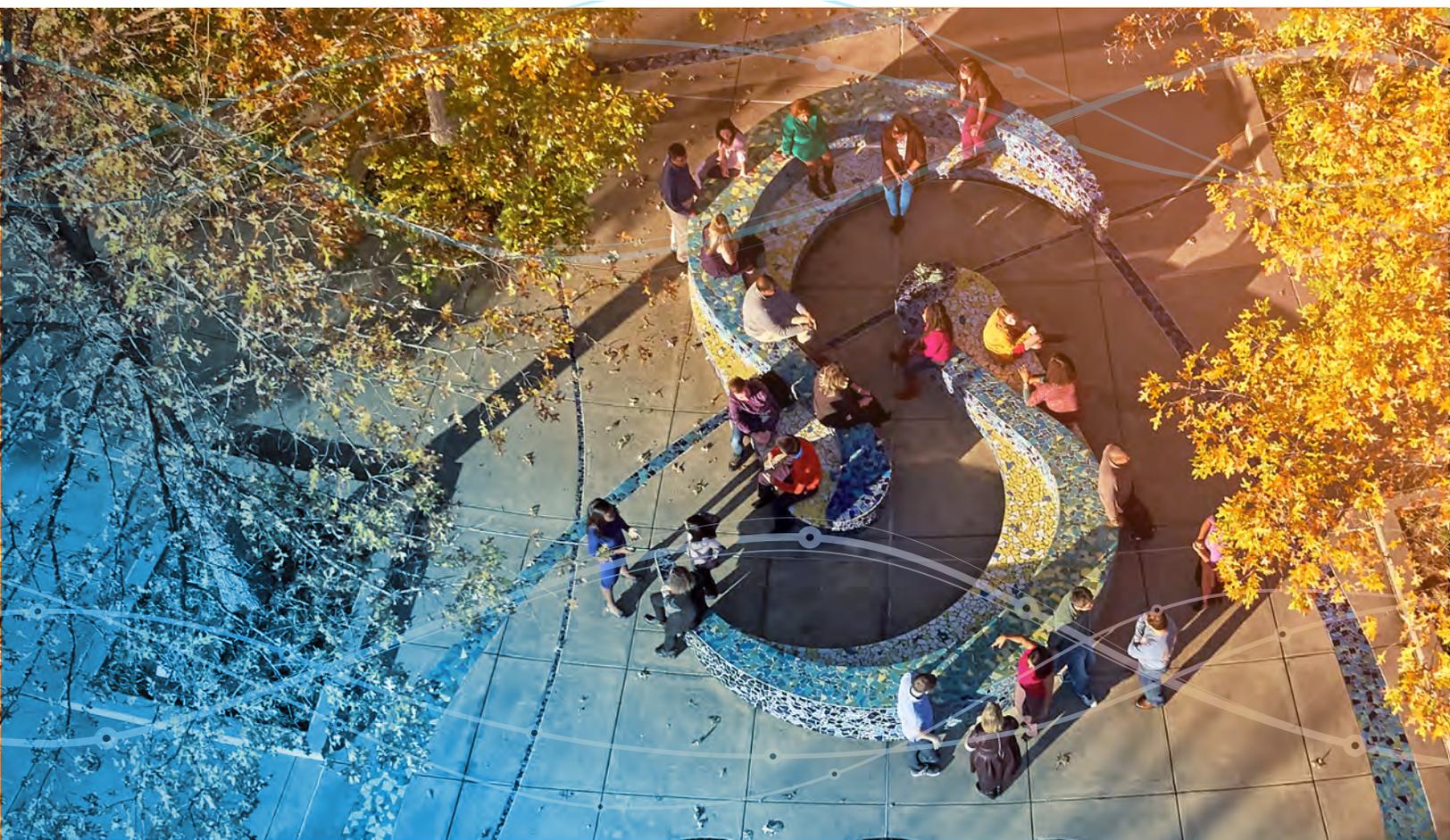


# The Quality Imperative: SAS Institute's Commitment to Quality

A corporate statement of SAS' commitment to product quality,  
service quality, and customer satisfaction



# Appendix 2: Validating an Analytical Component

## Numerical Accuracy

We use a variety of methods to verify the accuracy and precision of the results generated by our software. Libraries of regression tests using automated tools are run periodically throughout the testing cycle to test functionality and data integrity. These reusable libraries of tests cover syntax, options, functionality, valid and invalid data, errors, stress, and results for the procedure/function/solution. These tests are run, and differences are resolved, before the release is declared production.

