A02
2	labelname	Char	40	LABEL OF FORMER VARIABLE
1	varname	Char	8	NAME OF FORMER VARIABLE

Introduction to Transposing BY-Groups

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc sort data=dat1
          out=dat1_sort;
      by s_name;
run;

proc transpose data=dat1_sort
              out=dat1_out3;
      by s_name;
run;
```

By-variable:
S_Name

Introduction to Transposing BY-Groups

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc sort data=dat1
          out=dat1_sort;
          by s_name;
run;

proc transpose data=dat1_sort
              out=dat1_out3;
              by s_name;
run;
```

Obs	s_name	_NAME_	_LABEL_	COL1
1	John	e1	English1	89
2	John	e2	English2	90
3	John	e3	English3	92
4	Mary	e1	English1	92
5	Mary	e2	English2	.
6	Mary	e3	English3	81

Introduction to Transposing BY-Groups

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc sort data=dat1
        out=dat1_sort;
    by s_name;
run;
```

```
proc transpose data=dat1_sort
        out=dat1_out3;
    by s_name;
run;
```

Obs	s_name	_NAME_	_LABEL_	COL1
1	John	e1	English1	89
2	John	e2	English2	90
3	John	e3	English3	92
4	Mary	e1	English1	92
5	Mary	e2	English2	.
6	Mary	e3	English3	81

Introduction to Transposing BY-Groups

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc sort data=dat1
      out=dat1_sort;
      by s_name;
run;

proc transpose data=dat1_sort
      out=dat1_out3;
      by s_name;
run;
```

Obs	s_name	_NAME_	_LABEL_	COL1
1	John	e1	English1	89
2	John	e2	English2	90
3	John	e3	English3	92
4	Mary	e1	English1	92
5	Mary	e2	English2	.
6	Mary	e3	English3	81

Introduction to Transposing BY-Groups

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc transpose data=dat1_sort
                out=dat1_out3
                name=varname
                label=TEST;
    by s_name;
    copy s_id;
run;
```

Obs	s_name	s_id	varname	TEST	COL1
1	John	A01	e1	English1	89
2	John		e2	English2	90
3	John		e3	English3	92
4	Mary	A02	e1	English1	92
5	Mary		e2	English2	.
6	Mary		e3	English3	81

Introduction to Transposing BY-Groups

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc transpose data=dat1_sort
                out=dat1_out3
                name=varname
                label=TEST;

    by s_name;
    copy s_id;

run;
```

Obs	s_name	s_id	varname	TEST	COL1
1	John	A01	e1	Er	
2	John		e2	Er	
3	John		e3	Er	
4	Mary	A02	e1	Er	
5	Mary		e2	Er	
6	Mary		e3	Er	

COPY statement:
copy variable(s)
from the input
data set directly
to the transposed
data set

Introduction to Transposing BY-Groups

**N = 2 from
the input data**

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc transpose data=dat1_sort  
                out=dat1_out3  
                name=varname  
                label=TEST;  
    by s_name;  
    copy s_id;  
run;
```

Obs	s_name	s_id	varname	TEST	COL1
1	John	A01	e1	English1	89
			e2	English2	90
			e3	English3	92
		A02	e1	English1	92
			e2	English2	.
			e3	English3	81

**number of
observations
are copied = 2**

Introduction to Transposing BY-Groups

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc transpose data=dat1_sort  
                out=dat1_out3  
                name=varname  
                label=TEST;  
    by s_name;  
    copy s_id;  
run;
```

ID statement:
used to specify
the variable from
the input data set
that contains the
values to rename
the transposed
variables.

	s_id	varname	TEST	COL1
e	A01	e1	English1	89
		e2	English2	90
		e3	English3	92
e	A02	e1	English1	92
		e2	English2	.
		e3	English3	81

Where the ID Statement Does Not Work for Transposing BY-Groups

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc transpose data=dat1_sort
                out=dat1_out4
                name=varname
                label=TEST;

    by s_name;
    id s_id;

run;
```

Obs	s_name	varname	TEST	A01	A02
1	John	e1	English1	89	.
2	John	e2	English2	90	.
3	John	e3	English3	92	.
4	Mary	e1	English1	.	92
5	Mary	e2	English2	.	.
6	Mary	e3	English3	.	81

Where the ID Statement Does Not Work for Transposing BY-Groups

Dat1:

	S_NAME	S_ID	E1	E2	E3
1	John	A01	89	90	92
2	Mary	A02	92	.	81

