SAS/STAT® Software: Changes and Enhancements, Release 8.1

The Power to Know™

SAS Publishing

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/STAT® Software: Changes and Enhancements, Release 8.1*, Cary, NC: SAS Institute Inc., 2000

SAS/STAT® Software: Changes and Enhancements, Release 8.1

ISBN 1-58025-655-4

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, May 2000

Table of Contents

Chapter 1. The CATMOD Procedure 1
Chapter 2. The FACTOR Procedure 15
Chapter 3. The FREQ Procedure 31
Chapter 4. The GAM Procedure 35
Chapter 5. The GENMOD Procedure 69
Chapter 6. The LOESS Procedure 73
Chapter 7. The LOGISTIC Procedure 89
Chapter 8. The MIXED Procedure 109
Chapter 9. The MODECLUS Procedure 113
Chapter 10. The MULTTEST Procedure 117
Chapter 11. The NLMIXED Procedure 121
Chapter 12. The PHREG Procedure 125
Chapter 13. The SURVEYMEANS Procedure 129
Subject Index 151
Syntax Index 153

Chapter 1
The CATMOD Procedure

Chapter Table of Contents

OVERVIEW 3

SYNTAX 3
MODEL Statement 3

DETAILS 6
Missing Values 6
Zero Frequencies 6
Computational Formulas 6

EXAMPLES 7
Example 1.1 Log-Linear Independence Model with Structural and Sampling Zeros 7
Example 1.2 Identifying an Inappropriate Model 12

REFERENCES 14

Chapter 1
The CATMOD Procedure

Overview

The CATMOD procedure now offers the iterative proportional fitting (IPF) algorithm for fitting hierarchical log-linear models with one population (that is, there are no independent variables and no population variables). The advantage of the IPF algorithm is that you can obtain the log likelihood, G^2, and the predicted cell counts without performing expensive parameter estimation and covariance computations. To request IPF fitting, you specify the ML=IPF option. Several options for controlling the convergence criterion are available, as well as options for controlling how the degrees of freedom for G^2 are computed.

The new MISSING= option specifies whether a missing cell is treated as a sampling or structural zero. The new ZERO= option specifies whether a non-missing cell with zero weight is treated as a sampling or structural zero.

Syntax

MODEL Statement

The following options have been added to the MODEL statement.

ML $<$ **= NR | IPF** $<$ **(** *options* **)** $>>$
computes maximum likelihood estimates (MLE) using either a Newton-Raphson algorithm (NR) or an iterative proportional fitting algorithm (IPF).

The option ML=NR (or simply ML) is available when you use logits or generalized logits, and is the default for generalized logits.

The ML=IPF option is available for fitting a hierarchical log-linear model with one population (that is, there are no independent variables and no population variables). The use of bar notation to describe the log-linear effects guarantees that the model is *hierarchical*, meaning that the presence of any interaction term in the model requires the presence of all its lower-order terms. The underlying table in an IPF analysis is the cross-classification of the observed levels of all dependent variables. If the table is *incomplete*, which means that a zero or missing entry occurs in at least one cell, then all missing cells and all cells with zero weight are treated as structural zeros by default. This behavior can be modified with the ZERO= and MISSING= options in the MODEL statement.

You can control the convergence of the two algorithms with the EPSILON= and MAXITER= options in the MODEL statement.

Note: The RESTRICT statement is not available with the ML=IPF option.

You can specify the following *options* within parentheses after the ML=IPF option.

CONV=*keyword*

CONVCRIT=*keyword* specifies the method that determines when convergence of the IPF algorithm occurs. You can specify one of the following *keywords*:

CELL termination requires the maximum absolute difference between consecutive cell estimates to be less than 0.001 (or the value of the EPSILON= option, if specified).

LOGL termination requires the relative difference between consecutive estimates of the log likelihood to be less than 1E-8 (or the value of the EPSILON= option, if specified). This is the default.

MARGIN termination requires the maximum absolute difference between consecutive margin estimates to be less than 0.001 (or the value of the EPSILON= option, if specified).

DF=*keyword* specifies the method used to compute the degrees of freedom for the goodness of fit G^2 test (labeled "Likelihood Ratio" in the "Estimates" table).

For a *complete* table (a table having nonzero entries in every cell), the degrees of freedom are calculated as the number of cells in the table (n_c) minus the number of independent parameters specified in the model (n_p). For incomplete tables, these degrees of freedom may be adjusted by the number of fitted zeros (n_z, which includes the number of structural zeros) and the number of nonestimable parameters due to the zeros (n_n). If you are analyzing an incomplete table, you should verify that the degrees of freedom are correct.

You can specify one of the following *keywords*:

UNADJ computes the unadjusted degrees of freedom as $n_c - n_p$. These are the same degrees of freedom you would get if all cells in the table were positive.

ADJ computes the degrees of freedom as $(n_c - n_p) - (n_z - n_n)$ (Bishop, Fienberg, and Holland 1975), which adjusts for fitted zeros and nonestimable parameters. This is the default, and for complete tables gives the same results as the UNADJ option.

ADJEST computes the degrees of freedom as $(n_c - n_p) - n_z$, which adjusts for fitted zeros only. This gives a lower bound on the true degrees of freedom.

PARM computes parameter estimates, generates the "ANOVA," "Parameter Estimates," and "Predicted Values of Response Functions" tables, and includes the predicted standard errors in the "Predicted Values of Frequencies" and "Predicted Values of Probabilities" tables.

When you specify the PARM option, the algorithm used to obtain the maximum likelihood parameter estimates is weighted least squares on the IPF-predicted frequencies. This algorithm can be much faster than the Newton-Raphson algorithm used if you just specify the ML=NR option. In the resulting ANOVA table, the likelihood ratio is computed from the initial IPF fit while the degrees of freedom are generated from the WLS analysis; you can override this with

the DF= option. The initial response function, which the WLS method usually computes from the raw data, is computed from the IPF fitted frequencies.

If there are any zero marginals in the configurations that define the model, then predicted cell frequencies of zero will result and WLS cannot be used to compute the estimates. In this case, PROC CATMOD automatically changes the algorithm from ML=IPF to ML=NR and prints a note in the log.

MISS=*keyword* **|** *value*
MISSING=*keyword* **|** *value*

specifies whether a missing cell is treated as a sampling or structural zero.

Structural zero cells are removed from the analysis since their expected values are zero, while sampling zero cells may have nonzero expected value. For a single population, the missing cells are treated as structural zeros by default. For multiple populations, as long as some population has a nonzero count for a given population and response profile, the missing values are treated as sampling zeros by default.

The following table displays the available *keywords* and summarizes how PROC CATMOD treats missing values for one or more populations.

MISSING=	One Population	Multiple Populations
STRUCTURAL (default)	structural zeros	sampling zeros
SAMP | SAMPLING	sampling zeros	sampling zeros
value	sets missing weights and cells to *value*	sets missing weights and cells to *value*

ZERO=*keyword* **|** *value*
ZEROS=*keyword* **|** *value*
ZEROES=*keyword* **|** *value*

specifies whether a non-missing cell with zero weight in the data set is treated as a sampling or structural zero.

Structural zero cells are removed from the analysis since their expected values are zero, while sampling zero cells have nonzero expected value. For a single population, the zero cells are treated as structural zeros by default; with multiple populations, as long as some population has a nonzero count for a given population and response profile, the zeros are treated as sampling zeros by default.

The following table displays the available *keywords* and summarizes how PROC CATMOD treats zeros for one or more populations.

ZERO=	One Population	Multiple Populations
STRUCTURAL (default)	structural zeros	sampling zeros
SAMP | SAMPLING	sampling zeros	sampling zeros
value	sets zero weights to *value*	sets zero weights to *value*

Details

Missing Values

Observations with missing values for any variable listed in the MODEL or POPULATION statement are omitted from the analysis.

If the WEIGHT variable for an observation has a missing value, the observation is by default omitted from the analysis. You can modify this behavior by specifying the MISSING= option in the MODEL statement. The option MISSING=*value* sets all missing weights to *value* and all missing cells to *value*. The option MISSING=SAMPLING causes all missing cells in a contingency table to be treated as sampling zeros.

Zero Frequencies

There are two types of zero cells in a contingency table: structural and sampling. A structural zero cell has an expected value of zero, while a sampling zero cell may have nonzero expected value.

For any log-linear model analysis, PROC CATMOD creates response profiles only for the observed profiles. Thus, for an analysis with one population (the usual case), the resulting contingency table does not contain zero or missing cells, which means that these cells are treated as structural zeros. However, for a WLS or ML=NR analysis on more than one population, a zero or missing cell in the body of the contingency table is treated as a sampling zero (as long as some population has a nonzero count for that profile).

If you want to treat zero frequencies as sampling zeros, you can specify the ZERO=SAMPLING and MISSING=SAMPLING options in the MODEL statement. Alternatively, you can include a statement in the DATA step creating your SAS data set that changes each zero to a very small number (such as 1E-20).

Refer to Bishop, Fienberg, and Holland (1975) and Christensen (1997) for a discussion of the issues. See Example 1.1 on page 7 for an illustration of a log-linear model analysis of data that contain both structural and sampling zeros.

Computational Formulas

The algorithm used for iterative proportional fitting is described in Haberman (1972), Bishop, Fienberg, and Holland (1975), and Agresti (1990). To illustrate the method, consider the observed three-dimensional table $\{n_{ijk}\}$ for the variables X, Y, and Z. The statements

```
proc catmod;
   model X*Y*Z = _response_ / ml=ipf;
   loglin X|Y|Z@2;
run;
```

request that PROC CATMOD use IPF to fit the hierarchical model

$$\log(m_{ijk}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \lambda_{ij}^{XY} + \lambda_{ik}^{XZ} + \lambda_{jk}^{YZ}$$

where $\{m_{ijk}\}$ are the expected frequencies of the cells in the contingency table.

PROC CATMOD begins with a table of initial cell estimates $\{\hat{m}_{ijk}^{(0)}\}$ that are produced by setting the n_{sz} structural zero cells to 0 and all other cells to $n/(n_c - n_{sz})$, where n is the total weight of the table and n_c is the total number of cells in the table. It then iteratively adjusts the estimates at step $s-1$, $\{\hat{m}_{ijk}^{(s-1)}\}$, to the observed marginal tables specified in the model by stepping through the following three-stage process to produce the estimates at step s:

$$\hat{m}_{ijk}^{(s_1)} = \hat{m}_{ijk}^{(s-1)} \frac{n_{ij\cdot}}{\hat{m}_{ij\cdot}^{(s-1)}} \quad , \quad \hat{m}_{ijk}^{(s_2)} = \hat{m}_{ijk}^{(s_1)} \frac{n_{i\cdot k}}{\hat{m}_{i\cdot k}^{(s_1)}} \quad , \quad \hat{m}_{ijk}^{(s)} = \hat{m}_{ijk}^{(s_2)} \frac{n_{\cdot jk}}{\hat{m}_{\cdot jk}^{(s_2)}}$$

where the superscripts (s_1) and (s_2) indicate the two intermediate tables in the process, and the subscript "$\cdot$" indicates summation over the missing subscript. The log likelihood l_s is estimated at each step s by

$$l_s = \sum_{i,j,k} n_{ijk} \log\left(\frac{\hat{m}_{ijk}^{(s)}}{n}\right)$$

When the function $|(l_{s-1} - l_s)/l_{s-1}|$ is less than 1E$-$8, the iterations terminate. You can change the comparison value with the EPSILON= option, and you can change the convergence criterion with the CONV= option. The option CONV=CELL uses the maximum absolute cell difference

$$\max_{i,j,k} |d_{ijk}| < 0.001 \quad \text{where} \quad d_{ijk} = \hat{m}_{ijk}^{(s-1)} - \hat{m}_{ijk}^{(s)}$$

as the criterion while the option CONV=MARGIN uses the maximum absolute difference of the margins

$$\max\left(\max_{i,j} |d_{ij\cdot}|, \max_{i,k} |d_{i\cdot k}|, \max_{j,k} |d_{\cdot jk}|\right) < 0.001$$

Examples

Example 1.1. Log-Linear Independence Model with Structural and Sampling Zeros

This example illustrates a log-linear model of independence, using data that contain structural zero frequencies as well as sampling (random) zero frequencies.

In a population of six squirrel monkeys, the joint distribution of genital display with respect to active or passive role was observed. The data are from Fienberg (1980, Table 8-2). The following DATA step creates the SAS data set Display:

```
title 'Behavior of Squirrel Monkeys';
data Display;
   input Active $ Passive $ wt @@;
   datalines;
r r  .   r s  1   r t  5   r u  8   r v  9   r w  0
s r 29   s s  .   s t 14   s u 46   s v  4   s w  0
t r  0   t s  0   t t  .   t u  0   t v  0   t w  0
u r  2   u s  3   u t  1   u u  .   u v 38   u w  2
v r  0   v s  0   v t  0   v u  0   v v  .   v w  1
w r  9   w s 25   w t  4   w u  6   w v 13   w w  .
;
```

In this data set, since a monkey cannot have both active and passive roles in an interaction, the values on the diagonal are structural zeros. Any off-diagonal zeros are sampling zeros. Since there are two types of zeros in this data set, missing values are placed on the diagonal to represent the structural zeros.

Suppose you're interested in studying the independence of the active and passive roles. Since the diagonal cells are structural zeros, you are actually fitting a *quasi*-independence model; refer to Agresti (1990) for more information. Since monkey 't' never takes the active role, the frequencies predicted by an independence model for these cells are zero; these cells are removed from the analysis with the WHERE clause.

The following statements produce the analysis that treats the missing values on the diagonals as structural zeros (since the MISSING=STRUCTURAL option is the default for one population). The ZERO=SAMPLING option treats the remaining zeros as sampling zeros.

```
proc catmod data=Display;
   weight wt;
   where Active ^= 't';
   model Active*Passive=_response_
         / ml=ipf(parm) zero=sampling;
   loglin Active Passive;
run;
```

Output 1.1.1. Data Summary and Population Profile

```
                Behavior of Squirrel Monkeys

                    The CATMOD Procedure

                        Data Summary

Response           Active*Passive     Response Levels   25
Weight Variable    wt                 Populations        1
Data Set           DISPLAY            Total Frequency  220
Frequency Missing  0                  Observations      25


                    Population Profiles

                   Sample    Sample Size
                   ---------------------
                       1             220
```

The response profiles, shown in Output 1.1.2, include the off-diagonal zero cells because of the ZERO=SAMPLING option.

Output 1.1.2. Response Profiles

```
                    Response Profiles

                  Response    Active    Passive
                  ----------------------------
                      1       r         s
                      2       r         t
                      3       r         u
                      4       r         v
                      5       r         w
                      6       s         r
                      7       s         t
                      8       s         u
                      9       s         v
                     10       s         w
                     11       u         r
                     12       u         s
                     13       u         t
                     14       u         v
                     15       u         w
                     16       v         r
                     17       v         s
                     18       v         t
                     19       v         u
                     20       v         w
                     21       w         r
                     22       w         s
                     23       w         t
                     24       w         u
                     25       w         v
```

Because the PARM option is specified, a weighted least squares analysis is performed on the IPF fitted data and the _Response_ Matrix is displayed (Output 1.1.3); this table can be suppressed with the NORESPONSE option.

Output 1.1.3. _Response_ Matrix

Response Matrix

	1	2	3	4	5	6	7	8	9
1	1	0	0	0	0	1	0	0	0
2	1	0	0	0	0	0	1	0	0
3	1	0	0	0	0	0	0	1	0
4	1	0	0	0	0	0	0	0	1
5	1	0	0	0	-1	-1	-1	-1	-1
6	0	1	0	0	1	0	0	0	0
7	0	1	0	0	0	0	1	0	0
8	0	1	0	0	0	0	0	1	0
9	0	1	0	0	0	0	0	0	1
10	0	1	0	0	-1	-1	-1	-1	-1
11	0	0	1	0	1	0	0	0	0
12	0	0	1	0	0	1	0	0	0
13	0	0	1	0	0	0	1	0	0
14	0	0	1	0	0	0	0	0	1
15	0	0	1	0	-1	-1	-1	-1	-1
16	0	0	0	1	1	0	0	0	0
17	0	0	0	1	0	1	0	0	0
18	0	0	0	1	0	0	1	0	0
19	0	0	0	1	0	0	0	1	0
20	0	0	0	1	-1	-1	-1	-1	-1
21	-1	-1	-1	-1	1	0	0	0	0
22	-1	-1	-1	-1	0	1	0	0	0
23	-1	-1	-1	-1	0	0	1	0	0
24	-1	-1	-1	-1	0	0	0	1	0
25	-1	-1	-1	-1	0	0	0	0	1

The iteration history displays the value of the log likelihood and the convergence criterion for the IPF method as discussed in the “Computational Formulas” section on page 6.

Output 1.1.4. Iteration History

Maximum Likelihood Analysis

Iteration	-2 Log Likelihood	Convergence Criterion
0	1201.5105	1.0000
1	1198.5669	0.002450
2	1198.5604	5.4468E-6
3	1198.5603	7.702E-8
4	1198.5603	1.6932E-9

The IPF algorithm converged.

The “Response Functions and Design Matrix” table (Output 1.1.5) is displayed when the PARM option is specified; this table can be suppressed with the NODESIGN option. The logits are computed from the IPF fitted values rather than the original data.

Output 1.1.5. Response Functions, Design Matrix

Response Functions and Design Matrix

Sample	Function Number	Response Function	Design Matrix 1	2	3	4	5	6	7	8	9
1	1	-0.97354	2	1	1	1	0	1	0	0	-1
	2	-1.72504	2	1	1	1	0	0	1	0	-1
	3	-0.52752	2	1	1	1	0	0	0	1	-1
	4	-0.73927	2	1	1	1	0	0	0	0	0
	5	-3.56052	2	1	1	1	-1	-1	-1	-1	-2
	6	0.32061	1	2	1	1	1	0	0	0	-1
	7	-0.29932	1	2	1	1	0	0	1	0	-1
	8	0.89820	1	2	1	1	0	0	0	1	-1
	9	0.68645	1	2	1	1	0	0	0	0	0
	10	-2.13480	1	2	1	1	-1	-1	-1	-1	-2
	11	-0.24152	1	1	2	1	1	0	0	0	-1
	12	-0.10995	1	1	2	1	0	1	0	0	-1
	13	-0.86145	1	1	2	1	0	0	1	0	-1
	14	0.12432	1	1	2	1	0	0	0	0	0
	15	-2.69693	1	1	2	1	-1	-1	-1	-1	-2
	16	-4.14787	1	1	1	2	1	0	0	0	-1
	17	-4.01631	1	1	1	2	0	1	0	0	-1
	18	-4.76780	1	1	1	2	0	0	1	0	-1
	19	-3.57029	1	1	1	2	0	0	0	1	-1
	20	-6.60328	1	1	1	2	-1	-1	-1	-1	-2
	21	-0.36584	0	0	0	0	1	0	0	0	-1
	22	-0.23427	0	0	0	0	0	1	0	0	-1
	23	-0.98577	0	0	0	0	0	0	1	0	-1
	24	0.21175	0	0	0	0	0	0	0	1	-1

The ANOVA table and the parameter estimates are a by-product of running WLS on the IPF-fitted values. Note that the likelihood ratio chi-square (goodness-of-fit G^2) in the ANOVA table is computed from the IPF routine; however, the degrees of freedom for G^2 are calculated through WLS. If the PARM option was not specified, then only the likelihood ratio test would be displayed.

Output 1.1.6. ANOVA

Maximum Likelihood Analysis of Variance

Source	DF	Chi-Square	Pr > ChiSq
Active	4	56.57	<.0001
Passive	5	47.94	<.0001
Likelihood Ratio	15	135.17	<.0001

Output 1.1.7. Parameter Estimates

Analysis of Maximum Likelihood Estimates

Effect	Parameter	Estimate	Standard Error	Chi-Square	Pr > ChiSq
Active	1	0.00284	0.2660	0.00	0.9915
	2	1.4286	0.2277	39.35	<.0001
	3	0.8664	0.2428	12.73	0.0004
	4	-3.0399	0.8031	14.33	0.0002
Passive	5	0.3334	0.1739	3.67	0.0552
	6	0.4650	0.1990	5.46	0.0195
	7	-0.2865	0.2019	2.01	0.1558
	8	0.9110	0.1615	31.81	<.0001
	9	0.6992	0.1530	20.88	<.0001

Since the PARM option is specified, the predicted response functions are computed from the WLS fit (this table is not shown here). For the IPF method, the "Maximum Likelihood Predicted Values for Frequencies" table is displayed by default; however, the predicted standard errors are not computed unless the PARM option is specified. The predicted standard errors are computed through WLS.

