

## Is Something Missing? Part One

We often receive calls to our help desk where a user is getting a missing value as a result and does not know why. Missing values in SAS can be defined as an unknown or invalid response. This article will explore how SAS handles missing values. How to replace them if necessary, and ways to handle the missing values in both the DATA and the PROC step will be covered in our next issue.

### Checking for missing values

Frequencies are a great way to check for missing values. In our example, we have a dataset called HOURRATE and want to get a distribution of the variable HOURS3, so we run a PROC FREQ.

```
proc freq data=hourrate;
  table hours3;
run;
```

| The FREQ Procedure    |           |         |                      |                    |  |
|-----------------------|-----------|---------|----------------------|--------------------|--|
| HOURS3                | Frequency | Percent | Cumulative Frequency | Cumulative Percent |  |
| 1                     | 1         | 33.33   | 1                    | 33.33              |  |
| 10                    | 1         | 33.33   | 2                    | 66.67              |  |
| 20                    | 1         | 33.33   | 3                    | 100.00             |  |
| Frequency Missing = 1 |           |         |                      |                    |  |

Notice that the number of missing values is listed at the end. The value missing is not included in the distribution because, by default, all SAS procedures disregard missing values. To force the SAS procedure to include missing values, the MISSING option must be specified.

```
proc freq data=hourrate;
  table hours3 / missing;
run;
```

| The FREQ Procedure |           |         |                      |                    |  |
|--------------------|-----------|---------|----------------------|--------------------|--|
| HOURS3             | Frequency | Percent | Cumulative Frequency | Cumulative Percent |  |
| .                  | 1         | 25.00   | 1                    | 25.00              |  |
| 1                  | 1         | 25.00   | 2                    | 50.00              |  |
| 10                 | 1         | 25.00   | 3                    | 75.00              |  |
| 20                 | 1         | 25.00   | 4                    | 100.00             |  |

In SAS, arithmetic operators that use variables that contain missing values will always return a missing value. In this example, ALLHOURS is created by adding HOURS1, HOURS2, and HOURS3. If ANY of the values are missing, the result will be missing. For example:

```
data totals;
  set hourrate;
  allhours=hours1+hours2+hours3;
run;
proc print data=totals;
  title 'Sum Hours';
run;
```

Partial Log:

```
499 data totals;
500 set hourrate;
501 allhours=hours1+hours2+hours3;
502 run;
```

NOTE: Missing values were generated as a result of performing an operation on missing values. Each place is given by: (Number of times) at (Line): (Column).  
1 at 501:23

NOTE: There were 4 observations read from the data set WORK.HOURRATE.

NOTE: The data set WORK.TOTALS has 4 observations and 6 variables.

| Obs | NAME     | BI LL_AMT | HOURS1 | HOURS2 | HOURS3 | ALLHOURS |
|-----|----------|-----------|--------|--------|--------|----------|
| 1   | ROBERT   | 1422      | 15     | 15     | 10     | 40       |
| 2   | BEATRICE | 771       | 10     | 10     | 20     | 40       |
| 3   | RICHARD  | 2241      | 40     | 2      | .      | .        |
| 4   | ELAINE   | 1987      | 20     | 20     | 1      | 41       |

SAS functions ignore missing values. To minimize the creation of missing values in your calculation, replace the hard '+' operators with the SUM function. SUM adds values, but ignores any missing values instead of propagating them.

Continued on page 4.....



SYSTEMS SEMINAR CONSULTANTS, INC.  
2997 Yarmouth Greenway Drive  
Madison, WI 53711  
www.sys-seminar.com  
train@sys-seminar.com  
(608) 278-9964

## IN THIS ISSUE

**Is Something Missing? Part One**.....  
Gerry Frey.....Page 1

**President's Letter**.....  
Steve First.....Page 2

**Statistics Corner- ANOVAs**.....  
Katie Ronk.....Page 2

**Puzzler #6 Solution**.....  
Teresa Schudrowitz....Page 3

**Quick Tips**.....  
Katie Ronk.....Page 2, 3

**2005 Public Schedule**.....  
.....Page 8

**Technical Credit and Recognition**.....Page 8

## REMINDER:

The Missing Semicolon will be an e-newsletter. If you receive it via US mail, give us your email address at [www.sys-seminar.com](http://www.sys-seminar.com).