```
proc transpose data=dat1_sort
               out=dat1_out4
               name=varname
               label=TEST;

  by s_name;
  id s_id;

run;
```

Obs	s name	varname	TEST	A01	A02
e1	English1			89	.
e2	English2			90	.
e3	English3			92	.
e1	English1			.	92
e2	English2			.	.
e3	English3			.	81

You are using the ID variable (contains two values) to name the transposed variable that was supposed to occupy only one column.

Where the ID Statement is Essential for Transposing BY-Groups

Dat2:

	S_NAME	S_ID	Exam	Score
1	John	A01	1	89
2	John	A01	2	90
3	John	A01	3	92
4	Mary	A02	1	92
5	Mary	A02	3	81

```
proc sort data=dat2
        out=dat2_sort;
        by s_name;
run;
proc transpose data=dat2_sort
        out=dat2_out1;
        var score;
        by s_name;
run;
```

**By-variable:
S_Name**

Obs	s_name	_NAME_	COL1	COL2	COL3
1	John	score	89	90	92
2	MARY	SCORE	92	81	.

Where the ID Statement is Essential for Transposing BY-Groups

Dat2:

1

	S_NAME	S_ID	Exam	Score
1	John	A01	1	89
2	John	A01	2	90
3	John	A01	3	92
4	Mary	A02	1	92
5	Mary	A02	3	81

```
proc sort data=dat2
        out=dat2_sort;
        by s_name;
run;
proc transpose data=dat2_sort
        out=dat2_out1;

        var score;
        by s_name;
run;
```

Obs	s_name	_NAME_	COL1	COL2	COL3
1	John	score	89	90	92
2	MARY	SCORE	92	81	.

2

Where the ID Statement is Essential for Transposing BY-Groups

Dat2:

	S_NAME	S_ID	Exam	Score
1	John	A01	1	89
2	John	A01	2	90
3	John	A01	3	92
4	Mary	A02	1	92
5	Mary	A02	3	81

```
proc sort data=dat2
      out=dat2_sort;
      by s_name;
run;
proc transpose data=dat2_sort
      out=dat2_out1;

      var score;
      by s_name;
run;
```

Obs	s_name	_NAME_	COL1	COL2	COL3
1	John	score	89	90	92
2	MARY	SCORE	92	81	.



Where the ID Statement is Essential for Transposing BY-Groups

Dat2:

	S_NAME	S_ID	Exam	Score
1	John	A01	1	89
2	John	A01	2	90
3	John	A01	3	92
4	Mary	A02	1	92
5	Mary	A02	3	81

```
proc transpose data=dat2_sort  
    out=dat2_out2(drop=_name_)  
    prefix=test_;  
    var score;  
    by s_name;  
    id exam;  
run;
```

Obs	s_name	test_1	test_2	test_3
1	John	89	90	92
2	MARY	92	.	81

Handling Duplicated Observations by Using the LET Option

Dat3:

	S_NAME	S_ID	Exam	Score
1	John	A01	1	89
2	John	A01	2	90
3	John	A01	3	92
4	John	A01	3	95
5	Mary	A02	1	92
6	Mary	A02	3	81
7	Mary	A02	3	85

LET option: keep the last occurrence of a particular ID value within either the entire data set or a BY group.

```
proc transpose data=dat3_sort
    out=dat3_out(drop=_name_)
    prefix=test_
    let;
    var score;
    by s_name;
    id exam;
run;
```

Handling Duplicated Observations by Using the LET Option

Dat3:

	S_NAME	S_ID	Exam	Score
1	John	A01	1	89
2	John	A01	2	90
3	John	A01	3	92
4	John	A01	3	95
5	Mary	A02	1	92
6	Mary	A02	3	81
7	Mary	A02	3	85

```
proc sort data=dat3
    out=dat3_sort;
    by s_name exam score;
run;

proc transpose data=dat3_sort
    out=dat3_out(drop=_name_)
    prefix=test
    let;
    var score;
    by s_name;
    id exam;
run;
```

Keep the maximum score within each exam

Obs	s_name	test_1	test_2	test_3
1	John	89	90	95
2	Mary	92	.	85

Handling Duplicated Observations by Using the LET Option

Dat3:

	S_NAME	S_ID	Exam	Score
1	John	A01	1	89
2	John	A01	2	90
3	John	A01	3	92
4	John	A01	3	95
5	Mary	A02	1	92
6	Mary	A02	3	81
7	Mary	A02	3	85