Output 1.1.8. Predicted Frequencies

Maximum Likelihood Predicted Values for Frequencies

		Observed		Predicted		
Active	Passive	Frequency	Standard Error	Frequency	Standard Error	Residual
r	s	1	0.997725	5.259562	1.361573	-4.25956
r	t	5	2.210512	2.48072	0.691065	2.51928
r	u	8	2.776525	8.21586	1.855129	-0.21586
r	v	9	2.937996	6.648033	1.509317	2.351967
r	w	0	0	0.395767	0.240267	-0.39577
s	r	29	5.017696	19.18631	3.147955	9.813693
s	t	14	3.620648	10.32189	2.16963	3.678112
s	u	46	6.031734	34.18491	4.428728	11.81509
s	v	4	1.981735	27.66143	3.722828	-23.6614
s	w	0	0	1.646726	0.952727	-1.64673
u	r	2	1.407771	10.93611	2.12318	-8.93611
u	s	3	1.720201	12.47391	2.554314	-9.47391
u	t	1	0.997725	5.88343	1.380627	-4.88343
u	v	38	5.606814	15.76689	2.684647	22.23311
u	w	2	1.407771	0.938627	0.551631	1.061373
v	r	0	0	0.219965	0.22182	-0.21997
v	s	0	0	0.250896	0.253756	-0.2509
v	t	0	0	0.118337	0.120336	-0.11834
v	u	0	0	0.39192	0.393325	-0.39192
v	w	1	0.997725	0.018879	0.021731	0.981121
w	r	9	2.937996	9.657617	1.808652	-0.65762
w	s	25	4.707344	11.01564	2.275041	13.98436
w	t	4	1.981735	5.195624	1.18445	-1.19562
w	u	6	2.415857	17.20731	2.772074	-11.2073
w	v	13	3.497402	13.92365	2.241575	-0.92365

The model of independence does not fit since the likelihood ratio test for the interaction is significant. In other words, active and passive behaviors of the squirrel monkeys are dependent behavior roles.

Results from using the ML=NR option instead of the ML=IPF option are very similar, since these are just two different algorithms for maximum likelihood estimation. Due to the sampling zeros in the table, use of the WLS method is not recommended.

Example 1.2. Identifying an Inappropriate Model

Suppose you have the following data, and you want to use IPF to fit the "no three-factor effect" model:

```
data pathological;
   input X Y Z count @@;
   datalines;
1 1 1  0  1 1 2 15
1 2 1 15  1 2 2 24
```

```
2 1 1 17  2 1 2 14
2 2 1 16  2 2 2  0
;
```

For this model, it turns out that $n_{111} = n_{222} = 0$ implies the cell frequency estimates $\hat{m}_{111} = \hat{m}_{222} = 0$. This means that the table has only 6 degrees of freedom (non-structural zero cells) available, while the model requires 7 degrees of freedom (one degree for each of the mean, X, Y, Z, XY, XZ, and YZ). Therefore, in order to analyze the data appropriately, these two cells should be dropped from the table and treated as structural zeros, and the model should be reduced. You may be able to identify cases like this with PROC CATMOD by observing convergence problems or by noting that the predicted frequency of a cell seems to be converging to zero:

```
proc catmod data=pathological;
   weight count;
   model X*Y*Z=_response_ / ml=ipf zero=sampling;
   loglin X|Y|Z@2;
run;
```

Output 1.2.1. ML=IPF with ZERO=SAMPLING

```
WARNING: The IPF algorithm failed to converge.
```

When the sampling zeros are replaced by structural zeros, the adjusted degrees of freedom for the likelihood ratio are negative; this is another signal that the model is inappropriate for the data:

```
proc catmod data=pathological;
   weight count;
   model X*Y*Z=_response_ / ml=ipf;
   loglin X|Y|Z@2;
run;
```

Output 1.2.2. ML=IPF with Structural Zeros

```
                The IPF algorithm converged.


          Maximum Likelihood Analysis of Variance

 Source               DF   Chi-Square    Pr > ChiSq
 ------------------------------------------------
 Likelihood Ratio      0          .           .

WARNING: Negative adjusted degrees of freedom were
         calculated for the Likelihood Ratio test.
         The model may be inappropriate.
```

When using ML=NR, you receive a note about having redundant parameters in the model, and you may get messages about having infinite parameters:

```
proc catmod data=pathological;
   weight count;
   model X*Y*Z=_response_ / ml=nr;
   loglin X|Y|Z@2;
run;
```

Output 1.2.3. ML=NR with Structural Zeros

```
          Maximum likelihood computations converged.


           Maximum Likelihood Analysis of Variance

Source               DF   Chi-Square    Pr > ChiSq
-------------------------------------------------
X                     1         0.00        0.9931
Y                     1         0.73        0.3930
X*Y                   1         1.23        0.2682
Z                     1         0.32        0.5723
X*Z                   1         1.85        0.1739
Y*Z                   0*         .           .

Likelihood Ratio      0          .           .

NOTE: Effects marked with '*' contain one or more
      redundant or restricted parameters.
```

This example is discussed further in Bishop, Fienberg, and Holland (1975, p. 115), Agresti (1990, p. 245), and Christensen (1997, p. 292).

References

Agresti, A. (1990), *Categorical Data Analysis*, New York: John Wiley & Sons, Inc.

Bishop, Y.M.M., Fienberg, S.E., and Holland, P.W. (1975), *Discrete Multivariate Analysis: Theory and Practice*, Cambridge, MA: The MIT Press.

Christensen, R. (1997), *Log-Linear Models and Logistic Regression*, New York: Springer-Verlag.

Haberman, S.J. (1972), "Log-Linear Fit for Contingency Tables," *Applied Statistics*, 21, 218–225.

Haslett, S. (1990), "Degrees of Freedom and Parameter Estimability in Hierarchical Models for Sparse Complete Contingency Tables," *Computational Statistics and Data Analysis*, 9, 179–195.

Chapter 2
The FACTOR Procedure

Chapter Table of Contents

OVERVIEW 17

SYNTAX 17
PROC FACTOR Statement 17

DETAILS 21
Output Data Sets 21
Confidence Intervals and the Salience of Factor Loadings 21
Simplicity Functions for Rotations 23

EXAMPLE 24
Example 2.1 Using Confidence Intervals to Locate Salient Factor Loadings . 24

REFERENCES 29

Chapter 2
The FACTOR Procedure

Overview

Enhancements to the FACTOR procedure include:

- the generalized Crawford-Ferguson family of rotations (Jennrich 1973) using the ROTATE= option, including the direct oblimin, the quartimin, and many other specialized othogonal and oblique rotations
- control of the number of rotation cycles and the criterion for rotational convergence using the RITER= and the RCONVERGE= options
- standard error estimates for the unrotated and various rotated solutions using the SE option under the maximum likelihood estimation (Archer and Jennrich 1973, Hayashi and Yung 1999, Jennrich 1973)
- confidence intervals and coverage displays for the factor loadings, the factor correlations, and the factor structure loadings using the CI= option under the maximum likelihood estimation
- control of the percentage coverage of the confidence interval using the ALPHA= option

Syntax

New options for the PROC FACTOR statement are added.

PROC FACTOR Statement

PROC FACTOR < *options* > **;**

The following new or updated options are available:

ALPHA=*p*
specifies the level of confidence $1-p$ for interval construction. By default, p = 0.05, corresponding to $1-p$ = 95% confidence intervals. If p is greater than one, it is interpreted as a percentage and divided by 100. Because the coverage probability is not controlled simultaneously, you may consider supplying a nonconventional p using methods such as Bonferroni adjustment.

COVER <**=***p*>
CI <**=***p*>
computes the confidence intervals and optionally specifies the value of factor loading

for coverage detection. By default, p = 0. The specified value is represented by an asterisk '*' in the coverage display. This is useful for determining the salience of loadings. For example, if COVER=.4, a display '0*[ ]' indicates that the entire confidence interval is above 0.4, implying strong evidence for the salience of the loading. See the section "Confidence Intervals and the Salience of Factor Loadings" on page 21 for more details.

HKPOWER=*p*
HKP=*p*

specifies the power of the square roots of the eigenvalues used to rescale the eigenvectors for Harris-Kaiser (ROTATE=HK) rotation, assuming that the factors are extracted by the principal factor method. If the principal factor method is not used for factor extraction, the eigenvectors are replaced by the normalized columns of the unrotated factor matrix, and the eigenvalues replaced by the column normalizing constants. HKPOWER= values between 0.0 and 1.0 are reasonable. The default value is 0.0, yielding the independent cluster solution, in which each variable tends to have a large loading on only one factor. An HKPOWER= value of 1.0 is equivalent to an orthogonal rotation, with the varimax rotation as the default. You can also specify the HKPOWER= option with ROTATE=QUARTIMAX, ROTATE=BIQUARTIMAX, ROTATE=EQUAMAX, or ROTATE=ORTHOMAX, and so on. The only restriction is that the Harris-Kaiser rotation must be associated with an orthogonal rotation.

PREROTATE=*name*
PRE=*name*

specifies the prerotation method for the option ROTATE=PROMAX. Any rotation method other than PROMAX or PROCRUSTES can be used. See the ROTATE= option for the available prerotation methods. The default is PREROTATE=VARIMAX. If a previously rotated pattern is read using the option METHOD=PATTERN, you should specify the PREROTATE=NONE option.

RCONVERGE=*p*
RCONV=*p*

specifies the convergence criterion for rotation cycles. Rotation stops when the scaled change of the simplicity function value is less than the RCONVERGE= value. The default convergence criterion is

$$|f_{new} - f_{old}|/K < \epsilon$$

where f_{new} and f_{old} are simplicity function values of the current cycle and the previous cycle, respectively, $K = max(1, |f_{old}|)$ is a scaling factor, and ϵ is 1E-9 by default and is modified by the RCONVERGE= value.

RITER=*n*

specifies the maximum number of cycles for factor rotation. Except for promax and Procrustes, you can use the RITER= option with all rotation methods. The default is the maximum between 100 and ten times of the number of variables.

ROTATE=*name*
R=*name*

specifies the rotation method. The default is ROTATE=NONE.

Valid *name*s for orthogonal rotations are as follows:

BIQUARTIMAX | BIQMAX specifies orthogonal biquartimax rotation. This corresponds to the specification ROTATE=ORTHOMAX(.5).

EQUAMAX | E specifies orthogonal equamax rotation. This corresponds to the specification ROTATE=ORTHOMAX with GAMMA=*number of factors*/2.

FACTORPARSIMAX | FPA specifies orthogonal factor parsimax rotation. This corresponds to the specification ROTATE=ORTHOMAX with GAMMA=*number of variables*.

NONE | N specifies that no rotation be performed, leaving the original orthogonal solution.

ORTHCF(*p1*,*p2*) | ORCF(*p1*,*p2*) specifies the orthogonal Crawford-Ferguson rotation with the weights *p1* and *p2* for variable and factor parsimony, respectively. See the definitions of weights in the section "Simplicity Functions for Rotations" on page 23.

ORTHGENCF(*p1*,*p2*,*p3*,*p4*) | ORGENCF(*p1*,*p2*,*p3*,*p4*) specifies the orthogonal generalized Crawford-Ferguson rotation with the four weights *p1*, *p2*, *p3*, and *p4*. See the definitions of weights in the section "Simplicity Functions for Rotations" on page 23.

ORTHOMAX<(*p*)> | ORMAX<(*p*)> specifies the orthomax rotation. If ROTATE=ORTHOMAX is used, the orthomax weight is specified by the GAMMA= option. You can also specify the GAMMA= value in the parentheses of ROTATE=ORTHOMAX(*p*). See the definition of the orthomax weight in the section "Simplicity Functions for Rotations" on page 23.

PARSIMAX | PA specifies orthogonal parsimax rotation. This corresponds to the specification ROTATE=ORTHOMAX with

$$\text{GAMMA} = \frac{nvar \times (nfact - 1)}{nvar + nfact - 2}$$

where *nvar* is the number of variables, and *nfact* is the number of factors.

QUARTIMAX | QMAX | Q specifies orthogonal quartimax rotation. This corresponds to the specification ROTATE=ORTHOMAX(0).

VARIMAX | V specifies orthogonal varimax rotation. This corresponds to the specification ROTATE=ORTHOMAX with GAMMA=1.

Valid *name*s for oblique rotations are as follows:

BIQUARTIMIN | BIQMIN specifies biquartimin rotation. It corresponds to the specification ROTATE=OBLIMIN(.5) or ROTATE=OBLIMIN with TAU=.5.

COVARIMIN | CVMIN specifies covarimin rotation. It corresponds to the specification ROTATE=OBLIMIN(1) or ROTATE=OBLIMIN with TAU=1.

HK<(*p*)> | H<(*p*)> specifies Harris-Kaiser case II orthoblique rotation. When specifying this option, you can use the HKPOWER= option to set the power of the square roots of the eigenvalues by which the eigenvectors are scaled, assuming that the factors are extracted by the principal factor method. For other extraction methods, the unrotated factor pattern is column normalized. The power is then applied to the column normalizing constants, instead of the eigenvalues. You can also use ROTATE=HK(*p*), with *p* representing the HKPOWER= value. The default associated orthogonal rotation with ROTATE=HK is the varimax rotation without Kaiser normalization. You may associate the Harris-Kaiser with other orthogonal rotations using the ROTATE= option together with the HKPOWER= option.

OBBIQUARTIMAX | OBIQMAX specifies oblique biquartimax rotation.

OBEQUAMAX | OE specifies oblique equamax rotation.

OBFACTORPARSIMAX | OFPA specifies oblique factor parsimax rotation.

OBLICF(*p1*,*p2*) | OBCF(*p1*,*p2*) specifies the oblique Crawford-Ferguson rotation with the weights *p1* and *p2* for variable and factor parsimony, respectively. See the definitions of weights in the section "Simplicity Functions for Rotations" on page 23.

OBLIGENCF(*p1*,*p2*,*p3*,*p4*) | OBGENCF(*p1*,*p2*,*p3*,*p4*) specifies the oblique generalized Crawford-Ferguson rotation with the four weights *p1*, *p2*, *p3*, and *p4*. See the definitions of weights in the section "Simplicity Functions for Rotations" on page 23.

OBLIMIN<(*p*)> | OBMIN<(*p*)> specifies the oblimin rotation. If ROTATE=OBLIMIN is used, the oblimin weight is specified by the TAU= option. Alternatively, ROTATE=OBLIMIN(*p*) specifies *p* as the TAU= value. See the definition of the oblimin weight in the section "Simplicity Functions for Rotations" on page 23.

OBPARSIMAX | OPA specifies oblique parsimax rotation.

OBQUARTIMAX | OQMAX specifies oblique quartimax rotation. This is the same as the QUARTIMIN method.

OBVARIMAX | OV specifies oblique varimax rotation.

PROCRUSTES specifies oblique Procrustes rotation with the target pattern provided by the TARGET= data set. The unrestricted least squares method is used with factors scaled to unit variance after rotation.

PROMAX<(*p*)> | P<(*p*)> specifies oblique promax rotation. You can use the PREROTATE= option to set the desirable prerotation method, orthogonal or oblique. When using with ROTATE=PROMAX, the POWER= option lets you specify the power for forming the target. You can also use ROTATE=PROMAX(*p*), where *p* represents the POWER= value.

QUARTIMIN | QMIN specifies quartimin rotation. It is the same as the oblique quartimax method. It also corresponds to the specification ROTATE=OBLIMIN(0) or ROTATE=OBLIMIN with TAU=0.

SE
STDERR
computes standard errors for various classes of unrotated and rotated solutions under the maximum likelihood estimation.

Details

Output Data Sets

The OUTSTAT= Data Set

If standard error estimates are available, the output data set contains the following new observations:

TYPE	Contents
SE_PREFC	standard error estimates for pre-rotated interfactor correlations. The _NAME_ variable contains the name of the factor.
SE_PREPA	standard error estimates for the pre-rotated loadings. The _NAME_ variable contains the name of the factor.
SE_PREST	standard error estimates for pre-rotated structure loadings. The _NAME_ variable contains the name of the factor.
SE_FCORR	standard error estimates for interfactor correlations. The _NAME_ variable contains the name of the factor.
SE_PAT	standard error estimates for the rotated loadings. The _NAME_ variable contains the name of the factor.
SE_STRUC	standard error estimates for structure loadings. The _NAME_ variable contains the name of the factor.

Confidence Intervals and the Salience of Factor Loadings

The traditional approach to determining salient loadings (loadings that are considered large in absolute values) employs rules-of-thumb such as 0.3 or 0.4. However, this does not utilize the statistical evidence efficiently. The asymptotic normality of the distribution of factor loadings enables you to construct confidence intervals to gauge the salience of factor loadings. To guarantee the range-respecting properties of confidence intervals, a transformation procedure such as in CEFA (Browne, Cudeck, Tateneni, and Mels 1998) is used. For example, because the orthogonal rotated factor loading θ must be bounded between -1 and $+1$, the Fisher transformation

$$\varphi = \frac{1}{2}\log(\frac{1+\theta}{1-\theta})$$

is employed so that φ is an unbounded parameter. Assuming the asymptotic normality of $\hat{\varphi}$, a symmetric confidence interval for φ is constructed. Then, a back-transformation on the confidence limits yields an asymmetric confidence interval for θ. Applying the results of Browne (1982), a $(1-\alpha)100\%$ confidence interval for the orthogonal factor loading θ is

$$(\hat{\theta}_l = \frac{a/b - 1}{a/b + 1},\ \hat{\theta}_u = \frac{a \times b - 1}{a \times b + 1})$$

where

$$a = \frac{1 + \hat{\theta}}{1 - \hat{\theta}},\ \ b = \exp(z_{\alpha/2} \times \frac{2\hat{\sigma}}{1 - \hat{\theta}^2})$$

and $\hat{\theta}$ is the estimated factor loading, $\hat{\sigma}$ is the standard error estimate of the factor loading, and $z_{\alpha/2}$ is the $(1 - \alpha/2)100$ percentile point of a standard normal distribution.

Once the confidence limits are constructed, you can use the corresponding coverage displays for determining the salience of the variable-factor relationship. In a coverage display, the COVER= value is represented by an asterisk '*'. The following table summarizes the various displays and their interpretations.

Table 2.1. Interpretations of the Coverage Displays

Positive Estimate	Negative Estimate	COVER=0 specified	Interpretation
[0]*	*[0]		The estimate is not significantly different from zero and the CI covers a region of values that are smaller in magnitude than the COVER= value. This is strong statistical evidence for the non-salience of the variable-factor relationship.
0[ ]*	*[ ]0		The estimate is significantly different from zero but the CI covers a region of values that are smaller in magnitude than the COVER= value. This is strong statistical evidence for the non-salience of the variable-factor relationship.
[0*]	[*0]	[0]	The estimate is not significantly different from zero or the COVER= value. The population value might have been larger or smaller in magnitude than the COVER= value. There is no statistical evidence for the salience of the variable-factor relationship.
0[*]	[*]0		The estimate is significantly different from zero but not from the COVER= value. This is marginal statistical evidence for the salience of the variable-factor relationship.
0*[ ]	[ ]*0	0[ ] or [ ]0	The estimate is significantly different from zero and the CI covers a region of values that are larger in magnitude than the COVER= value. This is strong statistical evidence for the salience of the variable-factor relationship.

See Example 2.1 on page 24 for an illustration of the use of confidence intervals for interpreting factors.

Simplicity Functions for Rotations

To rotate a factor pattern is to apply a non-singular linear transformation to the unrotated factor pattern matrix. To arrive at an optimal transformation you must define a so-called simplicity function for assessing the optimal point. For the promax or the Procrustean transformation, the simplicity function is defined as the sum of squared differences between the rotated factor pattern and the target matrix. Thus, the solution of the optimal transformation is easily obtained by the familiar least-squares method.

For the class of the generalized Crawford-Ferguson family (Jennrich 1973), the simplicity function being optimized is

$$f = k_1 Z + k_2 H + k_3 V + k_4 Q$$

where

$$Z = (\sum_j \sum_i b_{ij}^2)^2, \quad H = \sum_i (\sum_j b_{ij}^2)^2$$

$$V = \sum_j (\sum_i b_{ij}^2)^2, \quad Q = \sum_j \sum_i b_{ij}^4$$

k_1, k_2, k_3, and k_4 are constants, and b_{ij} represents an element of the rotated pattern matrix. Except for specialized research purposes, it is rare in practice to use this simplicity function for rotation. However, it reduces to many well-known classes and special cases of rotations. One of these is the Crawford-Ferguson family (Crawford and Ferguson 1970), which minimizes

$$f_{cf} = c_1(H - Q) + c_2(V - Q)$$

where c_1 and c_2 are constants and $(H - Q)$ represents variable (row) parsimony and $(V - Q)$ represents factor (column) parsimony. Therefore, the relative importance of the variable and the factor parsimony is adjusted via the constants c_1 and c_2. The orthomax class (Carroll, see Harman 1976) maximizes the function

$$f_{or} = pQ - \gamma V$$

where γ is the orthomax weight and is usually between 0 and the number of variables p. The oblimin class minimizes the function

$$f_{ob} = p(H - Q) - \tau(Z - V)$$

where τ is the oblimin weight and is usually between 0 and the number of variables p.

All the above definitions are for rotations without row normalization. For rotations with Kaiser normalization the definition of b_{ij} is replaced by b_{ij}/h_i, where h_i is the communality of variable i.