## Letter From the President



Dear SAS User:

We are pleased to announce that *The Missing Semicolon* will be quarterly again! After this issue, our newsletter will only be available in electronic format, but that means that our next issue will be coming out this Spring. If you receive the newsletter via US Mail, please sign up by email ( or refer a colleague) at

[http://www.sys-seminar.com/publications\\_signup.php](http://www.sys-seminar.com/publications_signup.php).

I would like to thank our loyal readers, and I am very excited that we will be able to provide you with many more SAS tips and tricks with a quarterly newsletter!

Sincerely,

Steven First,  
President



## Statistics Corner - ANOVAs

Analysis of Variance (ANOVA) is a procedure used to determine if there is a statistically significant difference in a dependent variable between two or more values of a classification variable(s), or if the difference is related to chance alone. The test compares the variance of the dependent variable's mean overall with the variance of the dependent variable's mean in each of the test groups. If the overall variance is much more than the variance within the test groups, the test is more likely to be significant.

### PHARMACEUTICAL EXAMPLE

A drug company might be testing a drug that lowers blood pressure. In a blind study, they give one group of patients with high blood pressure a placebo and another group the experimental drug. After a certain time period, they collect the blood pressures of both groups and want to see if the drug had the desired affect. One way to test this would be to run an ANOVA comparing the two groups (placebo drug and active drug) to see if there was a significant difference in the change in blood pressure.

Let's take a look at our data:

```
TITLE "Blood Pressure Study Data";
PROC PRINT DATA=STUDYDATA(OBS=10);
  VAR DRUGTYPE BPCHANGE;
RUN;
```

Blood Pressure Study Data

| Obs | DrugType | BPChange |
|-----|----------|----------|
| 1   | P        | 3        |
| 2   | P        | -5       |
| 3   | A        | -20      |
| 4   | A        | -4       |
| 5   | P        | -2       |
| 6   | P        | 11       |
| 7   | P        | -10      |
| 8   | P        | 0        |
| 9   | P        | 0        |
| 10  | P        | 16       |

PROC ANOVA is the SAS procedure for running analysis of variance tests. The two required statements are CLASS and MODEL.

```
PROC ANOVA < options >;
  CLASS variables; (Required)
  MODEL dependents=effects < / options >; (Required)
  ABSORB variables;
  BY variables;
  FREQ variable;
  MANOVA < test-options >< / detail-options >;
  MEANS effects < / options >;
  REPEATED factor-specification < / options >;
  TEST < H=effects > E=effect;
RUN;
QUIT;
```

*Continued on page 5.....*



## QUICK TIP

When reading in a flat file, the LIST statement can be used for debugging. The list statement will display the contents of the input buffer.

For example:

```
data accounts;
  infile acctfile;
  input @1 acctnumber $10.
        @11 amtpaid 8.2
        @19 balance 8.2 ;
  if _n_ =1 then list;
run;
```

```
RULE:  ---+---1---+---2---+---3---+---4---+---5
      12345      120      231.59
```



**SYSTEMS SEMINAR CONSULTANTS, INC.**

Copyright © 2005 Systems Seminar Consultants, Inc. Madison, WI  
All rights reserved. Printed in USA. The Missing Semicolon is a trademark of  
Systems Seminar Consultants, Inc. SAS, SAS/IntrNet, and SAS/ACCESS are  
registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.

## Puzzler #6 Solution

Thank you to every one whom responded to last issue's puzzler.

As a reminder, here is the original problem. A recent request required a file of student test scores to be transposed. The original data contains one record per student with a variable for each test score. Later processing requires the data to be one observation per test score per student.

PROC TRANSPOSE was used to transpose the data in the data set. There are 10 test scores per student on the original data set. When the new data set is created with PROC TRANSPOSE there are only 8 observations per student, rather than the expected 10 observation.