## Validating an Analytical Component

Our development staff has the education, training, and experience to perform their assigned tasks. They use a variety of methods to verify, to the extent possible, that the software produces accurate, reliable, and numerically precise results. A combination of methods is used to validate an analytical product and the algorithms in it. These methods are listed below.

## Writing Independent Validation Code

Except when similar results are available within existing and previously tested SAS analytical software, independent verification of numerical results via SAS/IML, DATA step code, or hand calculations is performed whenever possible and practical. SAS/IML is a powerful matrix programming language that is used to re-create the numerical output, or pieces of the output, produced by the SAS software being validated. Sometimes the same algorithm used by the procedure, action, or other component being checked is coded into SAS/IML, but occasionally a different algorithm is used if the two algorithms are expected to produce the same results. Replication of results via SAS/IML or DATA step code is the most reliable method for validation because it is an independent and verified confirmation of the numerical output.

## Comparing to Similar Results in Other Algorithms in SAS®

Many SAS Components produce equivalent output, such as parameter estimates, covariances, or solutions to modeling problems. Similar output produced by new components is validated by comparing it to these previously validated results. Similarly, when a macro program exists that produces some of the same results as the new component, it can be used for validation.

## Running Simulation Studies

Simulation studies may be performed in instances where closed-form solutions do not exist or used as checks when computations are extremely time and memory intensive.

## Verifying against Published Results

Comparison against other software vendors' applications is sometimes made, especially in those cases where the vendor's application is highly regarded as producing quality results. Comparison is also sometimes made versus results published in books or journal articles. Note that matching another vendor's results or just verifying a match versus a published result is not usually considered sufficient validation.

## Comparing against Open-Source Software

Numerical validation against open-source software is sometimes performed to ensure that the results are as accurate or more accurate than comparable open-source software and that the performance is at least comparable, if not better. Our numerical validation test suites include a representative set of test scenarios running comparable algorithms for accuracy comparisons. Numerical results are compared for accuracy when the true or optimal model or solution is known.

## Validation of SAS® Components that Consume Open-Source Technology

SAS software in some cases leverages open-source technology in its components. To make sure that the components that use these open-source routines work seamlessly, OSAS aims to re-create the same results across a variety of scenarios as those derived from directly applying an equivalent wrapper directly to the open-source technology. Validating by simultaneously using replication and comparison ensures that SAS components are leveraging open-source technology effectively.

## Completing Consistency Checks within the Component

Certain consistency checks are performed to help validate results. Here are some examples:

- Checking that results with a WEIGHT variable that has all weight values equal to 1 are identical to results obtained without using the WEIGHT variable.
- Verifying that results with a FREQ variable match results when not using the FREQ variable but instead repeating each observation by the value of the FREQ variable.
- Verifying that results with a BY statement match those obtained for each value of the BY variable analyzed individually.

An example of our validation techniques for the REG procedure is included below.

## Example of Validation Techniques

While many analyses and results require complex code to validate, the following simple example is used to provide a basic illustration of some of the validation techniques that are performed at SAS. The test case verifies the results from the REG procedure by comparing them to a classic textbook analysis. The data comes from Neter, Wasserman, and Kutner, (1990), and the test case verifies the ANOVA table, the Fit Statistics, and the Parameter Estimates table that PROC REG produces.

The data consists of sales information from 15 marketing districts, and PROC REG fits a multiple regression model. Besides comparing the results to the textbook results, this example also illustrates validation using the IML procedure and cross validation using the GLM procedure. Finally, the example concludes by illustrating some basic consistency checks.

```

data Zarthan_Company;
input sales target_population discretionary_income @@;
datalines;
162 274 2450 120 180 3254 223 375 3802 131 205 2838
67 86 2347 169 265 3782 81 98 3008 192 330 2450
116 195 2137 55 53 2560 252 430 4020 232 372 4427
144 236 2660 103 157 2088 212 370 2605
;

ods listing close;
ods rtf file='Zarthan.rtf';
ods select ANOVA FitStatistics ParameterEstimates;
proc reg data=Zarthan_Company;
ods output ANOVA=reg_ANOVA
FitStatistics=reg_FitStatistics
ParameterEstimates=reg_ParameterEstimates;
model sales = target_population discretionary_income;
run;
ods rtf close;