Example

Example 2.1. Using Confidence Intervals to Locate Salient Factor Loadings

This example illustrates how you can utilize the standard errors and confidence intervals to understand the pattern of factor loadings under the maximum likelihood estimation. There are nine tests and you want a three-factor solution for a correlation matrix based on 200 observations. You apply quartimin rotation with (default) Kaiser normalization. You define loadings with magnitudes greater than 0.45 to be salient and use 90% confidence intervals to judge the salience:

```
data test(type=corr);
   title 'Quartimin-Rotated Factor Solution with Standard Errors';
   input _name_ $ test1-test9;
   _type_ = 'corr';
datalines;
Test1      1  .561  .602  .290  .404  .328  .367  .179 -.268
Test2   .561     1  .743  .414  .526  .442  .523  .289 -.399
Test3   .602  .743     1  .286  .343  .361  .679  .456 -.532
Test4   .290  .414  .286     1  .677  .446  .412  .400 -.491
Test5   .404  .526  .343  .677     1  .584  .408  .299 -.466
Test6   .328  .442  .361  .446  .584     1  .333  .178 -.306
Test7   .367  .523  .679  .412  .408  .333     1  .711 -.760
Test8   .179  .289  .456  .400  .299  .178  .711     1 -.725
Test9  -.268 -.399 -.532 -.491 -.466 -.306 -.760 -.725     1
;
proc factor data=test method=ml reorder rotate=quartimin
   nobs=200 n=3 se cover=.45 alpha=.1;
   title2 'A nine-variable-three-factor example';
run;
```

Output 2.1.1. Quartimin-Rotated Factor Solution with Standard Errors

```
      Quartimin-Rotated Factor Solution with Standard Errors
              A nine-variable-three-factor example

                    The FACTOR Procedure
                  Rotation Method: Quartimin

                  Inter-Factor Correlations
                 With 90% confidence limits
               Estimate/StdErr/LowerCL/UpperCL

                    Factor1         Factor2         Factor3

  Factor1           1.00000         0.41283         0.38304
                    0.00000         0.06267         0.06060
                     .              0.30475         0.27919
                     .              0.51041         0.47804

  Factor2           0.41283         1.00000         0.47006
                    0.06267         0.00000         0.05116
                    0.30475          .              0.38177
                    0.51041          .              0.54986

  Factor3           0.38304         0.47006         1.00000
                    0.06060         0.05116         0.00000
                    0.27919         0.38177          .
                    0.47804         0.54986          .
```

After the quartimin rotation, the correlation matrix for factors is shown in Output 2.1.1. The factors are medium to highly correlated. The confidence intervals seem to be very wide, suggesting that the estimation of factor correlations may not be very accurate for this sample size. For example, the 90% confidence interval for the correlation between Factor1 and Factor2 is (0.30, 0.51), a range of 0.21. You may need a larger sample to get a narrower interval, or a better estimation.

Output 2.1.2. Interpretations of Factors Using Rotated Factor Pattern

A nine-variable-three-factor example

The FACTOR Procedure
Rotation Method: Quartimin

Rotated Factor Pattern (Standardized Regression Coefficients)
With 90% confidence limits; Cover |*| = 0.45?
Estimate/StdErr/LowerCL/UpperCL/Coverage Display

	Factor1	Factor2	Factor3
test8	0.86810	-0.05045	0.00114
	0.03282	0.03185	0.03087
	0.80271	-0.10265	-0.04959
	0.91286	0.00204	0.05187
	0*[]	*[0]	[0]*
test7	0.73204	0.27296	0.01098
	0.04434	0.05292	0.03838
	0.65040	0.18390	-0.05211
	0.79697	0.35758	0.07399
	0*[]	0[]*	[0]*
test9	-0.79654	-0.01230	-0.17307
	0.03948	0.04225	0.04420
	-0.85291	-0.08163	-0.24472
	-0.72180	0.05715	-0.09955
	[]*0	*[0]	*[]0
test3	0.27715	0.91156	-0.19727
	0.05489	0.04877	0.02981
	0.18464	0.78650	-0.24577
	0.36478	0.96481	-0.14778
	0[]*	0*[]	*[]0
test2	0.01063	0.71540	0.20500
	0.05060	0.05148	0.05496
	-0.07248	0.61982	0.11310
	0.09359	0.79007	0.29342
	[0]*	0*[]	0[]*
test1	-0.07356	0.63815	0.13983
	0.04245	0.05380	0.05597
	-0.14292	0.54114	0.04682
	-0.00348	0.71839	0.23044
	*[]0	0*[]	0[]*
test5	0.00863	0.03234	0.91282
	0.04394	0.04387	0.04509
	-0.06356	-0.03986	0.80030
	0.08073	0.10421	0.96323
	[0]*	[0]*	0*[]
test4	0.22357	-0.07576	0.67925
	0.05956	0.03640	0.05434
	0.12366	-0.13528	0.57955
	0.31900	-0.01569	0.75891
	0[]*	*[]0	0*[]
test6	-0.04295	0.21911	0.53183
	0.05114	0.07481	0.06905
	-0.12656	0.09319	0.40893
	0.04127	0.33813	0.63578
	*[0]	0[]*	0[*]

The coverage displays in Output 2.1.2 show that Test8, Test7, and Test9 have salient relationships with Factor1. The coverage displays are either '0*[ ]' or '[ ]*0', indicating that the entire 90% confidence intervals for the corresponding loadings are beyond the salience value at 0.45. On the other hand, the coverage display for Test3 on Factor1 is '0[ ]*'. This indicates that even though the loading estimate is significantly larger than zero, it is not large enough to be salient. Similarly, Test3, Test2, and Test1 have salient relationships with Factor2, while Test5 and Test4 have salient relationships with Factor3. For Test6, its relationship with Factor3 is a little bit ambiguous; the 90% confidence interval covers approximately values between 0.40 and 0.64. This means that the population value might have been smaller or larger than 0.45. It is marginal evidence for a salient relationship.

Output 2.1.3. Interpretations of Factors Using Factor Structure

A nine-variable-three-factor example

The FACTOR Procedure
Rotation Method: Quartimin

Factor Structure (Correlations)
With 90% confidence limits; Cover |*| = 0.45?
Estimate/StdErr/LowerCL/UpperCL/Coverage Display

	Factor1	Factor2	Factor3
test8	0.84771	0.30847	0.30994
	0.02871	0.06593	0.06263
	0.79324	0.19641	0.20363
	0.88872	0.41257	0.40904
	0*[]	0[]*	0[]*
test7	0.84894	0.58033	0.41970
	0.02688	0.05265	0.06060
	0.79834	0.48721	0.31523
	0.88764	0.66041	0.51412
	0*[]	0*[]	0[*]
test9	-0.86791	-0.42248	-0.48396
	0.02522	0.06187	0.05504
	-0.90381	-0.51873	-0.56921
	-0.81987	-0.31567	-0.38841
	[]*0	[*]0	[*]0
test3	0.57790	0.93325	0.33738
	0.05069	0.02953	0.06779
	0.48853	0.86340	0.22157
	0.65528	0.96799	0.44380
	0*[]	0*[]	0[]*
test2	0.38449	0.81615	0.54535
	0.06143	0.03106	0.05456
	0.27914	0.75829	0.44946
	0.48070	0.86126	0.62883
	0[*]	0*[]	0[*]
test1	0.24345	0.67351	0.41162
	0.06864	0.04284	0.05995
	0.12771	0.59680	0.30846
	0.35264	0.73802	0.50522
	0[]*	0*[]	0[*]
test5	0.37163	0.46498	0.93132
	0.06092	0.04979	0.03277
	0.26739	0.37923	0.85159
	0.46727	0.54282	0.96894
	0[*]	0[*]	0*[]
test4	0.45248	0.33583	0.72927
	0.05876	0.06289	0.04061
	0.35072	0.22867	0.65527
	0.54367	0.43494	0.78941
	0[*]	0[]*	0*[]
test6	0.25122	0.45137	0.61837
	0.07140	0.05858	0.05051
	0.13061	0.34997	0.52833
	0.36450	0.54232	0.69465
	0[]*	0[*]	0*[]

For oblique factor solutions, some researchers prefer to examine the factor structure loadings, which represent correlations, for determining salient relationships. In Output 2.1.3, the factor structure loadings and the associated standard error estimates and coverage displays are shown. The interpretations based on the factor structure matrix do not change much except for Test3 and Test9. Test9 now has a salient correlation with Factor3. For Test3, it has salient correlations with both Factor1 and Factor2. Fortunately, there are still tests that only have salient correlations with either Factor1 or Factor2 (but not both). This would make interpretations of factors less problematic.

References

Archer, C.O. and Jennrich, R.I. (1973), "Standard Errors for Orthogonally Rotated Factor Loadings," *Psychometrika*, 38, 581–592.

Browne, M.W. (1982), "Covariance Structures," in *Topics in Applied Multivariate Analysis*, ed. D.M. Hawkins, Cambridge: Cambridge University Press, 72–141.

Browne, M.W., Cudeck, R., Tateneni, K., and Mels, G. (1998), *CEFA: Comprehensive Exploratory Factor Analysis.* [http://quantrm2.psy.ohio-state.edu/browne/].

Crawford, C.B. and Ferguson, G.A. (1970), "A General Rotation Criterion and Its Use in Orthogonal Rotation," *Psychometrika*, 35, 321–332.

Harman, H.H. (1976), *Modern Factor Analysis*, Third Edition, Chicago: University of Chicago Press.

Hayashi, K. and Yung, Y.F. (1999), "Standard Errors for the Class of Orthomax-Rotated Factor Loadings: Some Matrix Results," *Psychometrika*, 64, 451–460.

Jennrich, R.I. (1973), "Standard Errors for Obliquely Rotated Factor Loadings," *Psychometrika*, 38, 593–604.

Chapter 3
The FREQ Procedure

Chapter Table of Contents

OVERVIEW 33

SYNTAX 33
TABLES Statement 33

Chapter 3
The FREQ Procedure

Overview

PROC FREQ has three new TABLES statement options to customize procedure output. The CONTENTS= option specifies the HTML contents link for crosstabulation tables. The FORMAT= option formats the frequencies displayed in crosstabulation tables. The OUTCUM option includes cumulative frequencies and cumulative percentages in the output data set for one-way tables.

Syntax

TABLES Statement

The following options have been added to the TABLES statement:

CONTENTS=*link-text*

specifies the text for the HTML contents file links to crosstabulation tables. For information on HTML output, refer to *The Complete Guide to the SAS Output Delivery System*. The CONTENTS= option affects only the HTML contents file, and not the HTML body file.

If you omit the CONTENTS= option, by default the HTML link text for crosstabulation tables is "Cross-Tabular Freq Table."

Note that links to all crosstabulation tables produced by a single TABLES statement use the same text. To specify different text for different crosstabulation table links, request the tables in separate TABLES statements and use the CONTENTS= option in each TABLES statement.

The CONTENTS= option affects only links to crosstabulation tables. It does not affect links to other PROC FREQ tables. To specify link text for any other PROC FREQ table, you can use PROC TEMPLATE to create a customized table definition. The CONTENTS_LABEL attribute in the DEFINE TABLE statement of PROC TEMPLATE specifies the contents file link for the table. For detailed information, refer to the chapter titled "The TEMPLATE Procedure" in *The Complete Guide to the SAS Output Delivery System*.

FORMAT=*format-name*

specifies a format for the following crosstabulation table cell values: frequency, expected frequency, and deviation. PROC FREQ also uses this format to display the total row and column frequencies for crosstabulation tables.

You can specify any standard SAS numeric format or a numeric format defined with the FORMAT procedure. The format length must not exceed 24. If you omit

FORMAT=, by default PROC FREQ uses the BEST6. format to display frequencies less than 1E6, and the BEST7. format otherwise.

OUTCUM

includes the cumulative frequency and the cumulative percentage for one-way tables in the output data set when you specify the OUT= option in the TABLES statement. The variable CUM_FREQ contains the cumulative frequency for each level of the analysis variable, and the variable CUM_PCT contains the cumulative percentage for each level. The OUTCUM option has no effect for two-way or multiway tables.

Chapter 4
The GAM Procedure

Chapter Table of Contents

OVERVIEW 37

GETTING STARTED 37

SYNTAX 42
- PROC GAM Statement 42
- BY Statement 42
- CLASS Statement 43
- FREQ Statement 43
- ID Statement 44
- MODEL Statement 44
- OUTPUT Statement 45
- SCORE Statement 46

DETAILS 46
- Nonparametric Regression 46
- Additive Models and Generalized Additive Models 47
- Back-Fitting and Local Scoring Algorithms 48
- Smoothers 51
- Selection of Smoothing Parameters 52
- Distribution Family and Canonical Link 53
- Forms of Additive Models 53
- ODS Tables Produced by PROC GAM 54

EXAMPLES 54
- Example 4.1 Generalized Additive Model with Binary Data 54
- Example 4.2 Comparing PROC GAM with PROC TPSPLINE 61

REFERENCES 66

Chapter 4
The GAM Procedure

Overview

The GAM procedure is a new, experimental procedure that fits generalized additive models as those models are defined by Hastie and Tibshirani (1990). This procedure provides an array of powerful tools for data analysis, based on nonparametric regression and smoothing techniques.

Nonparametric regression relaxes the usual assumption of linearity and enables you to uncover structure in the relationship between the independent variables and the dependent variable that might otherwise be missed. The SAS System provides many procedures for nonparametric regression, such as the LOESS procedure for local regression and the TPSPLINE procedure for thin-plate smoothing splines. The generalized additive models fit by the GAM procedure combine

- an additive assumption (Stone 1985) that allows relatively many nonparametric relationships to be explored simultaneously with
- the distributional flexibility of generalized linear models (Nelder 1972)

Thus, you can use the GAM procedure when you have multiple independent variables whose effect you want to model nonparametrically, or when the dependent variable is not normally distributed. Refer to the "Nonparametric Regression" section on page 46 for more details on the form of generalized additive models.

The GAM procedure

- provides nonparametric estimates for additive models
- supports the use of multidimensional data
- supports multiple SCORE statements
- fits both generalized semiparametric additive models and generalized additive models
- enables you to choose a particular model by specifying the model degrees of freedom or smoothing parameter

Getting Started

The following example illustrates the use of the GAM procedure to explore in a nonparametrical way how two factors affect a response. The data come from a study (Sockett et al. 1987) of the factors affecting patterns of insulin-dependent diabetes

mellitus in children. The objective is to investigate the dependence of the level of serum C-peptide on various other factors in order to understand the patterns of residual insulin secretion. The response measurement is the logarithm of C-peptide concentration (pmol/ml) at diagnosis, and the predictor measurements are age and base deficit (a measure of acidity):

```
title 'Patterns of Diabetes';
data diabetes;
   input Age BaseDeficit CPeptide @@;
   logCP = log(CPeptide);
   datalines;
 5.2   -8.1  4.8   8.8  -16.1  4.1  10.5   -0.9  5.2
10.6   -7.8  5.5  10.4  -29.0  5.0   1.8  -19.2  3.4
12.7  -18.9  3.4  15.6  -10.6  4.9   5.8   -2.8  5.6
 1.9  -25.0  3.7   2.2   -3.1  3.9   4.8   -7.8  4.5
 7.9  -13.9  4.8   5.2   -4.5  4.9   0.9  -11.6  3.0
11.8   -2.1  4.6   7.9   -2.0  4.8  11.5   -9.0  5.5
10.6  -11.2  4.5   8.5   -0.2  5.3  11.1   -6.1  4.7
12.8   -1.0  6.6  11.3   -3.6  5.1   1.0   -8.2  3.9
14.5   -0.5  5.7  11.9   -2.0  5.1   8.1   -1.6  5.2
13.8  -11.9  3.7  15.5   -0.7  4.9   9.8   -1.2  4.8
11.0  -14.3  4.4  12.4   -0.8  5.2  11.1  -16.8  5.1
 5.1   -5.1  4.6   4.8   -9.5  3.9   4.2  -17.0  5.1
 6.9   -3.3  5.1  13.2   -0.7  6.0   9.9   -3.3  4.9
12.5  -13.6  4.1  13.2   -1.9  4.6   8.9  -10.0  4.9
10.8  -13.5  5.1
;
run;
```

The following statements perform the desired analysis. The PROC GAM statement invokes the procedure and specifies the diabetes data set as input. The MODEL statement specifies logCP as the response variable and requests that univariate BSPLINEs with 4 degrees of freedom be used to model the effect of Age and BaseDeficit. The OUTPUT statement specifies that partial prediction curves are to be saved in the data set estimates:

```
proc gam data=diabetes;
   model logCP = spline(age) spline(BaseDeficit);
   output out=estimates p;
run;
```

The results are shown in Figure 4.1 and Figure 4.2.

```
                          Patterns of Diabetes

                            The GAM Procedure
                        Dependent Variable: logCP
          Smoothing Model Component: spline(Age) spline(BaseDeficit)

                    Iteration Summary and Fit Statistics

          Final number of backfitting iterations                    5
          Final backfitting criterion                    5.542743E-10
          Final residual sum of squares                  0.4180802183


                         Summary of Input Data Set

                  Number of Observations                   43
                  Number of Missing Observations            0
                  Distribution                       Gaussian
                  Link Function                      Identity
```

Figure 4.1. Summary Statistics

Figure 4.1 shows two tables. The first table summarizes the convergence criterion for back-fitting, and the second one summarizes the input data set and the distribution family used for the model.

```
                          Patterns of Diabetes

                            The GAM Procedure
                        Dependent Variable: logCP
          Smoothing Model Component: spline(Age) spline(BaseDeficit)

                         Regression Model Analysis
                            Parameter Estimates

                        Parameter       Standard
       Parameter         Estimate          Error    t Value    Pr > |t|

       Intercept          1.48141        0.02588      57.24      <.0001
       L_Age              0.01437        0.00437       3.28      0.0024
       L_BaseDeficit      0.00807        0.00240       3.35      0.0020


                         Smoothing Model Analysis
                  Fit Statistics of Smoothing Components

                                                                    No. of
                        Smoothing                                   Unique
Component               Parameter             DF            GCV     Obs.

spline(Age)              0.995582       4.000000       0.011675       37
spline(BaseDeficit)      0.995299       4.000000       0.012437       39


                         Smoothing Model Analysis
                           Analysis of Deviance

                                          Sum of
    Source                      DF        Squares    F Value    Pr > F

    spline(Age)           4.000000       0.150760      12.26    0.0155
    spline(BaseDeficit)   4.000000       0.081272       6.61    0.1580
```

Figure 4.2. Analysis of Model

Figure 4.2 displays summary statistics for the model. It consists of three tables. The first is the Parameter Estimates table for the parametric part of the model. It indicates that the linear trends for both Age and BaseDeficit are highly significant with *p*-values of 0.0024 and 0.0020. The second table is the summary of smoothing components of the nonparametric part of the model. Since the GAM fit used the default DF = 4, the main point of this table is to present the smoothing parameter values that yield this DF for each component. Finally, the third table shows the Analysis of Deviance table for the nonparametric component of the model.

The P option in the OUTPUT statement puts the partial predictions for Age and BaseDeficit in the output data set. You can compute the entire partial prediction effect for each factor by adding the estimated linear terms to the respective partial predictions, as in the following statement:

```
data estimates; set estimates;
   P2_age         = P_age         + 0.01437*age;
   P2_BaseDeficit = P_BaseDeficit + 0.00807*BaseDeficit;
run;
```

Plotting the partial predictions is one way to explore the overall shape of the relationship between each factor and the response. First of all, the following statements set up the graphics options:

```
axis1 label=(angle=90 rotate=0) minor=none;
axis2 minor=none order=(0 to 16 by 4);
symbol1 color=red  interpol=join value=none line=1;
symbol2 color=blue interpol=join value=none line=2;
```

Then the following statements plot the partial prediction curves for Age and BaseDeficit:

```
proc sort data=estimates;
   by age;

proc gplot data=estimates;
   plot P_age *age = 1
        P2_Age*Age = 2 /overlay legend frame cframe=ligr
            name='gam1' vaxis=axis1 haxis=axis2;
run;

proc sort data=estimates;
   by BaseDeficit;

proc gplot data=estimates;
   plot P_BaseDeficit *BaseDeficit = 1
        P2_BaseDeficit*BaseDeficit = 2 /
           overlay legend frame cframe=ligr name='gam2'
           vaxis=axis1 haxis=axis2;
run;
```

Finally, the following statements redisplay the curves side by side for easy comparison:

```
goptions display;
proc greplay  tc=tempcat nofs;
   igout gseg;
tdef newtwo des='two plots of equal size'
1/llx=0   lly=0
  ulx=0   uly=100
  urx=50  ury=100
  lrx=50  lry=0
2/llx=50  lly=0
  ulx=50  uly=100
  urx=100 ury=100
  lrx=100 lry=0
;
template newtwo;
   treplay 1:gam1
           2:gam2;
run; quit;
```

The resulting plots for each predictor with and without the linear term are shown in Figure 4.3.