### Data set involved

The SCORES data set used in the puzzler is shown below:

| Obs | STUDENT | TEST1 | TEST2 | TEST3 | TEST4 | TEST5 | TEST6 |
|-----|---------|-------|-------|-------|-------|-------|-------|
| 1   | 1       | 90    | 95    | P     | P     | 92    | 96    |
| 2   | 2       | 100   | 98    | P     | P     | 95    | 97    |
| 3   | 3       | 82    | 90    | P     | F     | 85    | 92    |
| 4   | 4       | 85    | 80    | F     | P     | 89    | 87    |
| 5   | 5       | 80    | 88    | P     | P     | 95    | 100   |

TEST7 TEST8 TEST9 TEST10

|     |     |    |    |
|-----|-----|----|----|
| 93  | 100 | 88 | 94 |
| 100 | 92  | 90 | 95 |
| 87  | 90  | 82 | 89 |
| 85  | 88  | 78 | 85 |
| 92  | 100 | 90 | 93 |

### Initial Program and Output (Unexpected Results)

```
PROC TRANSPOSE DATA=SCORES
  OUT=SCORES2(RENAME=( _NAME_=TEST COL1=SCORE));
  BY STUDENT;
RUN;
```

```
PROC PRINT DATA=SCORES2;
RUN;
Obs  STUDENT  TEST  SCORE
1    1        TEST1  90
2    1        TEST2  95
3    1        TEST5  92
4    1        TEST6  96
5    1        TEST7  93
6    1        TEST8  100
7    1        TEST9  88
8    1        TEST10 94
9    2        TEST1  100
10   2        TEST2  98
11   2        TEST5  95
12   2        TEST6  97
...  .        ...    ..
38   5        TEST8  100
39   5        TEST9  90
40   5        TEST10 93
```

## QUICK TIP

Datasets can be password protected using the password dataset option. To create a dataset with a password, use code as shown below:

```
data safedata (password='Katie');
  set olddata;
run;
```

To process this data, code the password as a dataset option. For example:

```
proc print data=safedata (password='Katie');
run;
```



### Explanation

The variables TEST1 through TEST10 are a combination of both numeric and character variables. The PROC TRANSPOSE, as written, will only transpose the numeric variables. The character variables will be ignored. Thus, the observations for TEST3 and TEST4 will not be included in the final results.

There are several possible solutions for this issue. The most straightforward solution is to include a VAR statement. All variables listed with a VAR statement will be transposed.

### Revised Program and Output

```
PROC TRANSPOSE DATA=SCORES
  OUT=SCORES2(RENAME=( _NAME_=TEST COL1=SCORE));
  BY STUDENT;
  VAR TEST1 TEST2 TEST3 TEST4 TEST5 TEST6 TEST7
    TEST8 TEST9 TEST10;
RUN;
PROC PRINT DATA=SCORES2;
  TITLE1 'PUZZLER - RESULTS WITH VAR STATEMENT';
RUN;
```

| Obs | STUDENT | TEST   | SCORE |
|-----|---------|--------|-------|
| 1   | 1       | TEST1  | 90    |
| 2   | 1       | TEST2  | 95    |
| 3   | 1       | TEST3  | P     |
| 4   | 1       | TEST4  | P     |
| 5   | 1       | TEST5  | 92    |
| 6   | 1       | TEST6  | 96    |
| 7   | 1       | TEST7  | 93    |
| 8   | 1       | TEST8  | 100   |
| 9   | 2       | TEST9  | 88    |
| 10  | 2       | TEST10 | 94    |
| 11  | 2       | TEST1  | 100   |
| 12  | 2       | TEST2  | 98    |
| ... | ..      | ...    | ..    |
| 48  | 5       | TEST8  | 100   |
| 49  | 5       | TEST9  | 90    |
| 50  | 5       | TEST10 | 93    |

*Continued on page 7.....*

# Is Something Missing-Part 1

## CONTINUED FROM PAGE 1

```
data better;
set hourrate;
allhours=sum(hours1, hours2, hours3);
run;
proc print data=better;
title 'Allhours Is Sum Of Known Values';
run;
```

| Obs | NAME     | BILL_AMT | HOURS1 | HOURS2 | HOURS3 | ALLHOURS |
|-----|----------|----------|--------|--------|--------|----------|
| 1   | ROBERT   | 1422     | 15     | 15     | 10     | 40       |
| 2   | BEATRICE | 771      | 10     | 10     | 20     | 40       |
| 3   | RICHARD  | 2241     | 40     | 2      | .      | 42       |
| 4   | ELAINE   | 1987     | 20     | 20     | 1      | 41       |