```

## PROC REG Results

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	53845	26922	5679.47	<.0001
Error	12	56.88357	4.74030		
Corrected Total	14	53902			

Root MSE	2.17722	R-Square	0.9989
Dependent Mean	150.60000	Adj R-Sq	0.9988
Coeff Var	1.44570		

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr >  t
Intercept	1	3.45261	2.43065	1.42	0.1809
target_population	1	0.49600	0.00605	81.92	<.0001
discretionary_income	1	0.00920	0.00096811	9.50	<.0001

## Comparison to Textbook

The results from PROC REG can be compared to those given in the textbook to verify that there are no discrepancies. For this example, the following quantities are reported in the textbook:

### ANOVA Results

Statistic	Result	Page
SS Model	53,844.716	256
SS Error	56.884	256
SS Total	53,901.600	256
df Model	2	256
df Error	12	256
df Total	14	256
MS Model	26,922.358	256
MS Error	4.740	256
F*	5,680	257
p-value	< .001	257

\* Note that PROC REG reports as 5,679.47. The discrepancy is due to the textbook example rounding the quantities involved in the ratio before the ratio is computed.

### Fit Statistics

Statistic	Result	Page
R-Squared	0.9989	257

### Parameter Estimates

Statistic	Result	Page
Beta: Intercept	3.4526127900	252
Beta: target population	0.4960049761	252
Beta: disc. Income	0.009199080867	252
Std. Error: target pop.	0.006054	258
Std. Error: disc. Income	0.0009681	257

## Direct Validation

SAS/IML can be used to compute the corresponding quantities from the PROC REG output:

```
proc iml;
  use Zarthan_Company;
  read all var _all_ into data;
  y=data[,1];
  n=nrow(y);
  x=j(n,1,1)||data[,2:3];
  p=ncol(x);
  beta=inv(x`*x)*x`*y;
  yhat=x*beta;
  resid=y-yhat;
  sse=ssq(resid);
  dfe=nrow(x)-ncol(x);
  mse=sse/dfe;
  cssy=ssq(y-y[+]/n);
  rsquare=(cssy-sse)/cssy;
  stdbeta=sqrt(vecdiag(inv(x`*x))*mse);
  t=beta/stdbeta;
  df=j(nrow(t),1,1);
  t_prob=1-cdf('F',t##2,df,dfe);
  dft=n-1;
  dfm=dft-dfe;
  ssm=cssy-sse;
  msm=ssm/dfm;
  F=msm/mse;
  F_prob=1-cdf('F',F,dfm,dfe);
  root_mse=sqrt(mse);
  mean_y=y[+]/n;
  coeff_var=(root_mse/mean_y)*100;
  adj_r=1-((n-1)#(1-rsquare))/(n-p);

  * dependent variable;
  * sample size;
  * x matrix, augmented for intercept;
  * number of parameters;
  * parameter estimates;
  * predicted values;
  * residuals;
  * Sum of Squares for Error;
  * error degrees of freedom;
  * Mean Square Error;
  * Corrected Total Sum of Squares;
  * R-Square;
  * Std error of estimates;
  * parameter t-tests;
  * parameter degrees of freedom;
  * p-values for t-tests;
  * corrected total df;
  * Model degrees of freedom;
  * Sum of Squares for Model;
  * Mean Square Model;
  * F statistic;
  * p-value for F statistic;
  * Root MSE;
  * Dependent Mean;
  * Coefficient of Variation;
  * Adjusted R-Square;

  * create matrices of the corresponding REG tables;
  anova_table=(dfm//dfe//dft)|| (ssm//sse//cssy)|| (msm//mse//{._})
              ||(F//{._ .}_`)|| (F_prob//{._ .}_`);
  fit_statistics=(root_mse//mean_y//coeff_var)|| (rsquare//adj_r//{0});
  parameter_estimates=df||beta||stdbeta||t||t_prob;

  * create data sets of these matrices to be used with the COMPARE procedure;
  create iml_anova(label='Analysis of Variance' )
    from anova_table[colname={df ss ms fvalue probf}];
  append from anova_table;
  create iml_fitstatistics(label='Fit Statistics' )
    from fit_statistics[colname={nvalue1 nvalue2}];
  append from fit_statistics;
  create iml_parameterestimates(label='Parameter Estimates' )
    from parameter_estimates[colname={df estimate stderr tvalue
    probt}];
  append from parameter_estimates;
quit;
```

```

* print the SAS/IML validation results for a visual scan;
proc print data=iml_anova noobs; run;
proc print data=iml_fitstatistics noobs; run;
proc print data=iml_parameterestimates noobs; run;