```
proc sort data=dat3
    out=dat3_sort;
    by s_name exam
    descending score;
run;
proc transpose data=dat3_sort
    out=dat3_out(drop=_name_)
    prefix=test
    let;
    var score;
    by s_name;
    id exam;
run;
```

Keep the
minimum
score within
each exam

Obs	s_name	test_1	test_2	test_3
1	John	89	90	92
2	Mary	92	.	81

Situations Requiring Two or More Transpositions

Dat4_Transpose:

	Test_num	John_E	John_M	Mary_E	Mary_M
1	1	89	78	92	76
2	2	90	89	.	91
3	3	92	90	81	89

Dat4:

	S_NAME	E1	E2	E3	M1	M2	M3
1	John	89	90	92	78	89	90
2	Mary	92	.	81	76	91	89

To transpose from
Dat4 → Dat4_Transpose,
we need a “transitional”
data set.

Situations Requiring Two or More Transpositions

Dat4_Transpose:

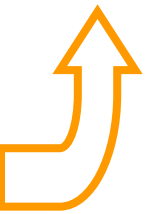
Dat4:

	S_NAME	E1	E2	E3	M1	M2	M3
1	John	89	90	92	78	89	90
2	Mary	92	.	81	76	91	89

	Test_num	John_E	John_M	Mary_E	Mary_M
1	1	89	78	92	76
2	2	90	89	.	91
3	3	92	90	81	89

To transpose from
Dat4 → Dat4_Transpose,
we need a “transitional”
data set.

	S_NAME	Score	Test_num	Class
1	John	89	1	E
2	John	78	1	M
3	Mary	92	1	E
4	Mary	76	1	M
5	John	90	2	E
6	John	89	2	M
7	Mary	.	2	E
8	Mary	91	2	M
9	John	92	3	E
10	John	90	3	M
11	Mary	81	3	E
12	Mary	89	3	M



Situations Requiring Two or More Transpositions

Dat4_Transpose:

Dat4:

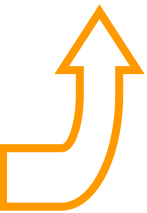
	S_NAME	E1	E2	E3	M1	M2	M3
1	John	89	90	92	78	89	90
2	Mary	92	.	81	76	91	89

	Test_num	John_E	John_M	Mary_E	Mary_M
1	1	89	78	92	76
2	2	90	89	.	91
3	3	92	90	81	89

To transpose from
Dat4 → Dat4_Transpose,
we need a “transitional”
data set.



	S_NAME	Score	Test_num	Class
1	John	89	1	E
2	John	78	1	M
3	Mary	92	1	E
4	Mary	76	1	M
5	John	90	2	E
6	John	89	2	M
7	Mary	.	2	E
8	Mary	91	2	M
9	John	92	3	E
10	John	90	3	M
11	Mary	81	3	E
12	Mary	89	3	M




Situations Requiring Two or More Transpositions

Dat4_Transpose:

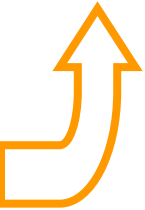
	Test_num	John_E	John_M	Mary_E	Mary_M
1	1	89	78	92	76
2	2	90	89	.	91
3	3	92	90	81	89

Dat4:

	S_NAME	E1	E2	E3	M1	M2	M3
1	John	89	90	92	78	89	90
2	Mary	92	.	81	76	91	89



	S_NAME	_NAME_	COL1
1	John	E1	89
2	John	E2	90
3	John	E3	92
4	John	M1	78
5	John	M2	89
6	John	M3	90
7	Mary	E1	92
8	Mary	E2	.
9	Mary	E3	81
10	Mary	M1	76
11	Mary	M2	91
12	Mary	M3	89



	S_NAME	Score	Test_num	Class
1	John	89	1	E
2	John	78	1	M
3	Mary	92	1	E
4	Mary	76	1	M
5	John	90	2	E
6	John	89	2	M
7	Mary	.	2	E
8	Mary	91	2	M
9	John	92	3	E
10	John	90	3	M
11	Mary	81	3	E
12	Mary	89	3	M


Situations Requiring Two or More Transpositions

Dat4_Transpose:

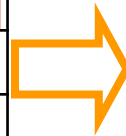
	Test_num	John_E	John_M	Mary_E	Mary_M
1	1	89	78	92	76
2	2	90	89	.	91
3	3	92	90	81	89

Dat4:

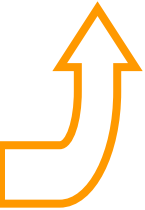
	S_NAME	E1	E2	E3	M1	M2	M3
1	John	89	90	92	78	89	90
2	Mary	92	.	81	76	91	89



	S_NAME	_NAME_	COL1
1	John	E1	89
2	John	E2	90
3	John	E3	92
4	John	M1	78
5	John	M2	89
6	John	M3	90
7	Mary	E1	92
8	Mary	E2	.
9	Mary	E3	81
10	Mary	M1	76
11	Mary	M2	91
12	Mary	M3	89



	S_NAME	Score	Test_num	Class
1	John	89	1	E
2	John	78	1	M
3	Mary	92	1	E
4	Mary	76	1	M
5	John	90	2	E
6	John	89	2	M
7	Mary	.	2	E
8	Mary	91	2	M
9	John	92	3	E
10	John	90	3	M
11	Mary	81	3	E
12	Mary	89	3	M



Situations Requiring Two or More Transpositions

Dat4:

	S_NAME	E1	E2	E3	M1	M2	M3
1	John	89	90	92	78	89	90
2	Mary	92	.	81	76	91	89

Dat4_Transpose:

	Test_num	John_E	John_M	Mary_E	Mary_M
1	1	89	78	92	76
2	2	90	89	.	91
3	3	92	90	81	89

	S_NAME	_NAME_	COL1
1	John	E1	89
2	John	E2	90
3	John	E3	92
4	John	M1	78
5	John	M2	89
6	John	M3	90
7	Mary	E1	92
8	Mary	E2	.
9	Mary	E3	81
10	Mary	M1	76
11	Mary	M2	91
12	Mary	M3	89

Step1:

```
proc sort data=dat4
        out=dat4_sort1;
        by s_name;
run;

proc transpose data=dat4_sort1
        out=dat4_out1;
        by s_name;
run;
```

Situations Requiring Two or More Transpositions

Dat4_Transpose:

Dat4:

	Test_num	John_E	John_M	Mary_E	Mary_M
1	1	89	78	92	76
2	2	90	89	.	91
3	3	92	90	81	89

	S_NAME	E1	E2	E3	M1	M2	M3
1	John	89	90	92	78	89	90
2	Mary	92	.	81	76	91	89

Step2:

```
data dat4_out1a;
  set dat4_out1;
  test_num=substr(_name_,2);
  class=substr(_name_,1,1);
run;
```

	S_NAME	_NAME_	COL1	Test_num	Class
1	John	E1	89	1	E
2	John	E2	90	2	E
3	John	E3	92	3	E
4	John	M1	78	1	M
5	John	M2	89	2	M
6	John	M3	90	3	M
7	Mary	E1	92	1	E
8	Mary	E2	.	2	E
9	Mary	E3	81	3	E
10	Mary	M1	76	1	M
11	Mary	M2	91	2	M
12	Mary	M3	89	3	M

Situations Requiring Two or More Transpositions

Dat4_Transpose:

	Test_num	John_E	John_M	Mary_E	Mary_M
1	1	89	78	92	76
2	2	90	89	.	91
3	3	92	90	81	89

Dat4:

	S_NAME	E1	E2	E3	M1	M2	M3
1	John	89	90	92	78	89	90
2	Mary	92	.	81	76	91	89

	S_NAME	_NAME_	COL1	Test_num	Class
1	John	E1	89	1	E
2	John	M1	78	1	M
3	Mary	E1	92	1	E
4	Mary	M1	76	1	M
5	John	E2	90	2	E
6	John	M2	89	2	M
7	Mary	E2	.	2	E
8	Mary	M2	91	2	M
9	John	E3	92	3	E
10	John	M3	90	3	M
11	Mary	E3	81	3	E
12	Mary	M3	89	3	M

Step3:

```
proc sort data=dat4_out1a
          out=dat4_sort2;
          by test_num s_name;
run;
```

Situations Requiring Two or More Transpositions

Dat4_Transpose:


Dat4:

	Test_num	John_E	John_M	Mary_E	Mary_M
1	1	89	78	92	76
2	2	90	89	.	91
3	3	92	90	81	89

	S_NAME	E1	E2	E3	M1	M2	M3
1	John	89	90	92	78	89	90
2	Mary	92	.	81	76	91	89

	S_NAME	_NAME_	COL1	Test_num	Class
1	John	E1	89	1	E
2	John	M1	78	1	M
3	Mary	E1	92	1	E
4	Mary	M1	76	1	M
5	John	E2	90	2	E
6	John	M2	89	2	M
7	Mary	E2	.	2	E
8	Mary	M2	91	2	M
9	John	E3	92	3	E
10	John	M3	90	3	M
11	Mary	E3	81	3	E
12	Mary	M3	89	3	M

Step4:



```

proc transpose
    data=dat4_sort2
    out=dat4_out2(drop=_name_)
    delimiter=_;
    by test_num;
    var col1;
    id name class;
run;
    
```