Sometimes, despite the use of functions, missing values can result from using a missing value in a calculation or doing some operation that is illegal mathematically. This will result in data error messages in the log.

```
data zero;
set hourrate;
hour_rate=input(bill_amt, 5.2)/sum(hours1, -hours2);
run;
proc print data=zero;
title 'Hour_rate is bill_amt / hours1-hours2';
run;
```

Results in the log messages: (Partial Log)

```
NOTE: Division by zero detected at line 322 column
25.
NAME=ROBERT BILL_AMT=1422 HOURS1=15 HOURS2=15
HOURS3=10 HOUR_RATE=. _ERROR_=1 _N_=1
NOTE: Division by zero detected at line 322 column
25.
NAME=BEATRICE BILL_AMT=771 HOURS1=10 HOURS2=10
HOURS3=20 HOUR_RATE=. _ERROR_=1 _N_=2
NOTE: Division by zero detected at line 322 column
25.
NAME=ELAINE BILL_AMT=1987 HOURS1=20 HOURS2=20
HOURS3=1 HOUR_RATE=. _ERROR_=1 _N_=4
NOTE: Mathematical operations could not be
performed at the following places.
The results of the operations have been set
to missing values.
Each place is given by: (Number of times) at
(Line):(Column). 3 at 322:25
NOTE: There were 4 observations read from the data
set WORK.HOURRATE.
```

| Obs | NAME     | BILL_AMT | HOURS1 | HOURS2 | HOURS3 | HOUR_RATE |
|-----|----------|----------|--------|--------|--------|-----------|
| 1   | ROBERT   | 1422     | 15     | 15     | 10     | .         |
| 2   | BEATRICE | 771      | 10     | 10     | 20     | .         |
| 3   | RICHARD  | 2241     | 40     | 2      | .      | 0.58974   |
| 4   | ELAINE   | 1987     | 20     | 20     | 1      | .         |

To avoid this error message, we can check the value of the denominator for a zero or missing and avoid the division in such cases.

```
data zero;
set hourrate;
if sum(hours1, -hours2) not in (.,0) then
hour_rate=input(bill_amt, 5.2)/sum(hours1, -
hours2);
run;
proc print data=zero;
title 'hour_rate is bill_amt / hours1-hours2';
run;
```

Partial Log:

```
NOTE: There were 4 observations read from the
data set WORK.HOURRATE.
NOTE: The data set WORK.ZERO has 4 observations
and 6 variables.
```

The resulting report looks the same:

| Obs | NAME     | BILL_AMT | HOURS1 | HOURS2 | HOURS3 | HOUR_RATE |
|-----|----------|----------|--------|--------|--------|-----------|
| 1   | ROBERT   | 1422     | 15     | 15     | 10     | .         |
| 2   | BEATRICE | 771      | 10     | 10     | 20     | .         |
| 3   | RICHARD  | 2241     | 40     | 2      | .      | 0.58974   |
| 4   | ELAINE   | 1987     | 20     | 20     | 1      | .         |

We can also use the SUM function in PROC SQL:

```
title 'Sql Select Report';
proc sql;
select name,
bill_amt,
hours1,
hours2,
hours3,
sum(hours1, hours2, hours3) as allhours
from hourrate;
quit;
```

Results in the report:

| NAME     | BILL_AMT | HOURS1 | HOURS2 | HOURS3 | ALLHOURS |
|----------|----------|--------|--------|--------|----------|
| ROBERT   | 1422     | 15     | 15     | 10     | 40       |
| BEATRICE | 771      | 10     | 10     | 20     | 40       |
| RICHARD  | 2241     | 40     | 2      | .      | 42       |
| ELAINE   | 1987     | 20     | 20     | 1      | 41       |

## Invited SUGI 30 Speakers

|                    |   |
|--------------------|---|
| Steven J. First    | <i>SAS® Macro Variables and Simple Macro Programs</i>                   |
| Teresa Schudrowitz | <i>Arrays Made Easy: An Introduction to Arrays and Array Processing</i> |
| Gerald D. Frey     | <i>SAS® Excels!</i>   |

More on Missing values in our Spring Issue!



# Stastics Corner - ANOVAs

## CONTINUED FROM PAGE 2

The CLASS statement is where the classification variable /variables are listed. These variables are generally categorical and not continuous, meaning that each observation is grouped into a category. For instance: placebo drug vs experimental drug or region of the country. We can create categorical variables from continuous variables as we will see later in the credit example. In our example, we will be using the variable DRUGTYPE as our classification example.