```

### SAS/IML Validation Results

#### ANOVA Table

DF	SS	MS	FVALUE	PROBF
2	53844.72	26922.36	5679.47	0
12	56.88	4.74	–	–
14	53901.60	–	–	–

#### Fit Statistics Table

NVALUE1	NVALUE2
2.177	0.99894
150.600	0.99877
1.446	0.00000

#### Parameter Estimates Table

DF	ESTIMATE	STDERR	TVALUE	PROBT
1	3.45261	2.43065	1.4204	0.18094
1	0.49600	0.00605	81.9242	0.00000
1	0.00920	0.00097	9.5021	0.00000

\* compare via PROC COMPARE the PROC REG results to the SAS/IML validation results

```

proc compare data=reg_anova compare=iml_anova
  error briefsummary note method=relative(1) criterion=1e-6;
  attrib _all_ format = label = '';
  var df ss ms fvalue probf;
run;

```

```

proc compare data=reg_fitstatistics compare=iml_fitstatistics
  error briefsummary note method=relative(1) criterion=1e-6;
  attrib _all_ format = label = '';
  var nvalue1 nvalue2;
run;

```

```

proc compare data=reg_parameterestimates compare=iml_parameterestimates
  error briefsummary note method=relative(1) criterion=1e-6;
  attrib _all_ format = label = '';
  var df estimate stderr tvalue probt;
run;

```

## PROC COMPARE Results

The COMPARE Procedure  
Comparison of WORK.REG\_ANOVA with WORK.IML\_ANOVA  
(Method=RELATIVE(1), Criterion=0.000001)

NOTE: All values compared are within the equality criterion used.

NOTE: One or both of the data sets WORK.REG\_ANOVA and WORK.IML\_ANOVA contain variables not in the other. However, all comparisons are equal for the variables specified.

The COMPARE Procedure  
Comparison of WORK.REG\_FITSTATISTICS with WORK.IML\_FITSTATISTICS  
(Method=RELATIVE(1), Criterion=0.000001)

NOTE: All values compared are within the equality criterion used.

NOTE: One or both of the data sets WORK.REG\_FITSTATISTICS and WORK.IML\_FITSTATISTICS contain variables not in the other. However, all comparisons are equal for the variables specified.

The COMPARE Procedure  
Comparison of WORK.REG\_PARAMETERESTIMATES with WORK.IML\_PARAMETERESTIMATES  
(Method=RELATIVE(1), Criterion=0.000001)

NOTE: All values compared are within the equality criterion used.

NOTE: One or both of the data sets WORK.REG\_PARAMETERESTIMATES and WORK.IML\_PARAMETERESTIMATES contain variables not in the other. However, all comparisons are equal for the variables specified.

## Comparison to Other SAS® Procedures

The PROC REG results can be compared to PROC GLM output:

\* run the corresponding model with PROC GLM;

```
proc glm data=Zarthan_Company;
  ods output OverallANOVA=glm_ANOVA(label='Analysis of Variance')
             FitStatistics=glm_FitStatistics
             ParameterEstimates=
             glm_ParameterEstimates(label='Parameter Estimates');
  model sales = target_population discretionary_income/solution;
run;
```

```

* compare the REG results to the GLM results;
proc compare data=reg_anova compare=glm_anova
  error briefsummary note method=relative(1) criterion=1e-6;
  attrib _all_ format = label = '';
  var df ss ms fvalue probf;
run;
proc compare data=reg_parameterestimates
  compare=glm_parameterestimates
  error briefsummary note method=relative(1) criterion=1e-6;
  attrib _all_ format = label = '';
  var estimate stderr tvalue probt;
run;