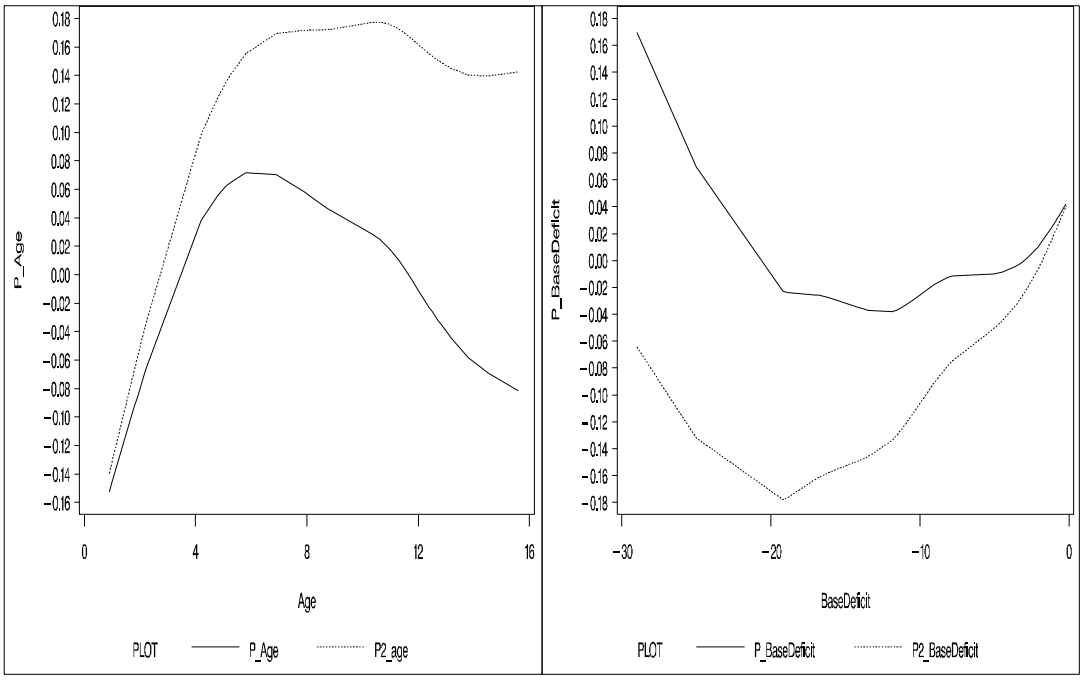


Figure 4.3. Partial Prediction for Each Predictor

Both plots show a strong quadratic pattern, with a possible indication of higher-order behavior. Further investigation is required to determine whether these patterns are real or not.

Syntax

PROC GAM < *option* > **;**
CLASS *variables* **;**
MODEL *dependent* = < *PARAM(effects)* >
smoothing effects < */options* > **;**
SCORE *data=SAS-data-set out=SAS-data-set* **;**
OUTPUT <*out=SAS-data-set*> *keyword* < · · ·*keyword*> <*/option* > **;**
BY *variables* **;**
ID *variables* **;**
FREQ *variable* **;**

The syntax of the GAM procedure is similar to that of other regression procedures in the SAS System. The PROC GAM and MODEL statements are required. The SCORE statement can appear multiple times; all other statements appear only once.

The syntax for PROC GAM is described in the following sections in alphabetical order after the description of the PROC GAM statement.

PROC GAM Statement

PROC GAM< *option* > **;**

The PROC GAM statement invokes the procedure. You can specify the following option.

DATA=*SAS-data-set*
specifies the SAS data set to be read by PROC GAM. The default value is the most recently created data set.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC GAM to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the GAM procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (ac-

cording to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.

- Create an index for the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide.*

CLASS Statement

CLASS *variables* **;**

The CLASS statement names the classification variables to be used in the analysis. Typical class variables are TREATMENT, SEX, RACE, GROUP, and REPLICATION. If the CLASS statement is used, it must appear before the MODEL statement.

Classification variables can be either character or numeric. Class levels are determined from the formatted values of the CLASS variables. Thus, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *SAS Procedures Guide*, and the discussions for the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*.

FREQ Statement

FREQ *variable* **;**

The FREQ statement names a variable that provides frequencies for each observation in the DATA= data set. Specifically, if *n* is the value of the FREQ variable for a given observation, then that observation is used *n* times.

The analysis produced using a FREQ statement reflects the expanded number of observations. You can produce the same analysis (without the FREQ statement) by first creating a new data set that contains the expanded number of observations. For example, if the value of the FREQ variable is 5 for the first observation, the first five observations in the new data set are identical. Each observation in the old data set is replicated n_i times in the new data set, where n_i is the value of the FREQ variable for that observation.

If the value of the FREQ variable is missing or is less than 1, the observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

ID Statement

ID *variables* **;**

The variables in the ID statement are copied from the input data set to the OUT= data set. If you omit the ID statement, only the variables used in the MODEL statement and requested statistics are included in the output data set.

MODEL Statement

MODEL *dependent* = <*PARAM(effects)*> <*smoothing effects*> </*options*> **;**

The MODEL statement specifies the dependent variable and the independent effects you want to use to model its values. Specify the independent parametric variables inside the parentheses of PARAM( ). The parametric variables can be either CLASS variables or continuous variables. Any number of smoothing effects can be specified, as follows:

Smoothing Effect	Meaning
spline(variable, <df=number>)	fit smoothing spline with the variable and with DF=number
spline2(variable, variable, <df=number>)	fit bivariate thin-plate spline with DF=number

Both parametric effects and smoothing effects are optional, but at least one of them must be present.

If only parametric variables are present, PROC GAM fits a parametric linear model using the terms inside the parentheses of PARAM( ). If only smoothing effects are present, PROC GAM fits a nonparametric additive model. If both types of effect are present, PROC GAM fits a semiparametric model using the parametric effects as the linear part of the model.

The following table shows how to specify various models for a dependent variable y and independent variables x, x1, and x2.

Table 4.1. Syntax for Common GAM Models

Type of Model	Syntax	Mathematical Form
Parametric	model y = param(x);	$E(y) = \beta_0 + \beta_1 x$
Nonparametric	model y = spline(x);	$E(y) = \beta_0 + s(x_2)$
Semiparametric	model y = param(x1) spline(x2);	$E(y) = \beta_0 + \beta_1 x_1 + s(x_2)$
Additive	model y = spline(x1) spline(x2);	$E(y) = \beta_0 + s_1(x_1) + s_2(x_2)$
Thin-plate spline	model y = spline(x1,x2);	$E(y) = \beta_0 + s(x_1, x_2)$

You can specify the following options in the MODEL statement.

ALPHA=*number*
specifies the significance level α of the confidence limits on the final nonparametric component estimates when you request confidence limits to be included in the output data set. Specify *number* as a value between 0 and 1. The default value is 0.05. Refer to the "OUTPUT Statement" section on page 45 for more information on the OUTPUT statement.

DIST=*distribution-id*
specifies the distribution family used in the model. The *distribution-id* can be GAUSSIAN or LOGISTIC. The canonical link is used with those distributions. Although theoretically, alternative links are possible, with nonparametric models the final fit is relatively insensitive to the precise choice of link function. Therefore, only the canonical link for each distribution family is implemented in PROC GAM.

EPSILON=*number*
specifies the convergence criterion for the back-fitting algorithm.

ITPRINT
produces an iteration table for the smoothing effects.

MAXITER=*number*
specifies the maximum number of iterations for the back-fitting algorithm.

METHOD=GCV
specifies that the value of the smoothing parameter should be selected by generalized cross validation. If you specify both METHOD=GCV and the DF= option for the smoothing effects, the user-specified DF= is used, and the METHOD=GCV option is ignored. Refer to the "Selection of Smoothing Parameters" section on page 52 for more details on the GCV method.

OUTPUT Statement

OUTPUT *OUT=SAS-data-set* < *keyword* ··· *keyword* > **;**

The OUTPUT statement creates a new SAS data set containing diagnostic measures calculated after fitting the model.

You can request a variety of diagnostic measures that are calculated for each observation in the data set. The new data set contains the variables specified in the MODEL statement in addition to the requested variables. If no *keyword* is present, the data set contains only the predicted values.

Details on the specifications in the OUTPUT statement are as follows.

OUT=*SAS-data-set*
specifies the name of the new data set to contain the diagnostic measures. This specification is required.

keyword
specifies the statistics to include in the output data set. The keywords and the statistics they represent are as follows:

PRED	predicted values
ADIAG	diagonal element of the hat matrix associated with the observation

The names of the new variables that contain the statistics are formed by using a prefix of one or more characters that identify the statistic, followed by an underscore (_), followed by the dependent variable name.

For example, suppose that you have a dependent variable y, and you specify the keywords PRED and ADIAG. In this case, the output SAS data set will contain the variables P_y and ADIAG_y.

SCORE Statement

SCORE *DATA=SAS-data-set OUT=SAS-data-set* **;**

The SCORE statement calculates predicted values for a new data set. If you have multiple data sets to predict, you can specify multiple SCORE statements. You must use a SCORE statement for each data set.

The following keywords must be specified in the SCORE statement.

DATA=*SAS-data-set*
specifies an input SAS data set containing all the variables included in independent effects in the MODEL statement. The predicted response is computed for each observation in the SCORE DATA= data set.

OUT=*SAS-data-set*
specifies the name of the SAS data set to contain the predictions.

Details

Nonparametric Regression

Nonparametric regression relaxes the usual assumption of linearity and enables you to explore the data visually, uncovering structure in the data that might otherwise be missed.

However, many forms of nonparametric regression do not perform well when the number of independent variables in the model is large. The sparseness of data in this setting causes the variances of the estimates to be unacceptably large unless the sample size is extremely large. The problem of rapidly increasing variance for increasing dimensionality is sometimes referred to as the "curse of dimensionality." Interpretability is another problem with nonparametric regression based on kernel and smoothing spline estimates. The information these estimates contain about the relationship between the dependent and independent variables is often difficult to comprehend.

To overcome these difficulties, Stone (1985) proposed additive models. These models estimate an additive approximation to the multivariate regression function. The

benefits of an additive approximation are at least twofold. First, since each of the individual additive terms is estimated using a univariate smoother, the curse of dimensionality is avoided, at the cost of not being able to approximate universally. Second, estimates of the individual terms explain how the dependent variable changes with the corresponding independent variables.

To extend the additive model to a wide range of distribution families, Hastie and Tibshirani (1990) proposed generalized additive models. These models enable the mean of the dependent variable to depend on an additive predictor through a nonlinear link function. The models permit the response probability distribution to be any member of the exponential family of distributions. Many widely used statistical models belong to this general class; they include additive models for Gaussian data, nonparametric logistic models for binary data, and nonparametric log-linear models for Poisson data.

Additive Models and Generalized Additive Models

This section describes the methodology and the fitting procedure behind generalized additive models.

Let Y be a response random variable and $X_1, X_2, \cdots, X_p$ be a set of predictor variables. A regression procedure can be viewed as a method for estimating the expected value of Y given the values of $X_1, X_2, \cdots, X_p$. The standard linear regression model assumes a linear form for the conditional expectation

$$E(Y|X_1, X_2, \cdots, X_p) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

Given a sample, estimates of $\beta_0, \beta_1, \cdots, \beta_p$ are usually obtained by the least squares method.

The additive model generalizes the linear model by modeling the conditional expectation as

$$E(Y|X_1, X_2, \cdots, X_p) = s_0 + s_1(X_1) + s_2(X_2) + \cdots + s_p(X_p)$$

where $s_i(X), i = 1, 2, \ldots, p$ are smooth functions.

In order to be estimable, the smooth functions s_i have to satisfy standardized conditions such as $Es_j(X_j) = 0$. These functions are not given a parametric form but instead are estimated in a nonparametric fashion.

While traditional linear models and additive models can be used in most statistical data analysis, there are types of problems for which they are not appropriate. For example, the normal distribution may not be adequate for modeling discrete responses such as counts or bounded responses such as proportions.

Generalized additive models address these difficulties, extending additive models to many other distributions besides just the normal. Thus, generalized additive models can be applied to a much wider range of data analysis problems.

Similar to generalized linear models, generalized additive models consist of a random component, an additive component, and a link function relating the two components. The response Y, the random component, is assumed to have exponential family density

$$f_Y(y;\theta;\phi) = \exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right\}$$

where θ is called the natural parameter and ϕ is the scale parameter. The mean of the response variable μ is related to the set of covariates $X_1, X_2, \cdots, X_p$ by $g(\mu) = \eta$. Here, η is defined as

$$\eta = s_0 + \sum_{i=1}^{p} s_i(X_i)$$

where $s_1(\cdot), \cdots, s_p(\cdot)$ are smooth functions, the quantity η is the linear component, and $g(\cdot)$ is the link function. The most commonly used link for a given f is called the canonical link, for which $\eta = \theta$.

Generalized additive models and generalized linear models can be applied in similar situations, but they serve different analytic purposes. Generalized linear models emphasize estimation and inference for the parameters of the model, while generalized additive models focus on exploring data nonparametrically. Generalized additive models are more suitable for exploring the data set and visualizing the relationship between the dependent variable and the independent variables.

Back-Fitting and Local Scoring Algorithms

Consider the estimation of the smoothing terms $s_0, s_1(\cdot), \cdots, s_p(\cdot)$ in the additive model

$$\mu(X) = s_0 + \sum_{i=1}^{p} s_j(X_j)$$

where $E\left[s_j(X_j)\right] = 0$ for every j. Since the algorithm for additive models is the basis for fitting generalized additive models, the algorithm for additive models is discussed first.

Many ways are available to approach the formulation and estimation of additive models. The back-fitting algorithm is a general algorithm that can fit an additive model using any regression-type fitting mechanisms.

Define the partial residual as

$$R_j = Y - s_0 - \sum_{k \neq j} s_k(X_k)$$

Then $E(R_j|X_j) = s_j(X_j)$. This observation provides a way for estimating each smoothing function $s_j(\cdot)$ given estimates $\{\hat{s}_i(\cdot), i \neq j\}$ for all the others. The resulting iterative procedure is known as the back-fitting algorithm (Friedman and Stuetzle 1981).

The Back-Fitting Algorithm

1. **Initialization:**
   $s_0 = E(Y), s_1^1 = s_2^1 = \cdots = s_p^1 = 0, m = 0.$
2. **Iterate:**
   $m = m + 1$
   for $j = 1$ to p do:
   $R_j = Y - s_0 - \sum_{k=1}^{j-1} s_k^m(X_k) - \sum_{k=j+1}^{p} s_k^{m-1}(X_k)$
   $S_j^m = E(R_j|X_j).$
3. **Until:**
   $RSS = \text{Avg}(Y - s_0 - \sum_{j=1}^{p} s_j^m(X_j))^2$ fails to decrease.

In the above notation, $s_j^m(\cdot)$ denotes the estimate of $s_j(\cdot)$ at the mth iteration. It can be shown that RSS never increases at any step, which implies that the algorithm always converges. However, the individual functions need not be unique, since dependence among the covariates can lead to more than one representation for the same fitted surface.

A weighted back-fitting algorithm has the same form as for the unweighted case, except that the smoothers are weighted. The weights might represent the relative precision of each observation or might arise as part of another iterative procedure. For example, weights are used in the local scoring procedure described later in this section.

The algorithm so far described fits just additive models. The algorithm for generalized additive models is a little more complicated. Generalized additive models extend generalized linear models in the same manner that additive models extend linear regression models, that is, by replacing form $\alpha + \sum_j X_j\beta_j$ with the additive form $\alpha + \sum_j f_j(X_j)$. Thus, it is helpful to review the iteratively reweighted least-square procedure for computing the maximum likelihood estimates in a generalized linear model.

For generalized linear models, the maximum likelihood estimate of β is defined by the score equations

$$\sum_{i=1}^{n} x_{ij} \left(\frac{\partial \mu_i}{\partial \eta_i}\right) V_i^{-1}(y_i - \mu_i),\ j = 0, 1, \cdots, p$$

where V_i is the variance matrix for Y_i. The Fisher scoring procedure is the standard method for solving these equations. It involves a Newton-Raphson algorithm using the expected (as opposed to the observed) information matrix. An equivalent procedure that is convenient for this problem is called dependent variable regression and is a form of iteratively reweighted least squares. Given a current coefficient vector

β^0, with corresponding linear predictor η^0 and fitted values μ^0, construct the adjusted dependent variable

$$z_i = \eta_i^0 + (y_i - \mu_i^0)\left(\frac{\partial \eta_i}{\partial \mu_i}\right)_0$$

Define weights w_i by

$$w_i^{-1} = \left(\frac{\partial \eta_i}{\partial \mu_i}\right)_0^2 V_i^0$$

The algorithm proceeds by regressing z_i on x with weight w_i to obtain a revised estimate β. Then a new μ^0 and η^0 are computed, new z_is are computed, and the process is repeated until the change in the deviance

$$D(y;\hat{\mu}) = 2(l(\mu_{max};y) - l(\hat{\mu}'y))$$

is sufficiently small.

Some adjusted dependent variables and weights for commonly used models are listed in the following table.

Distribution	Link	Adjusted Dependent(Z)	Weights(w)
Normal	identity	y	1
Bin(n,μ)	logit	$\eta + (y-\mu)/n\mu(1-\mu)$	$n\mu(1-\mu)$
Gamma	log	$\eta + (y-\mu)/\mu$	1
Poisson	log	$\eta + (y-\mu)/\mu$	μ

Generalized additive models differ from generalized linear models in that an additive predictor replaces the linear predictor. Estimation of the additive terms is accomplished by replacing the weighted linear regression in the adjusted dependent variable regression by the weighted back-fitting algorithm for fitting a weighted additive mode. This results in the algorithm described below as the *local scoring algorithm*. The name "local scoring" derives from the fact that local averaging is used to generalize the Fisher scoring procedure.

The General Local Scoring Algorithm

1. **Initialization:**
   $s_i = g(E(y)), s_1^0 = s_2^0 = \cdots = s_p^0 = 0, m = 0.$
2. **Iterate:**
   $m = m + 1$
   Form the adjusted dependent variable, predicator, and mean based on the previous iteration
   $Z = \eta^{m-1} + (Y - \mu^{m-1})(\partial\eta/\partial\mu^{m-1})$
   $\eta^{m-1} = s_0 + \sum_{j=1}^{p} s_j^{m-1}(X_j)$
   $\mu^{m-1} = g^{-1}(\eta^{m-1}).$

Form the weights
$w_i = (\partial\mu^{m-1}/\partial\eta^{m-1})^2 V_i^{-1}$.
Fit an additive model to Z using the back-fitting algorithm with weights W to obtain estimated functions $s_j^m(\cdot)$.

3. **Until:**
Avg($D(Y, \mu^m)$) fails to decrease, where Avg($D(Y, \mu^m)$) is an average of the deviance of estimate μ^m.

The estimating procedure for generalized additive models consists of two loops. Inside each step of the local scoring algorithm (outer loop), a weighted back-fitting algorithm (inner loop) is used until convergence, that is, until the RSS fails to decrease. Then, based on the estimates from this weighted back-fitting algorithm, a new set of weights is calculated and the next iteration of the scoring algorithm starts. The scoring algorithm stops when the deviance of the estimates ceases to decrease.

Smoothers

A smoother is a tool for summarizing the trend of a response measurement Y as a function of one or more predictor measurements $X_1, \cdots, X_p$. It produces an estimate of the trend that is less variable than Y itself. An important property of a smoother is its nonparametric nature. It doesn't assume a rigid form for the dependence of Y on $X_1, \cdots, X_p$. This section gives a brief overview of the smoothers that can be used with the GAM procedure.

Cubic Smoothing Spline

A smoothing spline is the solution to the following optimization problem: among all functions $s(x)$ with two continuous derivatives, find one that minimizes the penalized least square

$$\sum_{i=1}^{n}(y_i - s(x_i))^2 + \lambda \int_a^b (f^{''}(t))^2 dt$$

where λ is a fixed constant, and $a \leq x_1 \leq \cdots \leq x_n \leq b$. The first term measures closeness to the data while the second term penalizes curvature in the function. It can be shown that there exists an explicit, unique minimizer, and that minimizer is a natural cubic spline with knots at the unique values of x_i.

The parameter λ is the smoothing parameter. Large values of λ produce smoother curves while smaller values produce wiggly curves.

Thin-Plate Smoothing Spline

The theoretical foundations for the thin-plate smoothing spline are described in Duchon (1976, 1977) and Meinguet (1979). Further results and applications are given in Wahba and Wendelberger (1980). Refer to "The TPSPLINE Procedure" in *SAS/STAT User's Guide, Version 8* for more details.

Selection of Smoothing Parameters

CV and GCV

The smoothers discussed here have a single smoothing parameter. In choosing the smoothing parameter, cross validation can be used. Cross validation works by leaving points (x_i, y_i) out one at a time, estimating the squared residual for smooth function at x_i based on the remaining $n-1$ data points, and choosing the smoother to minimize the sum of those squared residuals. This mimics the use of training and test samples for prediction. The cross validation function is defined as

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{s}_{\lambda}^{-i}(x_i))^2$$

where $\hat{s}_{\lambda}^{-i}(x_i)$ indicates the fit at x_i, computed by leaving out the ith data point. The quantity $nCV(\lambda)$ is sometimes called the prediction sum of squares or $PRESS$ (Allen 1974).