The dependent variable and effects are listed on the MODEL statement. We are looking to see if the type of drug has an effect on BP. Therefore, BPCHANGE is our dependent variable, so it is listed to the left of the equal sign on the model statement.

The MEANS statement can be coded to generate a report of the averages for the dependent variable broken out by the class variables.

CODE:

```
PROC ANOVA DATA=STUDYDATA;
  CLASS DrugType;
  MODEL BPChange= DrugType;
  MEANS DrugType;
RUN;
QUIT;
```

OUTPUT Page 1:

The ANOVA Procedure

Dependent Variable: BPChange

| Source          | DF    | Squares     | Mean Square | F Value | Pr > F |
|-----------------|-------|-------------|-------------|---------|--------|
| Model           | 1     | 91693.648   | 91693.648   | 667.81  | <.0001 |
| Error           | 16928 | 2324289.806 | 137.304     |         |        |
| Corrected Total | 16929 | 2415983.454 |             |         |        |

R-Square    Coeff Var    Root MSE    BPChange Mean  
0.037953    -301.3438    11.71770    -3.888482

| Source   | DF | Anova SS    | Mean Square | F Value | Pr > F |
|----------|----|-------------|-------------|---------|--------|
| DrugType | 1  | 91693.64800 | 91693.64800 | 667.81  | <.0001 |

OUTPUT Page 2

| Level of DrugType | N    | -----BPChange-----<br>Mean | Std Dev     |
|-------------------|------|----------------------------|-------------|
| A                 | 7769 | -6.41562621                | 13.1222219  |
| P                 | 9161 | -1.74533348                | 10.3787282: |

The probability value of <.0001 means that the difference in blood pressure change between the placebo drug and the active drug is more than would be expected by chance alone. The output on page two shows that the active group had a decrease in blood pressure on average of -6.41 and the placebo group had a decrease of -1.74. As the probability is statistically significant, we have evidence that the active group is more effective in lowering blood pressure by a greater amount than is the placebo.

## CREDIT EXAMPLE

In the finance industry, credit scores are often used as a mechanism for companies to screen for customers that may default on loans. A company has collected data on it's customers and wants to check to see if there is a difference between the initial credit score of customers that defaulted on their loans and those that did not. What follows is the code to test

to see if there is a statistically significant difference.

```
PROC ANOVA DATA=CUSTOMERS;
  CLASS Default tLoan;
  MODEL Credi tScore=Default tLoan;
  MEANS Default tLoan / T;
RUN;
QUIT;
```

OUTPUT Page 1:

The ANOVA Procedure

Dependent Variable: Credi tScore

| Source          | DF    | Sum of Squares | Mean Square | F Value | Pr > F |
|-----------------|-------|----------------|-------------|---------|--------|
| Model           | 1     | 13116602.0     | 13116602.0  | 696.48  | <.0001 |
| Error           | 10021 | 188723175.5    | 18832.8     |         |        |
| Corrected Total | 10022 | 201839777.5    |             |         |        |

R-Square    Coeff Var    Root MSE    Credi tScore Mean  
0.064985    25.61534    137.2325    535.7435

| Source       | DF | Anova SS    | Mean Square | F Value | Pr > F |
|--------------|----|-------------|-------------|---------|--------|
| Defaul tLoan | 1  | 13116602.03 | 13116602.03 | 696.48  | <.0001 |

OUTPUT Page 2:

The ANOVA Procedure

| Level of Defaul tLoan | N    | -----Credi tScore-----<br>Mean | Std Dev    |
|-----------------------|------|--------------------------------|------------|
| N                     | 8509 | 551.002821                     | 144.397190 |
| Y                     | 1514 | 449.982827                     | 86.523169  |

The results show that there is a significant difference in the credit score between the group of customers that defaulted on their loans and those that did not. Note that we coded the credit score as our 'dependent' variable, meaning we are assuming the relationship means that credit score is dependent on whether someone defaulted on their loan or not. While this might be true and the test still works, we are actually trying to predict the opposite.

