# Advantages of PROC SCORE

Mark Tabladillo, Ph.D., Atlanta, GA

## ABSTRACT

The SCORE procedure multiplies values from two SAS® data sets, one containing coefficients (for example, factor-scoring coefficients or regression coefficients) and the other containing raw data to be scored using the coefficients from the first data set. The result of this multiplication is a SAS data set containing linear combinations of the coefficients and the raw data values. Version 9 SAS/STAT® extends the concept by making SCORE a statement on the LOGISTIC procedure.

This short presentation outlines why this procedure has practical automated advantages over creating coefficients another way. Examples using PROC FACTOR and PROC LOGISTIC are presented.

## INTRODUCTION

From the SAS Documentation (SAS Institute, 2002):

> The SCORE procedure multiplies values from two SAS data sets, one containing coefficients (for example, factor-scoring coefficients or regression coefficients) and the other containing raw data to be scored using the coefficients from the first data set. The result of this multiplication is a SAS data set containing linear combinations of the coefficients and the raw data values.

Regression coefficients (sometimes known as beta weights) are specific weights applied to individual variables as the result of a regression analysis; there a total of N weights, where N is the number of variables and combinations of variables (including transformations) in the model. This type of output comes from both PROC GLM and PROC LOGISTIC and PROC REG.

Factor scores are used in factor analysis, and represent the variable weights in each factor. This type of output comes from both PROC FACTOR and PROC PRINCOMP. There are a total of PxQ possible factor scores, where P is the number of variables considered and Q is the number of factors. Sometimes analysts consider only using unit-weight factor scores, in which case each position in the PxQ matrix has been methodically assigned one or zero. For example, if orthogonality is a desired outcome, then a variable can only have a factor score for a single factor only; nonorthogonal techniques allow a variable to contribute to multiple factors.

Both regression and factor analysis share the common goal of attempting to discover how variables contribute to an outcome. Regression specifies an outcome variable (included in the model) while factor analysis specifies perhaps several theoretical factors validated by data.

PROC SCORE is therefore of most use to people doing regression or factor analysis.

## BEFORE PROC SCORE

Before PROC SCORE, it certainly is possible to take the output from a PROC LOGISTIC or PROC FACTOR and use a data step to apply these coefficients to one or more datasets. Whether PROC SCORE is used or not, it's essential to how to specify output for either regression coefficients or factor scores, and also what types of standardized variables SAS/STAT will create. These output datasets are similar but not identical among procedures, so each procedure's output is both unique and documented in the SAS/STAT documentation.

The next two sections present a before and after example using first factor analysis (with PROC FACTOR) and then logistic regression analysis (with PROC LOGISTIC).

## BEFORE AND AFTER WITH PROC FACTOR

The SAS documentation presents a sample dataset of school grades, with a character ID value called type, and three grade values for English, Math, and Biology. These data are stored in the work.schools dataset.

The sample code then processes the data through PROC FACTOR, and determines a single factor with the standardized scores for the three component variables, the English, Math and Biology grades. Standardization calculations are made easier because PROC FACTOR outputs the mean and standard deviation (along with the sample size) for all the variables run through the model. The new factor is a conceptual index, whose value can be applied to the original dataset, and whose output can be comparatively grouped (in this case, the groupings would be by type).

The code is as follows:

```
proc factor data=schools score
outstat=scores noprint;
      var english math biology;
      run;
proc score data=schools score=scores
out=new;
      var english math biology;
      id type;
      run;
proc sort data=new;
      by type;
      run;
proc means data=new;
      by type;
      run;
```

The output of PROC MEANS (based on the normal distribution and independence of factor scores) shows no statistically significant differences among the three types. The advantage of PROC SCORE includes automatically applying standardized coefficients to the SCHOOLS dataset and putting the factor results (with the ID variable TYPE) into a new dataset.

Before PROC SCORE, the alternative could be a combination of PROC SORTs and merging, and perhaps PROC TRANSPOSE. There are any number of ways to alternatively code what PROC SCORE does, but the time savings are evident with multiple factors. In the above example, PROC FACTOR will output what the results are of the factor analysis, so the PROC SCORE simply removes the need to understand the unique data structure created by the OUTSTAT option.

## BEFORE AND AFTER WITH PROC LOGISTIC

The SAS documentation presents a sample dataset of crop data, with a ID value called CROP, and four crop variables, nominally named x1 through x4. These data are stored in the work.crops dataset.

The sample code then processes the data through PROC LOGISTIC, and determines a model where the outcome variable is the crop type, and the input variables are x1 through x4. The selected link is glogit (generalized logit function). Scores for PROC LOGISTIC (with the OUTEST option) are in a single observation with intercept values and beta weights for each type of crop.

Using the PROC SCORE statement, combined with a quick reading of the online documentation, you would believe that the code would be as follows:

```
proc logistic data=Crops
outest=estimates
outmodel=sasuser.CropModel;
      model Crop=x1-x4 / link=glogit;
      run;
proc score data=Crops score=estimates
out=Score1;
      var x1-x4;
      id Crop;
      run;
proc print data=score1; run;
```

However, the OUTEST from PROC LOGISTIC does NOT match what PROC SCORE is expecting, and the above code would therefore not work. In version 8 and before, it used to take more work to apply scoring coefficients as compared with PROC REG or PROC FACTOR. Specifically, there are no x1-x4 variables on the OUTEST dataset, and instead the variable names are as follows:

_LINK_
_TYPE_
_STATUS_
_NAME_
Intercept_Clover
Intercept_Corn
Intercept_Cotton
Intercept_Soybeans
x1_Clover
x1_Corn
x1_Cotton
x1_Soybeans
x2_Clover
x2_Corn
x2_Cotton
x2_Soybeans
x3_Clover
x3_Corn
x3_Cotton
x3_Soybeans
x4_Clover
x4_Corn
x4_Cotton
x4_Soybeans
_LNLIKE_

SAS version 9 adds the SCORE statement for PROC LOGISTIC, and the documentation reads as follows (SAS Institute, 2002):

> The SCORE statement enables you to score new data sets and output the scored values and, optionally, the corresponding confidence limits into a SAS data set.

The code should therefore be as follows:

```
proc logistic data=Crops
outest=estimates
outmodel=sasuser.CropModel;
      model Crop=x1-x4 / link=glogit;
      score data=Crops out=Score2;
      run;
proc print data=score2; run;
```

In other words, the functionality of PROC SCORE has been encapsulated into PROC LOGISTIC. Perhaps in the future, SAS may similarly encapsulate this functionality into PROC REG and PROC FACTOR.

As with PROC FACTOR, the PROC LOGISTIC example removes the need to understand the intermediate scoring dataset made with the OUTEST option, and instead the program simply outputs a new dataset with scores called SCORE2, and places the logistic model into a dataset called sasuser.CropModel.

## CONCLUSION

PROC SCORE provides some automated advantages for applying scoring coefficients, and is especially suited well for partnering with PROC REG and PROC FACTOR. The advantages have led to making SCORE a statement in PROC LOGISTIC.

## REFERENCES

SAS Institute Inc. (2002), *SAS OnlineDoc 9*, Cary, NC: SAS Institute, Inc.

## TRADEMARK CITATION

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Mark Tabladillo
Email: marktab@marktab.com
Web: http://www.marktab.com/