All of the smoothers fit by the GAM procedure can be formulated as a linear combination of the sample responses

$$\hat{s}(x) = A(\lambda)Y$$

for some matrix $A(\lambda)$, which depends on λ. (The matrix $A(\lambda)$ depends on x and the sample data, as well, but this dependence is suppressed in the preceding equation.) Let a_{ii} be the diagonal elements of the $A(\lambda)$. Then the CV function can be expressed as

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{(y_i - \hat{s}_{\lambda}(x_i))}{1 - a_{ii}}\right)^2$$

In most cases, it is very time consuming to compute the quantity a_{ii}. To solve this computational problem, Wahba (1990) has proposed the generalized cross validation function (GCV) that can be used to solve a wide variety of problems involving selection of a parameter to minimize the prediction risk.

The GCV function is defined as

$$GCV(\lambda) = \frac{\sum_{i=1}^{n}(y_i - \hat{s}_{\lambda}(x_i))^2}{(n - \mathrm{tr}(A(\lambda)))^2}$$

The GCV formula simply replaces the a_{ii} with $\mathrm{tr}(A(\lambda))/n$. Therefore, it can be viewed as a weighted version of CV. In most of the cases of interest, GCV is closely related to CV but much easier to compute. The GAM procedure uses the GCV function as the criterion for choosing the smoothing parameters.

The A matrix has the same role as the projection matrix in linear regression; therefore, nonparametric degrees of freedom (DF) for the model can be defined as $\mathrm{tr}(A)$.

Distribution Family and Canonical Link

For each distribution, more than one link can exist. Different link functions may result in a slight difference in estimates for parametric models. However, the difference will be less pronounced for nonparametric models because of the flexibility of nonparametric model forms. To simplify the calculation, the GAM procedure uses the canonical link.

The GAM procedure can fit the data from the Gaussian and binomial distributions:

The Gaussian Model

With this model, the link function is the identical function, and the generalized additive model is the additive model.

The Logistic Model

A binomial response model assumes that the proportion of successes Y is such that Y has a $Bin(n(x), p(x))$ distribution. The $Bin(n(x), p(x))$ refers to the binomial distribution with parameters $n(x)$ and $p(x)$. Often the data are binary, in which case $n(x) = 1$. The canonical link is

$$g(p(x)) = log\frac{p(x)}{1 - p(x)} = \eta(x)$$

Forms of Additive Models

Suppose that y is a continuous variable and x1 and x2 are two explanatory variables of interest. To fit an additive model, you can use a MODEL statement similar to that used in many regression procedures in the SAS system:

```
model y = spline(x1) spline(x2);
```

This model statement requires the procedure to fit the following model:

$$f(x1, x2) = \textit{Intercept} + s_1(x1) + s_2(x2)$$

where the $s_i()$ terms denote nonparametric spline functions of the respective explanatory variables.

The GAM procedure can fit semiparametric models. The following MODEL statement assumes a linear relation with x1 and an unknown functional relation with x2:

```
model y = param(x1) spline(x2);
```

If you want to fit a model containing a functional two-way interaction between x1 and x2, you can use the following MODEL statement:

```
model y = spline2(x1,x2);
```

In this case, the GAM procedure fits a model equivalent to that of PROC TPSPLINE.

ODS Tables Produced by PROC GAM

PROC GAM assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, refer to "Using the Output Delivery System" in *SAS/STAT User's Guide, Version 8*.

Table 4.2. ODS Tables Produced by PROC GAM

ODS Table Name	Description	Statement	Option
ANOVA	ANOVA table for parametric fit	PROC	default
ANODEV	Analysis of Deviance table for smoothing variables	PROC	default
ClassSummary	Summary of class variables	PROC	default
DataSummary	Data summary	PROC	default
IterSummary	Iteration summary	PROC	default
FitSummary	Fit parameters and fit summary	PROC	default
ParameterEstimates	Parameter estimation for regression variables	PROC	default
Iteration	Iteration history table	MODEL	ITPRINT

By referring to the names of such tables, you can use the ODS OUTPUT statement to place one or more of these tables in output data sets.

Examples

Example 4.1. Generalized Additive Model with Binary Data

The following example illustrates the capabilities of the GAM procedure and compares it to the GENMOD procedure.

The data used in this example are based on a study by Bell et al. (1989). Bell and his associates studied the result of multiple-level thoracic and lumbar laminectomy, a corrective spinal surgery commonly performed on children. The data in the study consist of retrospective measurements on 83 patients. The specific outcome of interest is the presence (1) or absence (0) of kyphosis, defined as a forward flexion of the spine of at least 40 degrees from vertical. The available predictor variables are Age in months at time of the operation, the starting of vertebrae levels involved in the operation (StartVert), and the number of levels involved (NumVert). The goal of this analysis is to identify risk factors for kyphosis. PROC GENMOD can be used to investigate the relationship among kyphosis and the predictors. The following DATA step creates the data kyphosis:

```
title 'Comparing PROC GAM with PROC GENMOD';
data kyphosis;
   input Age StartVert NumVert Kyphosis @@;
   datalines;
71  5  3  0     158 14 3 0    128 5   4 1
2   1  5  0     1   15 4 0    1   16  2 0
```

```
61  17 2  0     37  16 3 0    113 16  2 0
59  12 6  1     82  14 5 1    148 16  3 0
18  2  5  0     1   12 4 0    243 8   8 0
168 18 3  0     1   16 3 0    78  15  6 0
175 13 5  0     80  16 5 0    27  9   4 0
22  16 2  0     105 5  6 1    96  12  3 1
131 3  2  0     15  2  7 1    9   13  5 0
12  2 14  1     8   6  3 0    100 14  3 0
4   16 3  0     151 16 2 0    31  16  3 0
125 11 2  0     130 13 5 0    112 16  3 0
140 11 5  0     93  16 3 0    1   9   3 0
52  6  5  1     20  9  6 0    91  12  5 1
73  1  5  1     35  13 3 0    143 3   9 0
61  1  4  0     97  16 3 0    139 10  3 1
136 15 4  0     131 13 5 0    121 3   3 1
177 14 2  0     68  10 5 0    9   17  2 0
139 6  10 1     2   17 2 0    140 15  4 0
72  15 5  0     2   13 3 0    120 8   5 1
51  9  7  0     102 13 3 0    130 1   4 1
114 8  7  1     81  1  4 0    118 16  3 0
118 16 4  0     17  10 4 0    195 17  2 0
159 13 4  0     18  11 4 0    15  16  5 0
158 15 4  0     127 12 4 0    87  16  4 0
206 10 4  0     11  15 3 0    178 15  4 0
157 13 3  1     26  13 7 0    120 13  2 0
42  6  7  1     36  13 4 0
;

proc genmod;
   model Kyphosis = Age StartVert NumVert
                    / link=logit dist=binomial;
run;
```

Output 4.1.1. GENMOD Analysis: Partial Output

```
                  Comparing PROC GAM with PROC GENMOD

                        The GENMOD Procedure

                    Analysis Of Parameter Estimates

                          Standard       Wald 95%          Chi-
Parameter    DF  Estimate     Error   Confidence Limits   Square  Pr > ChiSq

Intercept     1    1.2497    1.2424   -1.1853    3.6848     1.01      0.3145
Age           1   -0.0061    0.0055   -0.0170    0.0048     1.21      0.2713
StartVert     1    0.1972    0.0657    0.0684    0.3260     9.01      0.0027
NumVert       1   -0.3031    0.1790   -0.6540    0.0477     2.87      0.0904
Scale         0    1.0000    0.0000    1.0000    1.0000

NOTE: The scale parameter was held fixed.
```

The GENMOD analysis of the independent variable effects is shown in Output 4.1.1. Based on these results, the only significant factor is StartVert with odds ratio of -0.1972. The variable NumVert has a p-value of 0.0904 with odds ratio of 0.3031.

The GENMOD procedure assumes a strict linear relationship between the response and the predictors. The following SAS statements use PROC GAM to investigate a less restrictive model, with moderately flexible spline terms for each of the predictors:

```
title 'Comparing PROC GAM with PROC GENMOD';
proc gam data=kyphosis;
   model Kyphosis=spline(Age,df=3) spline(StartVert,df=3)
                  spline(NumVert,df=3) /dist = logist;
   output out=estimate p;
run;
```

The MODEL statement requests an additive model using a univariate BSPLINE for each term. The option dist=logist specifies a logistic model. Each term is fitted using a smoothing spline with three degrees of freedom. Although this might seem to be an unduly modest amount of flexibility, it is better to be conservative with a data set this small. An output data set estimate containing predicted values is requested by the OUTPUT statement.

Output 4.1.2 and Output 4.1.3 list the output from PROC GAM.

Output 4.1.2. Summary Statistics

```
                  Comparing PROC GAM with PROC GENMOD

                        The GAM Procedure
                    Dependent Variable: Kyphosis
Smoothing Model Component: spline(Age) spline(StartVert) spline(NumVert)

                 Iteration Summary and Fit Statistics

       Number of local score iterations                          9
       Local score convergence criterion              3.9786363E-9
       Final number of backfitting iterations                    1
       Final backfitting criterion                    5.2326788E-9
       Final residual sum of squares                  46.610928346


                     Summary of Input Data Set

              Number of Observations                  83
              Number of Missing Observations           0
              Distribution                      Binomial
              Link Function                        Logit
```

Output 4.1.3. Analysis of Model

Comparing PROC GAM with PROC GENMOD

The GAM Procedure
Dependent Variable: Kyphosis
Smoothing Model Component: spline(Age) spline(StartVert) spline(NumVert)

Regression Model Analysis
Parameter Estimates

Parameter	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	-2.01545	0.74274	-2.71	0.0083
L_Age	0.01213	0.00622	1.95	0.0552
L_StartVert	-0.18615	0.06061	-3.07	0.0030
L_NumVert	0.38347	0.15264	2.51	0.0142

Smoothing Model Analysis
Fit Statistics of Smoothing Components

Component	Smoothing Parameter	DF	GCV	No. of Unique Obs.
spline(Age)	0.999996	3.000000	328.513619	66
spline(StartVert)	0.999551	3.000000	317.647039	16
spline(NumVert)	0.921758	3.000000	20.144078	10

Smoothing Model Analysis
Analysis of Deviance

Source	DF	Sum of Squares	F Value	Pr > F
spline(Age)	3.000000	10.494366	16.44	0.0009
spline(StartVert)	3.000000	5.494965	8.61	0.0135
spline(NumVert)	3.000000	2.184514	3.42	0.3311

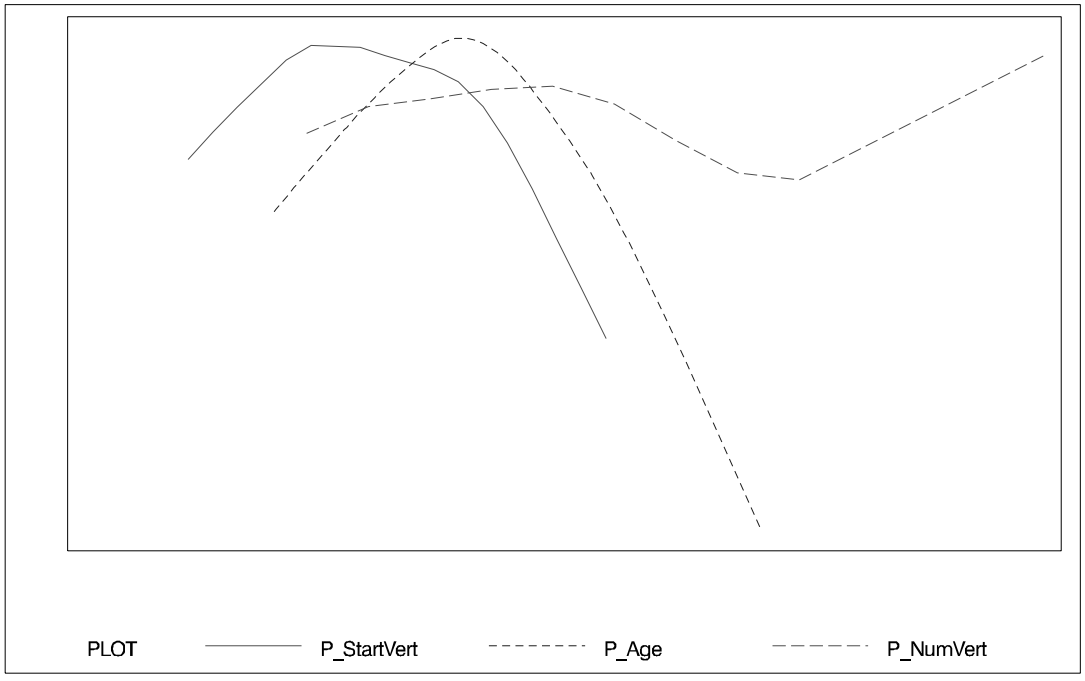
The critical part of the GAM results is the Analysis of Deviance table, shown in Output 4.1.3. For each smoothing effect in the model, this table gives an F-test comparing the deviance between the full model and the model without this variable. In this case, the analysis of deviance results indicates that the effect of Age and StartVert are highly significant, while the effect of NumVert is insignificant. Plots of predictions against predictor can be used to investigate why PROC GAM and PROC GENMOD produce different results.

The GAM statement requests an output data set of predicted values to be created. Since the estimate of the generalized additive model is the sum of functional estimates of individual predictors, plus a constant, the output data set will contain a column of partial prediction for each predictor. If requested, a Bayesian confidence interval or a point-wise standard-error band, as defined in Hastie and Tibshirani (1990), can be produced in the output data set.

Using the following statements, the data set estimate is plotted in Output 4.1.4:

```
proc sort data=estimate(keep=StartVert P_StartVert)
          out =StartVert;
   by StartVert;
proc sort data=estimate(keep=Age       P_Age      )
          out =Age;
   by Age;
proc sort data=estimate(keep=NumVert   P_NumVert  )
          out =NumVert;
   by NumVert;
data Plot; merge StartVert Age NumVert;
proc standard m=0 s=1 data=Plot out=Plot;
   var StartVert Age NumVert;
run;

legend1 frame cframe=ligr cborder=black label=none
        position=center;
axis1 label=(angle=90 rotate=0 " ") minor=none
      value=NONE major=NONE;
axis2 minor=none label=(" ")  major=NONE value=NONE;
symbol1 color=red   interpol=join value=none line=1;
symbol2 color=blue  interpol=join value=none line=2;
symbol3 color=green interpol=join value=none line=3;
proc gplot data=Plot;
   title;
   plot P_StartVert*StartVert=1
        P_Age      *Age      =2
        P_NumVert  *NumVert  =3 / overlay legend frame
        cframe=ligr vaxis=axis1 haxis=axis2;
run;
```

Output 4.1.4. Partial Prediction for Each Predictor



The plot shows that the partial predictions corresponding to both Age and StartVert have a strong quadratic pattern, while NumVert has a more complicated but weaker pattern. However, in the plot for NumVert, notice that about half the vertical range of the function is determined by the point at the upper extreme. It would be a good idea, therefore, to re-run the analysis without this point, to see how much it affects the conclusions. You can do this simply by including a WHERE clause when specifying the data set for the GAM procedure, as in the following code:

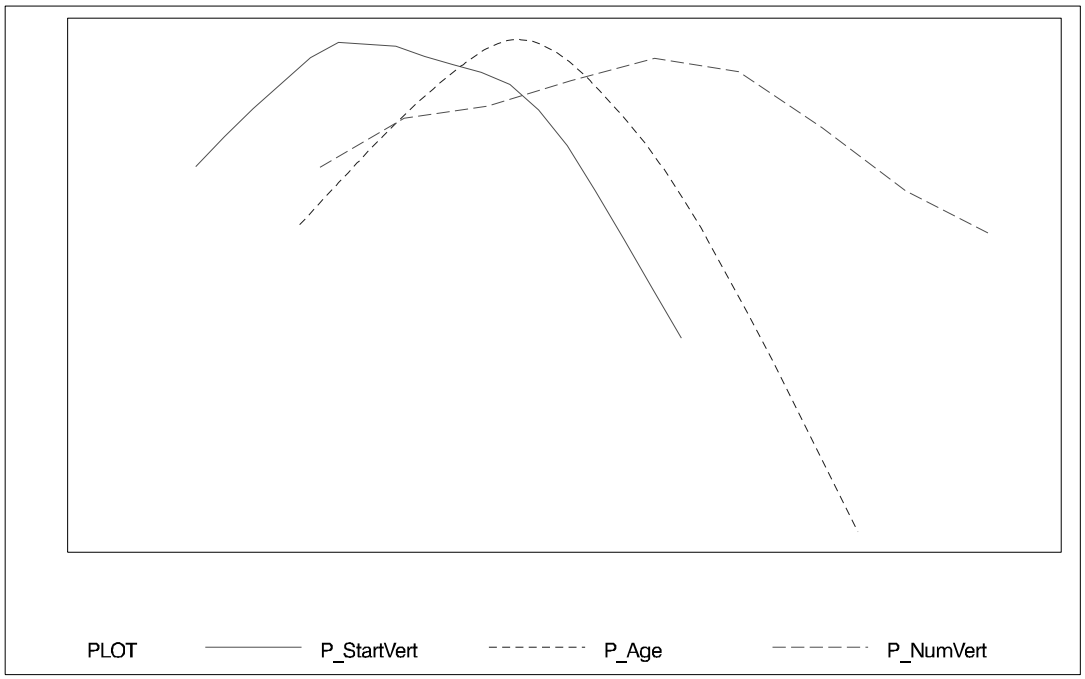
```
title 'Comparing PROC GAM with PROC GENMOD';
proc gam data=kyphosis(where=(NumVert^=14));
   model Kyphosis=spline(Age,df=3) spline(StartVert,df=3)
                  spline(NumVert, df=3) /dist = logist;
   output out=estimate p;
run;
```

The analysis of deviance table from this re-analysis is shown in Output 4.1.5, and Output 4.1.6 shows the re-computed partial predictor plots.

Output 4.1.5. Analysis After Removing NumVert=14

```
                  Comparing PROC GAM with PROC GENMOD

                         The GAM Procedure
                    Dependent Variable: Kyphosis
Smoothing Model Component: spline(Age) spline(StartVert) spline(NumVert)

                       Smoothing Model Analysis
                         Analysis of Deviance

                                          Sum of
Source                         DF        Squares    F Value    Pr > F

spline(Age)                 3.000000      10.587568      16.90    0.0007
spline(StartVert)           3.000000       5.477104       8.74    0.0126
spline(NumVert)             3.000000       3.209100       5.12    0.1630
```

Output 4.1.6. Partial Prediction After Removing NumVert=14



After removing data point NumVert=14, the predictors age and StartVert are still significant and the variable NumVert still insignificant. Based on the plot in Output 4.1.6, the removed point has almost no effect on estimates of curve shape for variables Age and StartVert. But the removal has a dramatic effect on the variable NumVert: this curve for this variable NumVert now also seems quadratic, though it is much less pronounced than for the other two variables.

Having used the GAM procedure to discover an appropriate form of the dependence of Kyphosis on each of the three independent variables, you can use the GENMOD procedure to fit and assess the corresponding parametric model. The following code fits a GENMOD model with quadratic terms for all three variables, including tests for the joint linear and quadratic effects of each variable. The resulting contrast tests are shown in Output 4.1.7.

```
title 'Comparing PROC GAM with PROC GENMOD';
proc genmod data=kyphosis(where=(NumVert^=14));
   model kyphosis = Age       Age      *Age
                    StartVert StartVert*StartVert
                    NumVert   NumVert  *NumVert
                    /link=logit  dist=binomial;
   contrast 'Age'       Age       1, Age*Age            1;
   contrast 'StartVert' StartVert 1, StartVert*StartVert 1;
   contrast 'NumVert'   NumVert   1, NumVert*NumVert     1;
run;
```

Output 4.1.7. Joint Linear and Quadratic Tests

Comparing PROC GAM with PROC GENMOD

The GENMOD Procedure

Contrast Results

Contrast	DF	Chi-Square	Pr > ChiSq	Type
Age	2	13.63	0.0011	LR
StartVert	2	15.41	0.0005	LR
NumVert	2	3.56	0.1684	LR

The results for the quadratic GENMOD model are now quite consistent with the GAM results.

From this example, you can see that PROC GAM is very useful in visualizing the data and detecting the nonlinearity among the variables.

Example 4.2. Comparing PROC GAM with PROC TPSPLINE

This example compares the GAM procedure with the TPSPLINE procedure, another nonparametric procedure that fits a smooth surface to multivariate data. It does not assume additivity of the model and uses very general basis functions for model fitting, making the TPSPLINE procedure much slower than the GAM procedure. For more details about the TPSPLINE procedure, refer to "The TPSPLINE Procedure" in *SAS/STAT User's Guide, Version 8*.