To turn the model around and make credit score into a classification variable and default into the dependent variable, some changes need to be made to the data. First, continuous variables, such as credit score, can not be coded on the CLASS statement. However, credit scores can easily be put into categories by coding a user defined format (see format below). The format is then applied in the PROC ANOVA step. Another problem with swapping the model is the variable DefaultLoan is currently a character categorical variable. If this field is to be used on the MODEL statement as the dependent variable, not the classification variable, it must be a numeric variable. A new numeric variable NumDefault is added to the table in the data step below.

*Continued on page 6.....*

# Statistics Corner-ANOVAs

## CONTINUED FROM PAGE 5

```
PROC FORMAT;
  VALUE SCOR
  300-475=' 300-475'
  476-650=' 476-650'
  651-725=' 651-725'
  726-820=' 726-820';
RUN;
```

```
DATA CUSTOMERS;
  SET CUSTOMERS;
  IF DefaultLoan='Y' then NumDefault=1;
  ELSE IF DefaultLoan='N' Then NumDefault=0;
RUN;
```

```
PROC ANOVA DATA=CUSTOMERS;
  CLASS CreditScore;
  FORMAT CreditScore SCOR.;
  MODEL NumDefault=CreditScore;
  MEANS CreditScore;
RUN;
QUIT;
```

### OUTPUT Page 1:

The ANOVA Procedure  
Class Level Information

| Class                  | Levels | Values                          |
|------------------------|--------|---------------------------------|
| CreditScore            | 4      | 300-475 476-650 651-725 726-820 |
| Number of observations | 10023  |                                 |

### OUTPUT Page 2:

The ANOVA Procedure

Dependent Variable: NumDefault

| Source          | DF    | Sum of Squares | Mean Square | F Value | Pr > F |
|-----------------|-------|----------------|-------------|---------|--------|
| Model           | 3     | 86.442712      | 28.814237   | 240.80  | <.0001 |
| Error           | 10019 | 1198.863683    | 0.119659    |         |        |
| Corrected Total | 10022 | 1285.306395    |             |         |        |

| R-Square | Coeff Var | Root MSE | NumDefault Mean |
|----------|-----------|----------|-----------------|
| 0.067255 | 229.0048  | 0.345918 | 0.151053        |

| Source      | DF | Anova SS    | Mean Square | F Value | Pr > F |
|-------------|----|-------------|-------------|---------|--------|
| CreditScore | 3  | 86.44271182 | 28.81423727 | 240.80  | <.0001 |

### OUTPUT Page 3:

The ANOVA Procedure

| Level of    | -----NumDefault----- |            |            |
|-------------|----------------------|------------|------------|
| CreditScore | N                    | Mean       | Std Dev    |
| 300-475     | 3842                 | 0.23321187 | 0.42293102 |
| 476-650     | 3597                 | 0.17180984 | 0.37726753 |
| 651-725     | 1330                 | 0.00000000 | 0.00000000 |
| 726-800     | 1254                 | 0.00000000 | 0.00000000 |

The output from this model now shows that there are significant differences between the four credit score groups on default rates. It appears that 23% of the loans from the 300-475 group defaulted on their loans, while 0% in the two highest groups did not. This begs the question, are

all of the groups significantly different from each other, or is there just an overall difference between the four groups? By adding the T-Test option to the means statement, a report will be produced that indicates the credit score groups with significant differences in their default rates.

```
PROC ANOVA DATA=CUSTOMERS;
  CLASS CreditScore;
  FORMAT CreditScore SCOR.;
  MODEL NumDefault=CreditScore;
  MEANS CreditScore / t;
RUN;
QUIT;
```

### OUTPUT Page 3:

The ANOVA Procedure

t Tests (LSD) for NumDefault

NOTE: This test controls the Type I comparisonwise error rate, not the experimentwise error rate.

|                          |          |
|--------------------------|----------|
| Alpha                    | 0.05     |
| Error Degrees of Freedom | 10019    |
| Error Mean Square        | 0.119659 |
| Critical Value of t      | 1.96020  |

Comparisons significant at the 0.05 level are indicated by \*\*\*.

