Paper CC03

Numeric Precision in SAS® – an Example

Varsha C. Shah, Dept. of Biostatistics, UNC-CH, NC

ABSTRACT

We are so much used to let the computers do our mathematical computations; it has become way of life.. We have more trust in the results produced by the computers than ourselves. And that is the way it should be.

So what is the purpose of this article? The transition from the world of absolute mathematics to practical world of computers is so smooth, so quick, that we are under influence of perfect conviction that it is one in the same. In general, we could almost say so. However, there is a small chance when the results are close, but not identical.

A small example of simple additions will illustrate one such case. Further, it illustrates that the functions in SAS® handle such anomalies.

INTRODUCTION

There are excellent articles available on SAS websites explaining numeric precision in calculations of numbers. For an applied social scientist the results are important, not the details of calculations in the computers. Same is true about a programmer.

All studies/projects/protocols create derived variables at some stage. It is desirable, even when not mandatory, to report the statistical summary of the derived variables using PROC MEANS or PROC UNIVARIATE etc. So, it is important that the results are unique.

In the following discussion, the fictitious data is created with carefully chosen values to illustrate the anomalies.

METHODOLOGY

Consider the following small example consisting of seven observations and four variables. A simple operation of addition/subtraction is performed in two different ways creating two derived variables:

1) DER_SUM_VAR, using SUM function,
2) DER_ARI_VAR, using arithmetic summation.

```
Data test1; input
Id $          var1      var2    var3     basevar;
Cards;
Aaa001        0       1.0     0.0330     1.0330
Bbb002        0       2.5     0.4268     2.9268
Ccc003        0       2.5     0.4268     2.9268
Ddd004        1       0.0     0.0330     1.0330
Xxx123        0       6.0     0.0000     1.0330
Yyy234        1       5.0     0.4268     2.9268
Zzz345        0       4.0     0.0330     1.0330
;run;
data test2;
     set test1;
     der_sum_var=basevar - sum(var1,var2,var3);*via sum function;
     der_ari_var =basevar-(var1+var2+var3);*plain arithmetic operation;
run;
proc means data=test2;
var der_sum_var der_ari_var;
run;
proc univariate data=test2;
var der_sum_var der_ari_var;
run;
proc print data=test2;
format der_sum_var 6.4 der_ari_var 20.18;
run;
```

RESULTS

The code should be authenticated by checking the results. The common practice is to print a few observations and check the calculations. But, as mentioned above, this is not sufficient, so let us investigate further, and examine the results of PROC MEANS.

Variable	N	Mean	Std Dev	Minimum	Maximum
der_sum_var	7	-1.6381429	2.1266760	-4.9670000	0
der_ari_var	7	-1.6381429	2.1266760	-4.9670000	5.551115E-17

We would expect the values in each column identical for the two rows representing the two derived variables. The last one has very close values but not identical.

Let us further examine the results of PROC UNIVARIATE.

Extreme Observations DER_SUM_VAR				Extreme Observations DER_ARI_VAR			
-----Lowest----		----Highest---		-----Lowest----		----Highest---	
Value	Obs	Value	Obs	Value	Obs	Value	Obs
-4.967	5	-3	7	-4.967	5	-3.00000E+00	7
-3.500	6	0	1	-3.500	6	-8.32667E-17	1
-3.000	7	0	2	-3.000	7	-8.32667E-17	4
0.000	4	0	3	-0.000	4	5.55112E-17	2
0.000	3	0	4	-0.000	1	5.55112E-17	3

The results on the left hand side under 'der_sum_var' match the hand calculations. However, some of the results on the right hand side differ, though the difference is infinitesimally small.

Obviously some values for the derived variables are identical, and some are not. The results of the derived variable 'der_sum_var' in both of the summaries above are exactly as expected, the other is not. Before we move on to 'how to avoid' such pitfalls, let us check observation by observation the computed values of the two variables. In order to do this, let us add a format statement for the derived variables for printing more spaces following the decimal point than default. The results of PROC PRINT are:

Id	var1	var2	var3	basevar	der_sum_var	der_ari_var
Aaa001	0	1.0	0.0330	1.0330	0.0000	-.000000000000000083
Bbb002	0	2.5	0.4268	2.9268	0.0000	0.000000000000000056
Ccc003	0	2.5	0.4268	2.9268	0.0000	0.000000000000000056
Ddd004	1	0.0	0.0330	1.0330	0.0000	-.000000000000000083
Xxx123	0	6.0	0.0000	1.0330	-4.967	-4.96700000000000000
Yyy234	1	5.0	0.4268	2.9268	-3.500	-3.50000000000000000
Zzz345	0	4.0	0.0330	1.0330	-3.000	-3.00000000000000000

The derived variables for the first four observations are slightly different.

CONCLUSION

The number system we are used to is base 10. Different machines use different bases like 2,8, or 16 for the operations depending on the platform. In general the results are identical, but there are exceptions as we examined above. Without further ado, the question is how to avoid the fuzzy difference and arrive at the unique results?

The various functions in SAS take care of such situation. As we observed above, the SUM function produced the desired results. Other functions like ROUND and TRUNC also will yield results consistent with what has become second nature to us all. Wherever possible, use the built-in functions.

The situation could also arise when the results are independently checked by two or more individuals. In this case, if the results are not identical, check the details of the data steps. As illustrated in the above example, if one person used the SAS functions and the other used the straight mathematical computations, they may have different end results – they both have correct code and correct logic and just different routes.

ACKNOWLEDGEMENT
The author would like to thank supervisor Kathy Roggenkamp for helpful comments.

CONTACT INFORMATION

Varsha C. Shah
CSCC, Department of Biostatistics, CB #8030
University of North Carolina
Chapel Hill, NC 27514
Phone: 919-966-6317
Fax: 919-966-6335
Email: uccvcs@mail.cscc.unc.edu