The data used here is also analyzed in "The TPSPLINE Procedure" in *SAS/STAT User's Guide, Version 8*. It presents age-adjusted melanoma incidences for 37 years from the Connecticut Tumor Registry (Houghton, Flannery, and Viola 1980):

```
title 'Comparing PROC GAM with PROC TPSPLINE';
data melanoma;
   input  year incidences @@;
   datalines;
1936    0.9   1937   0.8  1938   0.8  1939   1.3
1940    1.4   1941   1.2  1942   1.7  1943   1.8
1944    1.6   1945   1.5  1946   1.5  1947   2.0
1948    2.5   1949   2.7  1950   2.9  1951   2.5
1952    3.1   1953   2.4  1954   2.2  1955   2.9
1956    2.5   1957   2.6  1958   3.2  1959   3.8
1960    4.2   1961   3.9  1962   3.7  1963   3.3
1964    3.7   1965   3.9  1966   4.1  1967   3.8
1968    4.7   1969   4.4  1970   4.8  1971   4.8
1972    4.8
;
run;
```

The variable incidences records the number of melanoma cases per 100,000 people for the years 1936 to 1972.

Four to five degrees of freedom for each nonparametric term in a generalized additive model fits most data well. However, to select DF more objectively you can use the GCV option to minimize the generalized cross validation function, as shown in the following PROC GAM code:

```
proc gam data=melanoma;
   model incidences = spline(year) /method = GCV;
   output out=gam p;
run;
```

The results are listed in Output 4.2.1 and Output 4.2.2.

Output 4.2.1. Summary Statistics

```
              Comparing PROC GAM with PROC TPSPLINE

                        The GAM Procedure
                    Dependent Variable: incidences
                Smoothing Model Component: spline(year)

                 Iteration Summary and Fit Statistics

Final number of backfitting iterations                        2
Final backfitting criterion                                   0
Final residual sum of squares                    1.2242517494


                      Summary of Input Data Set

           Number of Observations                    37
           Number of Missing Observations             0
           Distribution                        Gaussian
           Link Function                       Identity
```

Output 4.2.2. Analysis of Model

```
                Comparing PROC GAM with PROC TPSPLINE

                          The GAM Procedure
                      Dependent Variable: incidences
                 Smoothing Model Component: spline(year)

                        Regression Model Analysis
                           Parameter Estimates

                     Parameter       Standard
      Parameter       Estimate          Error    t Value    Pr > |t|

      Intercept     -212.69706        7.00491     -30.36      <.0001
      L_year           0.11029        0.00358      30.77      <.0001


                        Smoothing Model Analysis
                  Fit Statistics of Smoothing Components

                                                                 No. of
                    Smoothing                                    Unique
Component           Parameter              DF             GCV     Obs.

spline(year)         0.634903       13.414936        0.088803       37


                        Smoothing Model Analysis
                          Analysis of Deviance

                                       Sum of
   Source                    DF        Squares    F Value    Pr > F

   spline(year)        13.414936        2.736763      50.49    <.0001
```

Based on the summary of the model, the final model has a DF = 13.414936 and the nonparametric trend is highly significant. Note that this DF is much greater than the default value of 4, indicating that there is a great deal of structure in the yearly incidence rates of melanoma. A prediction plot should reveal the nature of this structure:

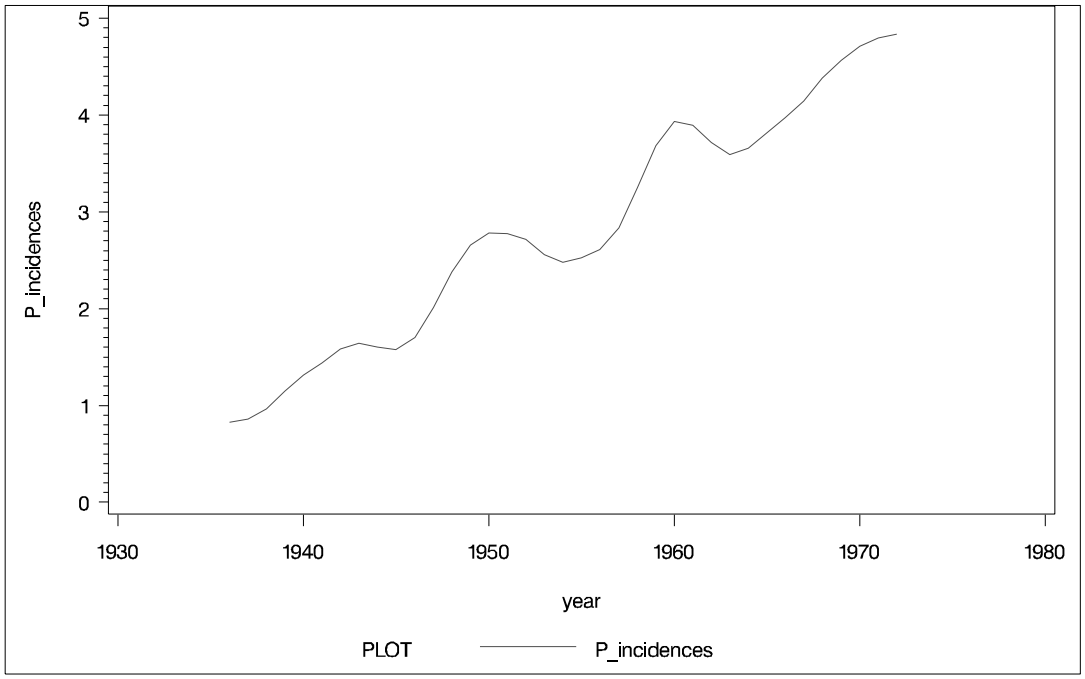
```
legend1 frame cframe=ligr cborder=black label=none
        position=center;
axis1   label=(angle=90 rotate=0);
axis2   minor=none;
symbol1 color=red interpol=join value=none line=1;

proc sort data=gam; by year;
proc gplot data=gam;
   title;
   plot p_incidences*year = 1 /overlay legend
   frame cframe=ligr vaxis=axis1 haxis=axis2;
run;
```

Output 4.2.3 shows the predicted melanoma rate over time. Two features stand out on this plot:

- Melanoma incidence on the whole increases over the period of the study.

- A strong periodic effect is evident, with a period of a little more than a decade. This has been attributed to the 11-year sunspot cycle: the more sunspots there are, the more melanoma cases there are likely to be.

Output 4.2.3. Predicted Melanoma Incidence Rates By Year



Since PROC TPSPLINE also fits a nonparametric model, PROC TPSPLINE and PROC GAM fits should be very similar for this univariate case. The following code produces the TPSPLINE analysis shown in Output 4.2.4:

```
title 'Comparing PROC GAM with PROC GENMOD';
proc tpspline data=melanoma;
   model incidences = (year);
   output out=tpspline p;
run;
```

Output 4.2.4. Analysis from PROC TPSPLINE

```
          Comparing PROC GAM with PROC TPSPLINE

                The TPSPLINE Procedure
          Dependent Variable:  incidences

               Summary of Input Data Set

Number of Non-Missing Observations    37
Number of Missing Observations         0
Unique Smoothing Design Points        37


                Summary of Final Model

Number of Regression Variables        0
Number of Smoothing Variables         1
Order of Derivative in the Penalty    2
Dimension of Polynomial Space         2


                  Summary Statistics
                 of Final Estimation

   log10(n*Lambda)            -0.0607
   Smoothing Penalty           0.5171
   Residual SS                 1.2243
   Tr(I-A)                    22.5852
   Model DF                   14.4148
   Standard Deviation          0.2328
```

The TPSPLINE model analysis shows that the DF for the model is 14.4148. This is consistent with the GAM model because the DF value in GAM excludes the degree of freedom of the linear L_year term. The OUTPUT statements in the PROC GAM and PROC TPSPLINE code create gam and tpspline data sets containing the predicted values for the respective procedures. You can use the following code to look at the values for the two procedures side by side:

```
data both; merge gam     (rename=(p_incidences=gam     ))
                 tpspline(rename=(p_incidences=tpspline));
proc print data=both;
   var year gam tpspline;
run;
```

The results, the first ten of which are displayed in Output 4.2.5, show that PROC GAM and PROC TPSPLINE give essentially the same predictions for this problem.

Output 4.2.5. Melanoma Predictions for First Ten Years

Comparing PROC GAM with PROC TPSPLINE

Obs	year	gam	tpspline
1	1936	0.82425	0.82424
2	1937	0.85580	0.85580
3	1938	0.96379	0.96379
4	1939	1.15046	1.15046
5	1940	1.31044	1.31044
6	1941	1.43881	1.43881
7	1942	1.58218	1.58218
8	1943	1.64382	1.64382
9	1944	1.60148	1.60148
10	1945	1.57498	1.57499

References

Allen, D.M. (1974), "The relationship between variable selection and data augmentation and a method of prediction," *Technometrics*, 16, 125–127.

Bell, D., Walker, J., O'Connor, G., Orrel, J. and Tibshirani, R. (1989), "Spinal Deformation Following Multi-Level Thoracic and Lumbar Laminectomy in Children." Submitted for publication.

Duchon, J. (1976), "Fonctions-Spline et Esperances Conditionnelles de Champs Gaussiens," *Ann. Sci. Univ. Clermont Ferrand II Math.*, 14, 19–27.

Duchon, J. (1977), "Splines Minimizing Rotation-Invariant Semi-Norms in Sovolev Spaces," in *Constructive Theory of Functions of Several Variables*, eds. W. Schempp and K. Zeller, New York: Springer-Verlag, 85–100.

Hastie, T.J. and Tibshirani, R.J. (1990), *Generalized Additive Models*, New York: Chapman and Hall.

Houghton, A. N., Flannery, J. and Viola, M. V. (1980), "Malignant Melanoma in Connecticut and Denmark," *International Journal of Cancer*, 25, 95–104.

Meinguet, J. (1979), "Multivariate Interpolation at Arbitrary Points Made Simple," *Journal of Applied Mathematics and Physics (ZAMP)*, 30, 292–304.

Nelder, J.A. and Wedderburn, R.W.M. (1972) "Generalized Linear models" *Journal of the Royal Statistical Society, Series A*, 135, 370–384.

SAS Institute Inc. (1999), *SAS/STAT User's Guide, Version 8*, Cary, NC: SAS Institute Inc.

Sockett, E.B., Daneman, D., Clarson, C., and Ehrich, R.M. (1987), "Factors Affecting and Patterns of Residual Insulin Secretion During the First Year of Type I (Insulin Dependent) Diabetes Mellitus in Children," *Diabet*, 30, 453–459.

Stone, C.J. (1985), "Additive Regression and Other Nonparametric Models," *Annals of Statistics*, 13, 689–705.

Wahba, G. (1990), *Spline Models for Observational Data*, Philadelphia: Society for Industrial and Applied Mathematics.

Wahba, G. and Wendelberger, J. (1980), "Some New Mathematical Methods for Variational Objective Analysis Using Splines and Cross Validation," *Monthly Weather Review*, 108, 1122–1145.

Chapter 5
The GENMOD Procedure

Chapter Table of Contents

OVERVIEW 71

Chapter 5
The GENMOD Procedure

Overview

The probability of the lower level of binary (dichotomous) response variables specified with the single response variable syntax and the binomial distribution is now modeled by default in the GENMOD procedure. Previously, the higher level of the response variable was modeled.

The DESCENDING option in the PROC GENMOD statement now applies to binary response variables.

Chapter 6
The LOESS Procedure

Chapter Table of Contents

OVERVIEW 75

SYNTAX 75
MODEL Statement 75
SCORE Statement 77

DETAILS 77
kd Trees and Blending 77
Automatic Smoothing Parameter Selection 78
ODS Table Names 81

EXAMPLE 81
Example 6.1 Automatic Smoothing Parameter Selection 81

REFERENCES 86

Chapter 6
The LOESS Procedure

Overview

Enhancements to the LOESS procedure include:

- automatic smoothing parameter selection
- higher order blending for models with one or two regressors
- a table summarizing the fits obtained for each value of the smoothing parameter

Syntax

MODEL Statement

The following new or updated options are available in the MODEL statement after a slash (/).

DETAILS < **(** *tables* **)** >

selects which tables to display, where *tables* is one or more of KDTREE, MODELSUMMARY, OUTPUTSTATISTICS, and PREDATVERTICES:

- KDTREE displays the kd tree structure.
- MODELSUMMARY displays the fit criteria for all smoothing parameter values that are specified in the SMOOTH= option in the MODEL statement, or which are fit with automatic smoothing parameter selection.
- OUTPUTSTATISTICS displays the predicted values and other requested statistics at the points in the input data set.
- PREDATVERTICES displays fitted values and coordinates of the kd tree vertices where the local least squares fitting is done.

The KDTREE and PREDATVERTICES specifications are ignored if the DIRECT option is specified in the MODEL statement. Specifying the option DETAILS with no qualifying list outputs all tables.

INTERP= LINEAR | CUBIC

specifies the degree of the interpolating polynomials used for blending local polynomial fits at the kd tree vertices. This option is ignored if the DIRECT option is specified in the model statement. INTERP=CUBIC is not supported for models with more than two regressors. The default is INTERP=LINEAR.

TRACEL

specifies that the trace of the prediction matrix as well as the GCV and AICC statistics are to be included in the "FIT Summary" table. The use of any of the MODEL statement options ALL, CLM, DFMETHOD=EXACT, DIRECT, SELECT=, or T implicitly selects the TRACEL option.

SELECT=*criterion* **< (<GLOBAL> <STEPS> <RANGE(***lower*,*upper***)>)>**

specifies that automatic smoothing parameter selection be done using the named *criterion*, where the *criterion* is one of AICC, AICC1, or GCV. The smoothing parameter value is selected to yield a minimum of the *criterion*, as follows:

- If you specify the SMOOTH=*value-list* option, then PROC LOESS selects the largest value in this list that yields the global minimum of the specified criterion.
- If you do not specify the SMOOTH= option, then PROC LOESS finds a local minimum of the specified criterion using a golden section search of values less than or equal to one.

You can specify the following modifiers in parentheses after the specified criterion to alter the behavior of the SELECT= option:

- GLOBAL specifies that a global minimum be found within the range of smoothing parameter values examined. This modifier has no effect if you also specify the SMOOTH= option in the MODEL statement.
- STEPS specifies that all models evaluated in the selection process be displayed.
- RANGE(*lower*,*upper*) specifies that only smoothing parameter values greater than or equal to *lower* and less than or equal to *upper* be examined.

For models with one dependent variable, if you specify neither the SELECT= nor the SMOOTH= options in the MODEL statement, then PROC LOESS uses SELECT=AICC.

The following table summarizes how the smoothing parameter values are chosen for various combinations of the SMOOTH= option, the SELECT= option, and the SELECT= option modifiers.

Table 6.1. Smoothing Parameter Value(s) Used for Combinations of SMOOTH= and SELECT= Options for Models with One Dependent Variable

SYNTAX	SEARCH METHOD	SEARCH DOMAIN
default	golden section using AICC	$(0, 1]$
SMOOTH=*list*	no selection	values in *list*
SMOOTH=*list* SELECT=*criterion*	global	values in *list*
SMOOTH=*list* SELECT=*criterion* (RANGE(l, u))	global	values in *list* within $[l, u]$
SELECT=*criterion*	golden section	$(0, 1]$
SELECT=*criterion* (RANGE(*l,u*))	golden section	$[l, u]$
SELECT=*criterion* (GLOBAL)	global	$(0, 1]$
SELECT=*criterion* (GLOBAL RANGE(l, u))	global	$[l, u]$

Note: The SELECT= option cannot be used for models with more than one dependent variable.

SMOOTH=*value-list*

specifies a list of positive smoothing parameter values. If you do not SELECT= option in the MODEL statement, then a separate fit is obtained for each SMOOTH= value specified. If you do specify the SELECT= option, then models with all values specified in the SMOOTH= list are examined, and PROC LOESS selects the value that minimizes the criterion specified in the SELECT= option.

For models with two or more dependent variables, if the SMOOTH= option is not specified in the MODEL statement, then SMOOTH=0.5 is used as a default.

SCORE Statement

The following new option is available in the SCORE statement after a slash (/).

STEPS

requests that all models evaluated during smoothing parameter value selection be scored, provided that the SELECT= option together with the STEPS modifier is specified in the MODEL statement. By default only the selected model is scored.

Details

kd Trees and Blending

PROC LOESS uses a kd tree to divide the box (also called the *initial cell* or *bucket*) enclosing all the predictor data points into rectangular cells. The vertices of these cells are the points at which local least squares fitting is done.

Starting from the initial cell, the direction of the longest cell edge is selected as the split direction. The median of this coordinate of the data in the cell is the split value. The data in the starting cell are partitioned into two child cells. The left child consists of all data from the parent cell whose coordinate in the split direction is less than the

split value. The above procedure is repeated for each child cell that has more than a prespecified number of points, called the *bucket size* of the kd tree.

You can specify the bucket size with the BUCKET= option in the MODEL statement. If you do not specify the BUCKET= option, the default value used is the largest integer less than or equal to $ns/5$, where n is the number of observations and s is the value of the smoothing parameter. Note that if fitting is being done for a range of smoothing parameter values, the bucket size may change for each value.

The set of vertices of all the cells of the kd tree are the points at which PROC LOESS performs its local fitting. The fitted value at an original data point (or at any other point within the original data cell) is obtained by blending the fitted values at the vertices of the kd tree cell that contains that data point.

The univariate blending methods available in PROC LOESS are linear and cubic polynomial interpolation, with linear interpolation being the default. You can request cubic interpolation by specifying the INTERP=CUBIC option in the MODEL statement. In this case, PROC LOESS uses the unique cubic polynomial whose values and first derivatives match those of the fitted local polynomials evaluated at the two endpoints of the kd tree cell edge.

In the multivariate case, such univariate interpolating polynomials are computed on each edge of the kd-tree cells, and are combined using blending functions (Gordon 1971). In the case of two regressors, if you specify INTERP=CUBIC in the MODEL statement, PROC LOESS uses Hermite cubic polynomials as blending functions. If you do not specify INTERP=CUBIC, or if you specify a model with more than two regressors, then PROC LOESS uses linear polynomials as blending functions. In these cases, the blending method reduces to tensor product interpolation from the 2^p vertices of each kd tree cell, where p is the number of regressors.

While the details of the kd tree and the fitted values at the vertices of the kd tree are implementation details that seldom need to be examined, PROC LOESS does provide options for their display. Each kd tree subdivision of the data used by PROC LOESS is placed in the "kdTree" table. The predicted values at the vertices of each kd tree are placed in the "PredAtVertices" table. You can request these tables using the DETAILS option in the MODEL statement.

Automatic Smoothing Parameter Selection

There are several methodologies for automatic smoothing parameter selection. One class of methods chooses the smoothing parameter value to minimize a criterion that incorporates both the tightness of the fit and model complexity. Such a criterion can usually be written as a function of the error mean square, $\hat{\sigma}^2$, and a penalty function designed to decrease with increasing smoothness of the fit. This penalty function is usually defined in terms of the matrix L such that

$$\hat{y} = Ly$$

where y is the vector of observed values and $\hat{y}$ is the corresponding vector of predicted values of the dependent variable. Examples of specific criteria are general-

ized cross-validation (Craven and Wahba 1979) and the Akaike information criterion (Akaike 1973). These classical selectors have two undesirable properties when used with local polynomial and kernel estimators: they tend to undersmooth and tend to be non-robust in the sense that small variations of the input data can change the choice of smoothing parameter value significantly. Hurvich, Simonoff, and Tsai (1998) obtained several bias-corrected AIC criteria that limit these unfavorable properties and perform comparably with the *plug-in selectors* (Ruppert, Sheather, and Wand 1995). PROC LOESS provides automatic smoothing parameter selection using two of these bias-corrected AIC criteria, named AIC_{C_1} and AIC_C in Hurvich, Simonoff, and Tsai (1998), and generalized cross-validation, denoted by the acronym GCV.

The relevant formulae are

$$\begin{aligned} AIC_{C_1} &= n\log(\hat{\sigma}^2) + n\frac{\delta_1/\delta_2(n+\nu_1)}{\delta_1^2/\delta_2 - 2} \\ AIC_C &= \log(\hat{\sigma}^2) + 1 + \frac{2\left(\text{Trace}(L)+1\right)}{n - \text{Trace}(L) - 2} \\ GCV &= \frac{n\hat{\sigma}^2}{(n-\text{Trace}(L))^2} \end{aligned}$$

where n is the number of observations and

$$\begin{aligned} \delta_1 &\equiv \text{Trace}(I-L)^T(I-L) \\ \delta_2 &\equiv \text{Trace}\left((I-L)^T(I-L)\right)^2 \\ \nu_1 &\equiv \text{Trace}(L^TL) \end{aligned}$$

You invoke automatic smoothing parameter selection by specifying the SELECT=*criterion* option in the MODEL statement, where *criterion* is one of AICC1, AICC, or GCV. PROC LOESS evaluates the specified criterion for a sequence of smoothing parameter values and selects the value in this sequence that minimizes the specified criterion. If multiple values yield the optimum, then the largest of these values is selected. The results are summarized in the "Smoothing Criterion" table. This table is displayed whenever automatic smoothing parameter selection is performed. You can obtain details of the sequence of models examined by specifying the DETAILS(MODELSUMMARY) option in the model statement to display the "Model Summary" table.