| CreditScore Comparison | Difference Between Means | 95% Confidence Limits   |
|------------------------|--------------------------|-------------------------|
| 300-475 - 476-650      | 0.061402                 | 0.045670 0.077134 ***   |
| 300-475 - 651-725      | 0.233212                 | 0.211639 0.254784 ***   |
| 300-475 - 726-800      | 0.233212                 | 0.211159 0.255265 ***   |
| 476-650 - 300-475      | -0.061402                | -0.077134 -0.045670 *** |
| 476-650 - 651-725      | 0.171810                 | 0.150049 0.193570 ***   |
| 476-650 - 726-800      | 0.171810                 | 0.149573 0.194047 ***   |
| 651-725 - 300-475      | -0.233212                | -0.254784 -0.211639 *** |
| 651-725 - 476-650      | -0.171810                | -0.193570 -0.150049 *** |
| 651-725 - 726-800      | 0.000000                 | -0.026690 0.026690      |
| 726-800 - 300-475      | -0.233212                | -0.255265 -0.211159 *** |
| 726-800 - 476-650      | -0.171810                | -0.194047 -0.149573 *** |
| 726-800 - 651-725      | 0.000000                 | -0.026690 0.026690      |

In the final output, each group is compared with each other group to see if they are statistically significant with each other. All comparisons that are statistically significant (defined as having a p-value of <.04) are indicated by '\*\*\*'.

As you can see, most of the groups are indeed statistically different from one another. However, the credit score groupings 651-725 and 726-800 are not significantly different from each other, in fact, they both have a default rate of 0%.

Please note that the data used in both of these examples are fictitious. An ANOVA is just one way to determine relationships between variables.



# Puzzler # 6 Solution

## CONTINUED FROM PAGE 3

Since the VAR statement is being used all variables to be transposed must be listed. This data only contains ten scores. What about if there are 100 scores? The TEST variables are all named sequentially. Therefore, the VAR statement can be modified to use a range of variables.

```
VAR TEST1-TEST10;
```

All variables between TEST1 and TEST10 will now be used for the transpose. Another way the range of variables could be written is as:

```
VAR TEST;
```

All variable names beginning with TEST will be used for the transpose. In this situation, the variable names allow any of the above variations for listing the variables on the VAR statement. Since all variable names may not be named sequentially or begin with common prefixes this often is not the case. In this scenario the variable reference `_ALL_` can be used instead of listing every single variable on the VAR statement.

```
VAR _ALL_;
```

Now every single variable on the data set will be used for the transposition. Therefore some additional modifications to the PROC TRANSPOSE are necessary since we do not want the variable STUDENT to be included.

```
PROC TRANSPOSE DATA=SCORES
  OUT=SCORES2(RENAME=(NAME=TEST COL1=SCORE)
  WHERE =(TEST NE 'STUDENT'));
  BY STUDENT;
  VAR _ALL_;
RUN;
```

While these solutions now give us the correct observations, we still have a problem with the display of the SCORE value. The character values are left justified, while the numeric values are right justified.

Again, we have many possible solutions. One possibility is to use the RIGHT or LEFT function in a data step after the PROC TRANSPOSE step to reset the values.

```
DATA SCORES3;
  SET SCORES2;
  SCORE = RIGHT(SCORE);
RUN;
```

A second solution is to set special missing values for the character values prior to completing the transpose. This would be accomplished by including the MISSING statement in a data step.

```
DATA SCORES_REVISED (KEEP=STUDENT TEST);
  MISSING P F;
  LENGTH TEST1-TEST10 3;
  SET SCORES (RENAME=(TEST3=CHAR_TEST3
  TEST4=CHAR_TEST4));
  TEST3 = INPUT(CHAR_TEST3, ?? BEST4.);
  TEST4 = INPUT(CHAR_TEST4, ?? BEST4.);
RUN;
```

The original PROC TRANSPOSE can now be executed and receives the expected results with the scores right justified.

### Final Results

| Obs | STUDENT | TEST   | SCORE |
|-----|---------|--------|-------|
| 1   | 1       | TEST1  | 90    |
| 2   | 1       | TEST2  | 95    |
| 3   | 1       | TEST3  | P     |
| 4   | 1       | TEST4  | P     |
| 5   | 1       | TEST5  | 92    |
| 6   | 1       | TEST6  | 96    |
| 7   | 1       | TEST7  | 93    |
| 8   | 1       | TEST8  | 100   |
| 9   | 2       | TEST9  | 88    |
| 10  | 2       | TEST10 | 94    |
| 11  | 2       | TEST1  | 100   |
| 12  | 2       | TEST2  | 98    |
| ... | ...     | ....   | ...   |
| 48  | 5       | TEST8  | 100   |
| 49  | 5       | TEST9  | 90    |
| 50  | 5       | TEST10 | 93    |

### Winners

The first three readers to send in the correct solution were:

- " Shalini Maujanatha, Arkansas Department of Health
- " John Hunter, Bone Care International
- " Bala Krishnan, Bandag, Inc.