```

\* visually compare the statistics that correspond in Fit Statistics because the two tables have a different structure:

```

proc print data=reg_FitStatistics; run;
proc print data=glm_FitStatistics; run;

```

## PROC REG Results

Obs	Model	Dependent	Label1	cValue1	nValue1	Label2	cValue2	nValue2
1	MODEL1	sales	Root MSE	2.17722	2.177222	R-Square	0.9989	0.998945
2	MODEL1	sales	Dependent Mean	150.60000	150.600000	Adj R-Sq	0.9988	0.998769
3	MODEL1	sales	Coeff Var	1.44570	1.445699			0

## PROC GLM Results

Obs	Dependent	RSquare	CV	RootMSE	DepMean
1	sales	0.998945	1.445699	2.177222	150.6000

## Consistency Checking

A simple check of the WEIGHT statement can be performed. Note that complete testing for WEIGHT would include tests where the weight values are not all equal to 1, with SAS/IML validation performed.

\* add a weight variable to the data set, with values all equal to 1;

```

data check_weight; set Zarthan_Company;
  weight=1;
run;

```

\* run PROC REG with the weight variable;

```

proc reg data=check_weight;
  ods output ANOVA=reg_ANOVA_weight
             FitStatistics=reg_FitStatistics_weight
             ParameterEstimates=reg_ParameterEstimates_weight;
  model sales = target_population discretionary_income;
  weight weight;
run;

```

\* compare the results from the REG run without a weight to the REG run with weights all equal to 1;

```
proc compare data=reg_anova compare=reg_anova_weight
  error briefsummary note method=relative(1) criterion=1e-6;
  attrib _all_ format = label = '';
run;
```

```
proc compare data=reg_fitstatistics
  compare=reg_fitstatistics_weight
  error briefsummary note method=relative(1) criterion=1e-6;
  attrib _all_ format = label = '';
run;
```

```
proc compare data=reg_parameterestimates
  compare=reg_parameterestimates_weight
  error briefsummary note method=relative(1) criterion=1e-6;
  attrib _all_ format = label = '';
run;
```

## PROC COMPARE Results:

The COMPARE Procedure  
Comparison of WORK.REG\_ANOVA with WORK.REG\_ANOVA\_WEIGHT  
(Method=RELATIVE(1), Criterion=0.000001)

NOTE: All values compared are within the equality criterion used.

NOTE: The data sets WORK.REG\_ANOVA and WORK.REG\_ANOVA\_WEIGHT compare equal.

The COMPARE Procedure  
Comparison of WORK.REG\_FITSTATISTICS with WORK.REG\_FITSTATISTICS\_WEIGHT  
(Method=RELATIVE(1), Criterion=0.000001)

NOTE: All values compared are within the equality criterion used.

NOTE: The data sets WORK.REG\_FITSTATISTICS and WORK.REG\_FITSTATISTICS\_WEIGHT compare equal.

The COMPARE Procedure  
Comparison of WORK.REG\_PARAMETERESTIMATES with  
WORK.REG\_PARAMETERESTIMATES\_WEIGHT  
(Method=RELATIVE(1), Criterion=0.000001)

NOTE: All values compared are within the equality criterion used.

NOTE: The data sets WORK.REG\_PARAMETERESTIMATES and  
WORK.REG\_PARAMETERESTIMATES\_WEIGHT compare equal.

## References

National Institute of Standards and Technology 2007. "NIST STRD Background Information." Accessed November 6, 2015. See <http://www.itl.nist.gov/div898/strd/general/bkgground.html>.

Neter J., W. Wasserman, and M. H. Kutner. 1990. Applied Linear Statistical Models, Third Edition. Boston, MA: Irwin.

### Release Information

The version of this paper is January 2022.

Unless otherwise indicated, this document relates only to SAS 9.4, SAS Viya, and the products that are available with SAS 9.4 and SAS Viya. It also relates to services from the date of this paper forward. Quality processes are continually evolving. Therefore, SAS reserves the right to modify the processes described in this document at any time. If you are using SAS 9.4 and SAS Viya and have questions about processes in those releases, send email to [qualitypaper@sas.com](mailto:qualitypaper@sas.com).

Learn more about SAS Solutions at [sas.com](https://sas.com).