There are several ways in which you can control the sequence of models examined by PROC LOESS. If you specify the SMOOTH=*value-list* option in the MODEL statement, then only the values in this list are examined in performing the selection. For example, the following statements select the model that minimizes the AICC1 criterion among the three models with smoothing parameter values 0.1, 0.3, and 0.4:

```
proc loess data=notReal;
   model y= x1/ smooth=0.1 0.3 0.4 select=AICC1;
run;
```

If you do not specify the SMOOTH= option in the model statement, then by default PROC LOESS uses a golden section search method to find a local minimum of the specified criterion in the range $(0, 1]$. You can use the RANGE(*lower*,*upper*) modifier in the SELECT= option to change the interval in which the golden section search is performed. For example, the following statements request a golden section search to find a local minimizer of the GCV criterion for smoothing parameter values in the interval [0.1,0.5]:

```
proc loess data=notReal;
   model y= x1/select=GCV( range(0.1,0.5) );
run;
```

If you want to be sure of obtaining a global minimum in the range of smoothing parameter values examined, you can specify the GLOBAL modifier in the SELECT= option. For example, the following statements request that a global minimizer of the AICC criterion be obtained for smoothing parameter values in the interval $[0.2, 0.8]$:

```
proc loess data=notReal;
   model y= x1/select=AICC( global range(0.2,0.8) );
run;
```

Note that even though the smoothing parameter is a continuous variable, a given range of smoothing parameter values corresponds to a finite set of local models. For example, for a data set with 100 observations, the range $[0.2, 0.4]$ corresponds to models with $20, 21, 22, \ldots, 40$ points in the local neighborhoods. If the GLOBAL modifier is specified, all possible models in the range are evaluated sequentially.

Note that by default PROC LOESS displays a "Fit Summary" and other optionally requested tables only for the selected model. You can request that these tables be displayed for all models in the selection process by adding the STEPS modifier in the SELECT= option. Also note that by default scoring requested with SCORE statements is done only for the selected model. However, if you specify the STEPS in both the MODEL and SCORE statements, then all models evaluated in the selection process are scored.

In terms of computation, AIC_C and GCV depend on the smoothing matrix L only through its trace. In the direct method, this trace can be computed efficiently. In the interpolated method using kd trees, there is some additional computational cost but the overall work is not significant compared to the rest of the computation. In contrast, the quantities δ_1, δ_2, and ν_1, which appear in the AIC_{C_1} criterion, depend on the entire L matrix and for this reason, the time needed to compute these quantities dominates the time required for the model fitting. Hence SELECT=AICC1 is much more computationally expensive than SELECT=AICC and SELECT=GCV, especially when combined with the GLOBAL modifier. Hurvich, Simonoff, and Tsai

(1998) note that $AICC$ can be regarded as an approximation of $AICC_1$ and that "the $AICC$ selector generally performs well in all circumstances."

For models with one dependent variable, PROC LOESS uses SELECT=AICC as its default, if you specify neither the SMOOTH= nor SELECT= options in the MODEL statement. With two or more dependent variables, automatic smoothing parameter selection needs to be done separately for each dependent variable. For this reason automatic smoothing parameter selection is not available for models with multiple dependent variables. In such cases you should use a separate PROC LOESS step for each dependent variable, if you want to use automatic smoothing parameter selection.

ODS Table Names

PROC LOESS assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. The following new tables have been added.

Table 6.2. ODS Tables Produced by PROC LOESS

ODS Table Name	Description	Statement	Option
ModelSummary	Summary of all models evaluated	MODEL	DETAILS(ModelSummary)
SmoothingCriterion	Criterion value and selected smoothing parameter value	MODEL	SELECT

Example

Example 6.1. Automatic Smoothing Parameter Selection

The following data set contains measurements of monthly averaged atmospheric pressure differences between Easter Island and Darwin, Australia, for a period of 168 months (NIST 1998):

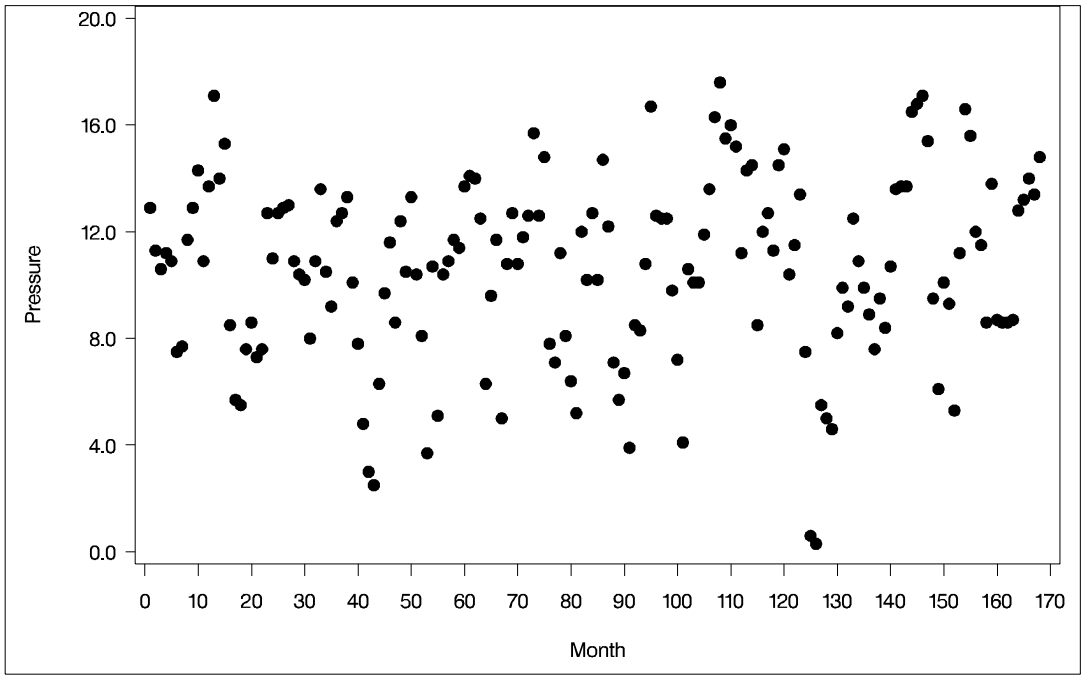
```
data ENSO;
  input Pressure @@;
  Month=_N_;
  format Pressure 4.1;
  format Month 3.0;
datalines;
12.9  11.3  10.6  11.2  10.9   7.5   7.7  11.7
12.9  14.3  10.9  13.7  17.1  14.0  15.3   8.5
 5.7   5.5   7.6   8.6   7.3   7.6  12.7  11.0
12.7  12.9  13.0  10.9  10.4  10.2   8.0  10.9
13.6  10.5   9.2  12.4  12.7  13.3  10.1   7.8
 4.8   3.0   2.5   6.3   9.7  11.6   8.6  12.4
10.5  13.3  10.4   8.1   3.7  10.7   5.1  10.4
```

```
10.9  11.7  11.4  13.7  14.1  14.0  12.5   6.3
 9.6  11.7   5.0  10.8  12.7  10.8  11.8  12.6
15.7  12.6  14.8   7.8   7.1  11.2   8.1   6.4
 5.2  12.0  10.2  12.7  10.2  14.7  12.2   7.1
 5.7   6.7   3.9   8.5   8.3  10.8  16.7  12.6
12.5  12.5   9.8   7.2   4.1  10.6  10.1  10.1
11.9  13.6  16.3  17.6  15.5  16.0  15.2  11.2
14.3  14.5   8.5  12.0  12.7  11.3  14.5  15.1
10.4  11.5  13.4   7.5   0.6   0.3   5.5   5.0
 4.6   8.2   9.9   9.2  12.5  10.9   9.9   8.9
 7.6   9.5   8.4  10.7  13.6  13.7  13.7  16.5
16.8  17.1  15.4   9.5   6.1  10.1   9.3   5.3
11.2  16.6  15.6  12.0  11.5   8.6  13.8   8.7
 8.6   8.6   8.7  12.8  13.2  14.0  13.4  14.8
;
```

The following PROC GPLOT statements produce the simple scatter plot of these data, displayed in Output 6.1.1:

```
symbol1 color=black value=dot ;
proc gplot data=ENSO;
   plot Pressure*Month /
        hminor = 0
        vminor = 0
        vaxis  = axis1
        frame cframe=ligr;
        axis1 label = ( r=0 a=90 ) order=(0 to 20  by 4);
run;
```

Output 6.1.1. Scatter Plot of ENSO Data



You can compute a loess fit and plot the results for these data using the following statements:

```
ods output OutputStatistics=ENSOstats;

proc loess data=ENSO;
   model Pressure=Month ;
run;

symbol1 color=black value=dot h=2.5 pct;
symbol2 color=black interpol=join value=none width=2;
proc gplot data=ENSOstats;
   plot (depvar pred)*Month / overlay
        hminor = 0
        vminor = 0
        vaxis  = axis1
        frame cframe=ligr;
        axis1 label = ( r=0 a=90 ) order=(0 to 20  by 4);
run; quit;
```

The "Smoothing Criterion" and "Fit Summary" tables are shown in Output 6.1.2 and the fit is plotted in Figure 6.1.3.

Output 6.1.2. Output from PROC LOESS

The LOESS Procedure
Dependent Variable: Pressure

Optimal Smoothing Criterion

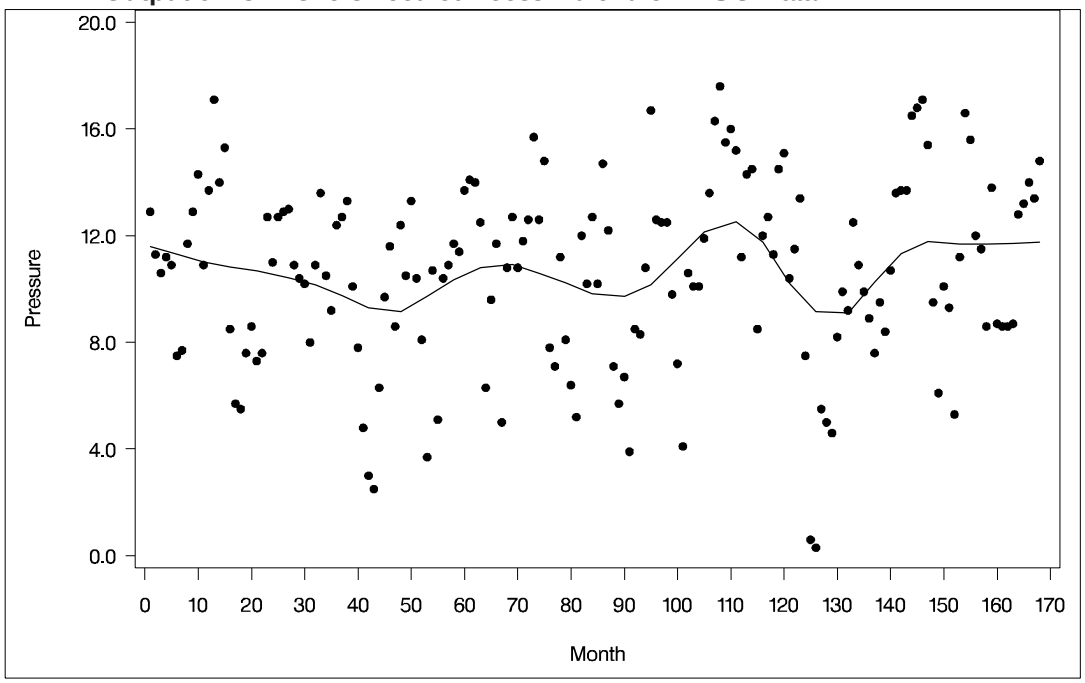
AICC	Smoothing Parameter
3.41105	0.22321

The LOESS Procedure
Selected Smoothing Parameter: 0.223
Dependent Variable: Pressure

Fit Summary

Fit Method	kd Tree
Blending	Linear
Number of Observations	168
Number of Fitting Points	33
kd Tree Bucket Size	7
Degree of Local Polynomials	1
Smoothing Parameter	0.22321
Points in Local Neighborhood	37
Residual Sum of Squares	1654.27725
Trace[L]	8.74180
GCV	0.06522
AICC	3.41105

Output 6.1.3. Oversmoothed Loess Fit for the ENSO Data



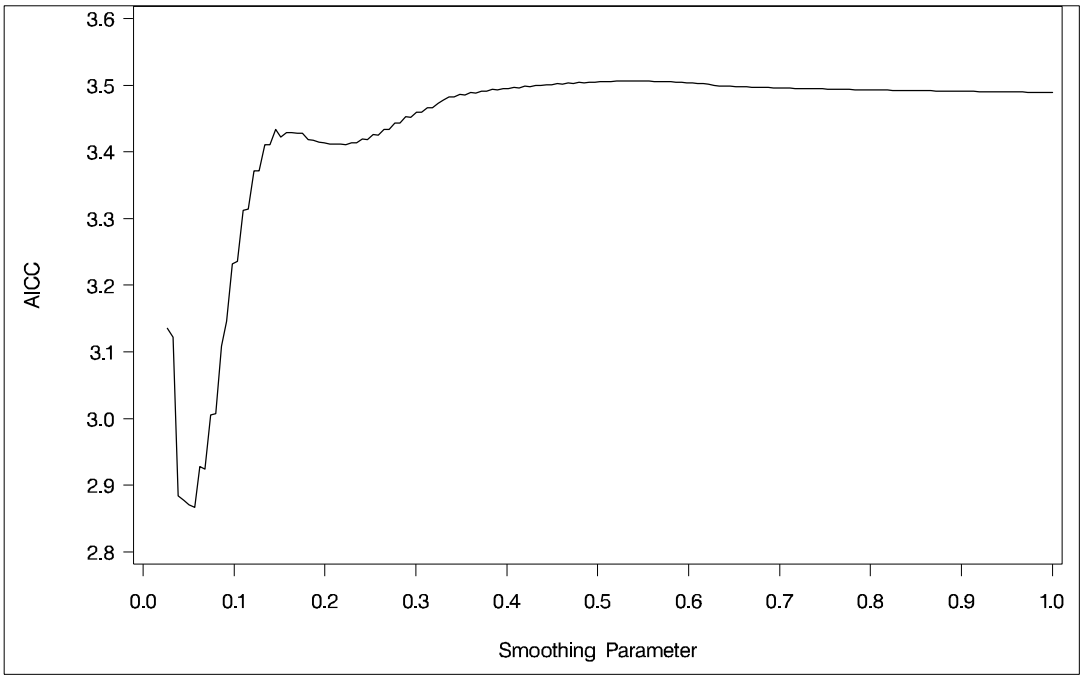
The smoothing parameter value used for the loess fit shown in Figure 6.1.3 was chosen using the default method of PROC LOESS, namely a golden section minimization of the AICC criterion over the interval $(0, 1]$. The fit seems to be oversmoothed. What accounts for this poor fit?

One possibility is that the golden section search has found a local rather than a global minimum of the AICC criterion. You can test this by redoing the fit requesting a global minimum. It is also helpful to plot the AICC criterion as a function of the smoothing parameter value used. You do this with the following statements:

```
ods output ModelSummary=ENSOsummary;

proc loess data=ENSO;
   model Pressure=Month/select=AICC(global);
run;

proc sort data=ENSOsummary;
    by smooth;
run;

symbol1 color=black interpol=join value=none width=2;
proc gplot data=ENSOsummary;
   format AICC f4.1;
   format smooth f4.1;
   plot AICC*Smooth /
        hminor = 0 vminor = 0
        vaxis  = axis1 frame cframe=ligr;
        axis1 label = ( r=0 a=90 );
run; quit;
```

The results are shown in Figure 6.1.4.

Output 6.1.4. AICC versus Smoothing Parameter Showing Local Minima

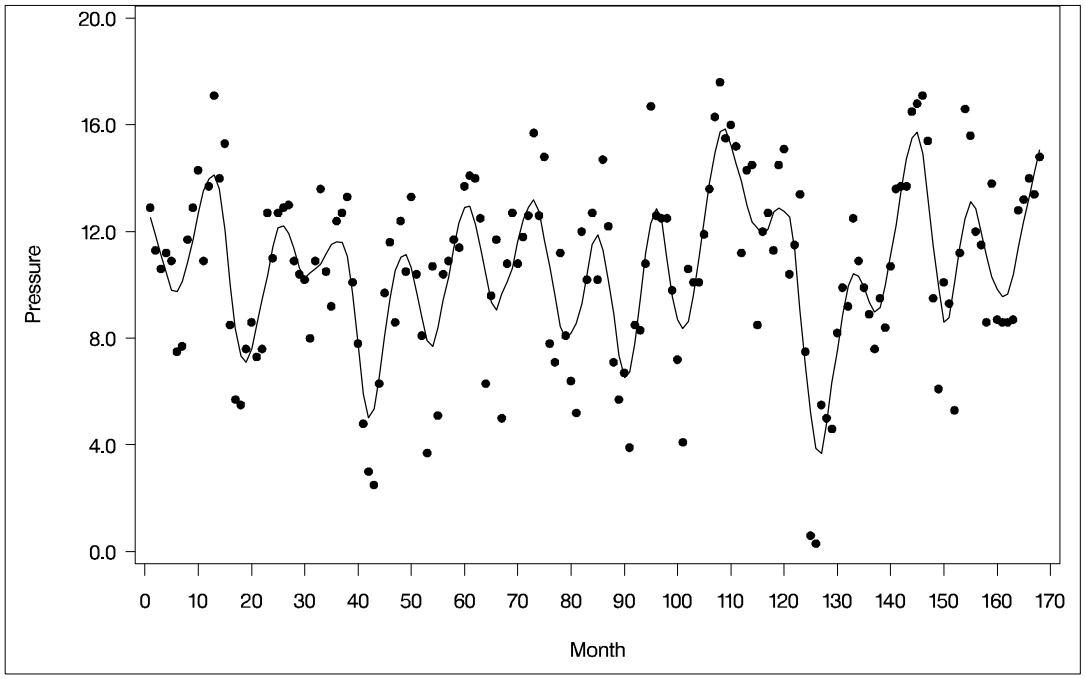


The explanation for the oversmoothed fit in Figure 6.1.3 is now apparent. The golden section search algorithm found the local minimum that occurs near the value 0.22 of the smoothing parameter rather than the global minimum that occurs near 0.06. Note that if you restrict the range of smoothing parameter values examined to lie below 0.2, then the golden section search finds the global minimum as the following statements demonstrate:

```
ods output OutputStatistics=ENSOstats;

proc loess data=ENSO;
   model Pressure=Month/select=AICC( range(0.03,0.2) );
run;

symbol1 color=black value=dot h=2.5 pct;
symbol2 color=black interpol=join value=none width=2;
proc gplot data=ENSOstats;
   plot (depvar pred)*Month / overlay
        hminor = 0
        vminor = 0
        vaxis  = axis1
        frame cframe=ligr;
        axis1 label = ( r=0 a=90 ) order=(0 to 20  by 4);
run; quit;
```

The fit obtained is shown in Figure 6.1.5.

Output 6.1.5. Loess Fit for the ENSO Data



The loess fit shown in Figure 6.1.5 clearly shows an annual cycle in the data. An interesting question is whether there is some phenomenon captured in the data that would explain the presence of the local minimum near 0.22 in the AICC curve. Note that there is some evidence of a cycle of about 42 months in the oversmoothed fit in Figure 6.1.3. You can see this cycle because the strong annual cycle in Figure 6.1.5 has been smoothed out. The physical phenomenon that accounts for the existence of this cycle has been identified as the periodic warming of the Pacific Ocean known as "El Niño."

References

Akaike, H. (1973), "Information Theory and an Extension of the Maximum Likelihood Principle," in *Proceedings of the Second International Symposium on Information Theory,* eds. Petrov and Csaki, 267-281.

Craven, P. and Wahba, G. (1979), "Smoothing Noisy Data with Spline Functions," *Numerical Mathematics*, 31, 377–403.

Gordon, W.J. (1971) "Blending-function Methods of Bivariate and Multivariate Interpolation and Approximation," *SIAM Journal of Numerical Analysis*, 8, No. 1, 158–177.

Hurvich, C.M., Simonoff, J.S., and Tsai, C.L. (1998), "Smoothing Parameter Selection in Nonparametric Regression Using an Improved Akaike Information Criterion" *Journal of the Royal Statistical Society B*, 60, 271–293.

NIST (1998), "Statistical Reference Data Sets," [http://www.nist.gov/itl/div898/strd], accessed 20 January 1999.

Ruppert, D., Sheather, S.J., and Wand, M.P. (1995), "An Effective Bandwidth Selector for Local Least Squares Regression," *Journal of the American Statistical Association*, 90, 1257–1270.