These winners will each receive a \$100 training certificate.

We would also like to extend recognition to the following readers for note-worthy solutions:

- " Michael Bieberitz, Wisconsin DOT
- " Rick Allen, Citi Capital
- " Peter Crawford, Crawford Software Consultancy Limited, UK
- " Pon Su, VA Palo Alto Health Care System

Congratulations to all of our winners! Look for a new puzzler in future issues of The Missing Semicolon.



**SYSTEMS SEMINAR CONSULTANTS, INC.**  
1-800-997-7081 ♦ www.sys-seminar.com

**We're Hiring!**

Fill out our online application:  
<http://www.sys-seminar.com/onlineapp.php>

**Staff Placement**

Call us to discuss your need for full-time and contract employees.

♦ **The Missing Semicolon**

Tips from the experts!  
Sign up for our complimentary professional newsletter.

♦ **SAS Training**

Free follow up help desk  
Customized courses available  
Public and Onsite Training



|                          |                         |                    |
|--------------------------|-------------------------|--------------------|
| Accessing Databases      | Application Development | Data Cleaning      |
| Data & System Validation | Data Conversion         | Data Warehousing   |
| Data Extraction          | Efficiencies            | Project Management |
| Process Automation       | Analysis                | Reporting          |



## SAS Training Options

### **Onsite Training**

Get the benefits of training in your own environment.  
Keep travel to a minimum and save on cost.  
Course material can be customized to fit your needs.  
**Call (608) 278-9964 ext. 306 for details.**

### **Public On Demand**

Can't make it to a scheduled public class?  
Can't get the class you need onsite?  
Get it with Public On Demand!  
**Call (608) 278-9964 ext. 306 for details.**

## New Course Announcements

### **Tips, Tricks, and Techniques**

An eclectic mix of SAS tips and techniques that our consultants have used in contract programming.  
Visit [www.sys-seminar.com/pdfs/tips&tricks.pdf](http://www.sys-seminar.com/pdfs/tips&tricks.pdf) for details.

### **Advanced Macros**

A course designed for experienced SAS macro programmers who would like to use the advanced features of the SAS macro language.  
Visit [www.sys-seminar.com/pdfs/advancedmacros.pdf](http://www.sys-seminar.com/pdfs/advancedmacros.pdf) for details.

## PUBLIC CLASS SCHEDULE - MADISON, WI

### **Introduction to SAS®**

May 9-11  
September 12-14  
December 5-7

### **SAS® Report Writing**

September 26-27

### **Introduction to Proc Report**

September 28

### **Advanced SAS®**

June 6-8  
October 3-5

### **Exploiting ODS**

May 12-13  
December 8-9

### **SAS® Macros**

September 29-30

### **The SAS® SQL Procedure**

September 15

### **SAS® Efficiencies**

September 16

### **Tips & Tricks from the Experts**

June 10  
October 7

### **What's New in SAS® 9**

June 9  
October 6

To register call **(608) 278-9964** or visit [www.sys-seminar.com](http://www.sys-seminar.com).

## TECHNICAL CREDIT

### AND RECOGNITION



#### **Steve First**

President  
Letter from the President, Page 2

#### **Katie Minten Ronk**

Director of Operations  
Statistics Corner - ANOVAs, Page 2  
Quick Tips, Page 2 & 3



#### **Teresa Schudrowitz**

Trainer/Consultant  
Puzzler #6 Solution, Page 3



#### **Gerry Frey**

Trainer/Consultant  
Is Something Missing? Part One, Page 1



Editors.....Susan Bakken, Jennifer First, Ann Vinge

BULK RATE  
U.S. POSTAGE  
**PAID**  
MADISON, WI  
PERMIT #2783

**The next issue of The Missing Semicolon™ will be an e-newsletter. Make sure we have your email address! Sign up at [www.sys-seminar.com](http://www.sys-seminar.com)**