Chapter 7
The LOGISTIC Procedure

Chapter Table of Contents

OVERVIEW 91

SYNTAX 91
PROC LOGISTIC Statement 91
EXACT Statement 92

DETAILS 94
Exact Conditional Analysis 94
OUTDIST= Output Data Set 98
Displayed Output 99
ODS Table Names 100

EXAMPLES 101
Example 7.1 Dose-Response Study 101
Example 7.2 Crossover Clinical Trial 105

REFERENCES 106

Chapter 7
The LOGISTIC Procedure

Overview

Exact conditional inference capability for binary (dichotomous) response variables is now available in the LOGISTIC procedure. Exact methods can be useful for analyzing data sets for which the usual asymptotic assumptions are inadequate; for example, when sample sizes are small or the data are sparse or skewed.

Syntax

The additional syntax to the LOGISTIC procedure consists of one or more EXACT statements for specifying the effects to be analyzed and the EXACTONLY and (global) EXACTOPTIONS options in the PROC LOGISTIC statement. You still need the MODEL statement to specify the effects in the model. Loosely speaking, the exact inference on the effects in the EXACT statement are conditional on all other effects in the MODEL statement.

PROC LOGISTIC Statement

The following options are added to the PROC LOGISTIC statement:

EXACTONLY
requests only the exact analyses. The unconditional likelihood analysis that PROC LOGISTIC usually performs is suppressed.

EXACTOPTIONS(*options***)**
specifies options that apply to every EXACT statement in the program. The following options are available:

MAXTIME=*seconds* specifies the maximum clock time (in seconds) that PROC LOGISTIC can use to calculate the permutation distributions. If the limit is exceeded, the procedure halts all computations. The default maximum clock time is seven days.

STATUSTIME=*seconds* specifies the time interval (in seconds) for printing a status line in the SAS Log. You can use this status line to track the progress of the computation of the exact conditional distributions. The time interval you specify is a lower limit; the actual time interval may vary for larger problems. By default, no status reports are produced.

EXACT Statement

EXACT <*'label'*>< *Intercept* >< *effects* >< / *options* > **;**

The EXACT statement performs the exact tests of the parameters for the specified effects and optionally estimates the parameters and outputs the exact conditional distributions. You can specify the keyword INTERCEPT and any effects in the MODEL statement. Inference on parameters for the specified effects is performed by conditioning on the sufficient statistics of all the other model parameters (possibly including the intercept).

You can specify several EXACT statements, but they must follow the MODEL statement. Each statement can optionally include an identifying label. If several EXACT statements are specified, any statement without a label is assigned a label of the form "Exact$n$", where "n" indicates the nth EXACT statement. The label is included in the headers of the exact analysis tables.

The CONTRAST, TEST, and UNITS statements do not apply to the exact analyses. Exact analyses are not performed when you specify a WEIGHT statement, a link other than LINK=LOGIT, an offset variable, the NOFIT option, or a model-selection method.

The effects specified in the MODEL statement can have at most 32 parameters. If you have too many observations with the same covariate values, an integer overflow may occur and computations will cease.

The following options can be specified in each EXACT statement after a slash (/):

ALPHA=*p*
specifies the level of significance p for $100(1-p)$% confidence limits for the parameters or odds ratios. The value p must be between 0 and 1. If the ALPHA= option is not specified, p is equal to 0.05 or the value of the ALPHA= option in the PROC LOGISTIC statement.

ESTIMATE < **=***keyword* >
estimates the individual parameters (conditional on all other parameters) for the effects specified in the EXACT statement. For each parameter, a point estimate, a confidence interval, and a p-value for a two-sided test that the parameter is zero are displayed. Note that the two-sided p-value is twice the one-sided p-value. You can optionally specify one of the following keywords:

PARM specifies that the parameters be estimated. This is the default.

ODDS specifies that the odds ratios be estimated. For classification variables, use of the reference parameterization (the PARAM=REF option in the CLASS statement) is recommended.

BOTH specifies that the parameters and odds ratios be estimated.

JOINT
performs the joint test that all of the parameters for the EXACT statement are simul-

taneously equal to zero along with tests that the parameters of the individual variables are zero. The joint test of all the parameters is indicated in the "Conditional Exact Tests" table by the label "Joint."

JOINTONLY
performs only the joint test that all of the parameters for the EXACT statement are simultaneously equal to zero. The test is indicated in the "Conditional Exact Tests" table by the label "Joint." When this option is specified, tests for the parameters of the individual variables are not performed.

ONESIDED
requests one-sided confidence intervals and p-values for the individual parameter estimates and odds ratios. The one-sided p-value is the smaller of the left and right tail probabilities for the observed sufficient statistic for the parameter under the null hypothesis that the parameter is zero. The two-sided p-values (default) are twice the one-sided p-values.

OUTDIST=*SAS-data-set*
names the SAS data set containing the exact conditional distributions. This data set contains all of the exact conditional distributions required to process the corresponding EXACT statement. The data set contains the possible sufficient statistics for the parameters of the effects specified in the EXACT statement, the counts, and, when hypothesis tests are performed on the parameters, the probability of occurrence and the score value for each sufficient statistic. When you request an OUTDIST= data set, the observed sufficient statistics are displayed in the "Sufficient Statistics" table. See the "OUTDIST= Output Data Set" section on page 98 for more information.

EXACT Statement Examples

- In the following example, two exact tests are computed: one is for x1 and the other is for x2. The test for x1 is based on the exact conditional distribution of the sufficient statistic for the x1 parameter given the observed values of the sufficient statistics for the intercept, x2, and x3 parameters; likewise, the test for x2 is conditional on the observed sufficient statistics for the intercept, x1, and x3:

```
proc logistic;
   model y = x1 x2 x3;
   exact x1 x2;
run;
```

- If x3 is a CLASS variable with three levels and the reference parameterization is specified, then there are two associated parameters. If you submit the C13 statement, an individual test that the parameter of x1 is zero and a joint test for testing that the two parameters of x3 are both zero is performed. If you specify the CJ13 statement, both of these tests and a joint test that all the parameters for the EXACT statement are zero are performed. If you request the CE3 statement, then a joint test for testing that the two parameters of x3 are both zero is performed, and estimates for each of the two parameters are produced:

```
class x3 / param=ref;
exact 'C13'  x1 x3;
exact 'CJ13' x1 x3 / joint;
exact 'CE3'  x3    / estimate;
```

- You can specify multiple EXACT statements in the same PROC LOGISTIC invocation. PROC LOGISTIC determines, from all the EXACT statements, the distinct conditional distributions that need to be evaluated. For example, only one exact conditional distribution is required for the following two EXACT statements:

```
exact x1 / estimate=parm;
exact x1 / estimate=parm onesided;
```

- For each EXACT statement, individual tests for the parameters of the specified effects are computed unless the JOINTONLY option is specified. Consider the following EXACT statements:

```
exact 'E12' x1 x2 / estimate;
exact 'E1'  x1    / estimate;
exact 'E2'  x2    / estimate;
exact 'J12' x1 x2 / joint;
```

  In the E12 statement, the parameters for x1 and x2 are estimated and tested separately. Specifying the E12 statement is equivalent to specifying both the E1 and E2 statements. In the J12 statement, the joint test for the parameters of x1 and x2 is computed as well as the individual tests for x1 and x2.

- All exact conditional distributions for the tests and estimates computed in a single EXACT statement are output to the corresponding OUTDIST= data set. For example, consider the following EXACT statements:

```
exact 'O1'   x1    /           outdist=o1;
exact 'OJ12' x1 x2 / jointonly outdist=oj12;
exact 'OA12' x1 x2 / joint     outdist=oa12;
exact 'OE12' x1 x2 / estimate  outdist=oe12;
```

  The O1 statement outputs a single exact conditional distribution. The OJ12 statement outputs only the joint distribution for x1 and x2. The OA12 statement outputs three conditional distributions: one for x1, one for x2, and one jointly for x1 and x2. The OE12 statement outputs two conditional distributions: one for x1 and the other for x2. Data set oe12 contains both the x1 and x2 variables; the distribution for x1 has missing values in the x2 column while the distribution for x2 has missing values in the x1 column.

  See the "OUTDIST= Output Data Set" section on page 98 for more information.

Details

Exact Conditional Analysis

Asymptotic methods may be inadequate when sample sizes are small or the data are sparse or skewed. In such cases, the asymptotic p-values are not close approximations to the true p-values. Exact conditional inference remains valid in such situations.

The following sections summarize the unconditional and conditional analyses, discuss the network algorithm used in the computations and the necessary computational resources, and provide details on the statistics displayed in the output tables.

Background

The theory of exact conditional logistic regression analysis was originally laid out by Cox (1970). Other useful references include Cox and Snell (1989), Agresti (1990, 1992), and Mehta and Patel (1995).

Consider N independent Bernoulli random variables $Y_1, ..., Y_N$ having observed values $\mathbf{y}_0 = (y_{01}, ..., y_{0N})'$. For each observation $i = 1, ..., N$, let $\mathbf{x}_i = (x_{i1}, ..., x_{in}, x_{i,n+1}, ..., x_{i,n+m})'$ be an $n + m = s$ vector of explanatory variables, and denote $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_N)'$. Let $p_i = p(\mathbf{x}_i) = \Pr(Y_i = 1|\mathbf{x}_i)$ be the event probability for each $i = 1, ..., N$, and denote $\mathbf{p} = (p_1, ..., p_N)'$. Then the logistic regression model is $\text{logit}(\mathbf{p}) = \mathbf{X}\boldsymbol{\beta}$, or

$$\text{logit}(p_i) = \log\left(\frac{p_i}{1 - p_i}\right) = \mathbf{x}_i'\boldsymbol{\beta}$$

where $\boldsymbol{\beta} = (\beta_1, ..., \beta_s)'$ is the unknown parameter vector.

Unconditional likelihood inference is based on maximizing the likelihood function:

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{N} p_i^{y_{0i}}(1 - p_i)^{1-y_{0i}} = \frac{\exp(\mathbf{y}_0'\mathbf{X}\boldsymbol{\beta})}{\prod_{i=1}^{N}[1 + \exp(\mathbf{x}_i'\boldsymbol{\beta})]}$$

To perform conditional inference, first observe that the sufficient statistics for the β_j in the unconditional likelihood function are the corresponding $T_j = \sum_{i=1}^{N} y_i x_{ij}$, where y_i is a realization of Y_i. To create the probability density function (pdf) for $\mathbf{T} = (T_1, ..., T_s)'$, sum over all binary sequences $\mathbf{y}$ that generate an observable $\mathbf{t}$:

$$\Pr(\mathbf{T} = \mathbf{t}) = \frac{C(\mathbf{t})\exp(\mathbf{t}'\boldsymbol{\beta})}{\prod_{i=1}^{N}[1 + \exp(\mathbf{x}_i'\boldsymbol{\beta})]}$$

where $C(\mathbf{t}) = ||\{\mathbf{y} : \mathbf{y}'\mathbf{X} = \mathbf{t}'\}||$ is the number of sequences $\mathbf{y}$ that generate $\mathbf{t}$. Suppose the m parameters $\boldsymbol{\beta}_2 = (\beta_{n+1}, ..., \beta_{n+m})'$ are *nuisance* parameters; that is, the current analysis is geared toward the first n parameters $\boldsymbol{\beta}_1 = (\beta_1, ..., \beta_n)'$. Denote the sufficient statistics for the parameters of interest as $\mathbf{T}_1 = (T_1, ..., T_n)'$, the corresponding observed values as $\mathbf{t}_1$, and the corresponding columns of $\mathbf{X}$ as $\mathbf{X}_1$. Similarly, define $\mathbf{T}_2$, $\mathbf{t}_2$, and $\mathbf{X}_2$ for the nuisance parameters. The nuisance parameters can be removed from the analysis by conditioning on their sufficient statistics to create the conditional likelihood

$$\Pr(\mathbf{T}_1 = \mathbf{t}_1|\mathbf{T}_2 = \mathbf{t}_2) = \frac{\Pr(\mathbf{T} = \mathbf{t})}{\Pr(\mathbf{T}_2 = \mathbf{t}_2)} = \frac{C(\mathbf{t})\exp(\mathbf{t}_1'\boldsymbol{\beta}_1)}{\sum_{\mathbf{u}} C(\mathbf{u}, \mathbf{t}_2)\exp(\mathbf{u}'\boldsymbol{\beta}_1)}$$

where $C(\mathbf{u}, \mathbf{t}_2)$ is the number of vectors $\mathbf{y}$ such that $\mathbf{y}'\mathbf{X}_1 = \mathbf{u}$ and $\mathbf{y}'\mathbf{X}_2 = \mathbf{t}_2$.

Conditional asymptotic inference is performed by maximizing the conditional likelihood and producing conditional statistics similar to those for the unconditional likelihood case.

Conditional exact inference is based on generating the conditional distribution for the parameters of interest. This distribution is called the *permutation* or *exact conditional* distribution. The conditional pdf $\Pr(\mathbf{T}_1 = \mathbf{t}_1 | \mathbf{T}_2 = \mathbf{t}_2)$ is denoted as $f_{\beta_1}(\mathbf{t}_1 | \mathbf{t}_2)$.

Computational Algorithm

The goal of the exact conditional analysis is to determine how likely the observed response $\mathbf{y}_0$ is with respect to all 2^N possible responses $\mathbf{y} = (y_1, ..., y_N)'$. One way to proceed is to generate every $\mathbf{y}$ vector for which $\mathbf{y}'\mathbf{X}_2 = \mathbf{t}_2$ and count the number of vectors $\mathbf{y}$ for which $\mathbf{y}'\mathbf{X}_1$ is equal to each unique $\mathbf{t}_1$. Generating the conditional distribution from complete enumeration of the joint distribution is conceptually simple; however, this method becomes computationally infeasible very quickly. For example, if you had only 30 observations, you'd have to scan through 2^{30} different $\mathbf{y}$ vectors.

PROC LOGISTIC employs a network algorithm developed by Hirji, Mehta, and Patel (1987) to generate and count the $\mathbf{y}$ vectors. The algorithm is based on the following observation: given any $\mathbf{y} = (y_1, ..., y_N)'$ and a design $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_N)'$, let $\mathbf{y}_{(i)} = (y_1, ..., y_i)'$ and $\mathbf{X}_{(i)} = (\mathbf{x}_1, ..., \mathbf{x}_i)' = \begin{pmatrix} x_{1,1} & ... & x_{1,s} \\ \vdots & & \vdots \\ x_{i,1} & ... & x_{i,s} \end{pmatrix}$ be the first i rows of each matrix. Write the sufficient statistic based on these i rows as $\mathbf{t}'_{(i)} = \mathbf{y}'_{(i)}\mathbf{X}_{(i)}$. A recursion relation results: $\mathbf{t}_{(i+1)} = \mathbf{t}_{(i)} + y_{i+1}\mathbf{x}_{i+1}$. Combining this relation with a method of determining which $\mathbf{y}$ vectors produce the $\mathbf{t}_2$ margins makes the generation of the permutation distribution feasible.

The bulk of the computation time and memory is consumed by the creation of the exact joint distribution. After the joint distribution for a set of effects is created, the computational effort required to produce hypothesis tests and parameter estimates for any subset of the effects is (relatively) trivial.

Computational Resources

PROC LOGISTIC uses a relatively fast and efficient algorithm for the exact analyses. This recently developed algorithm, together with improvements in computer power, now make it feasible to perform exact computations for data sets where previously only asymptotic methods could be applied. Nevertheless, there are still large problems that may require a prohibitive amount of time and memory for exact computations, depending on the speed and memory available on your computer. For large problems, consider whether exact methods are really needed or whether asymptotic methods might give results quite close to the exact results, while requiring much less computer time and memory.

A formula does not exist that can predict the amount of time and memory necessary to generate the permutation distributions for a particular problem. The time and memory required depend on several factors, including the total sample size, the number of parameters of interest, the number of nuisance parameters, and the order in which the parameters are processed. If you run out of memory, refer to the SAS Companion for your system to see how to allocate more.

You can use the MAXTIME= option in the EXACTOPTIONS option to limit the total amount of time PROC LOGISTIC uses to derive all of the exact distributions. If PROC LOGISTIC does not finish within that time, the procedure terminates. If you

need to derive several distributions, it may be more feasible to request one distribution at a time.

At any time while PROC LOGISTIC is deriving the distributions, you can terminate the computations by pressing the system interrupt key sequence (refer to the SAS Companion for your system) and choosing to stop computations.

Hypothesis Tests

Consider testing the null hypothesis H_0: $\boldsymbol{\beta}_1 = \mathbf{0}$ against the alternative H_A: $\boldsymbol{\beta}_1 \neq \mathbf{0}$, conditional on $\mathbf{T}_2 = \mathbf{t}_2$. Under the null hypothesis, the test statistic for the *exact probability test* is just $f_{\beta_1=0}(\mathbf{t}_1|\mathbf{t}_2)$, while the corresponding p-value is the probability of getting a less likely (more extreme) statistic,

$$p(\mathbf{t}_1|\mathbf{t}_2) = \sum_{u \in \Omega_p} f_0(\mathbf{u}|\mathbf{t}_2)$$

where $\Omega_p = \{\mathbf{u}$: for some $\mathbf{y}$, $\mathbf{y}'\mathbf{X}_1 = \mathbf{u}$, $\mathbf{y}'\mathbf{X}_2 = \mathbf{t}_2$, and $f_0(\mathbf{u}|\mathbf{t}_2) \leq f_0(\mathbf{t}_1|\mathbf{t}_2)\}$.

For the *exact conditional scores test*, the conditional mean $\boldsymbol{\mu}_1$ and variance matrix $\boldsymbol{\Sigma}_1$ of the $\mathbf{T}_1$ (conditional on $\mathbf{T}_2 = \mathbf{t}_2$) are calculated, and the score statistic for the observed value,

$$s = (\mathbf{t}_1 - \boldsymbol{\mu}_1)'\boldsymbol{\Sigma}_1^{-1}(\mathbf{t}_1 - \boldsymbol{\mu}_1)$$

is compared to the score for each member of the distribution

$$S(\mathbf{T}_1) = (\mathbf{T}_1 - \boldsymbol{\mu}_1)'\boldsymbol{\Sigma}_1^{-1}(\mathbf{T}_1 - \boldsymbol{\mu}_1)$$

The resulting p-value is

$$p(\mathbf{t}_1|\mathbf{t}_2) = Pr(S \geq s) = \sum_{u \in \Omega_s} f_0(\mathbf{u}|\mathbf{t}_2)$$

where $\Omega_s = \{\mathbf{u}$: for some $\mathbf{y}$, $\mathbf{y}'\mathbf{X}_1 = \mathbf{u}$, $\mathbf{y}'\mathbf{X}_2 = \mathbf{t}_2$, and $S(\mathbf{u}) \geq s\}$.

The mid-p statistic was originally proposed by Lancaster (1961) to compensate for the discreteness of the distribution. Hirji, Tsiatis, and Mehta (1989) recommend its use with small or sparse data sets; Vollset, Hirji, and Afifi (1991) suggest the statistic for matched case-control studies; and Hirji and Tang (1998) recommend the statistic for tests of trend. The mid-p is defined as

$$\frac{1}{2} f_{\beta_1}(\mathbf{t}_1|\mathbf{t}_2) + \sum_{u \in \Omega} f_{\beta_1}(\mathbf{u}|\mathbf{t}_2)$$

where Ω is either Ω_s or Ω_p except that strict inequalities are used.

Inference for a Single Parameter

Exact parameter estimates are derived for a single parameter β_i by regarding all the other parameters $\boldsymbol{\beta}_2 = (\beta_1, ..., \beta_{i-1}, \beta_{i+1}, ..., \beta_s)'$ as nuisance parameters. The appropriate sufficient statistics are $\mathbf{T}_1 = T_i$ and $\mathbf{T}_2 = (T_1, ..., T_{i-1}, T_{i+1}, ..., T_s)'$, with

their observed values denoted by the lowercase t. Hence, the conditional pdf used to create the parameter estimate for β_i is

$$f_{\beta_i}(t_i|\mathbf{t}_2) = \frac{C(t_i, \mathbf{t}_2)\exp(t_i\beta_i)}{\sum_{u\in\Omega} C(u, \mathbf{t}_2)\exp(u\beta_i)}$$

for $\Omega = \{u\text{: for some } \mathbf{y}, T_i = u \text{ and } \mathbf{T}_2 = \mathbf{t}_2\}$.

The maximum exact conditional likelihood estimate is the quantity $\widehat{\beta}_i$ which maximizes the conditional pdf. A Newton-Raphson algorithm is used to perform this search. However, if the observed t_i attains either its minimum or maximum value in the permutation distribution (that is, either $t_i = \min\{u : u \in \Omega\}$ or $t_i = \max\{u : u \in \Omega\}$), then the conditional pdf is monotonically increasing in β_i and cannot be maximized. In this case, a median unbiased estimate (Hirji, Tsiatis, and Mehta 1989; Hirji and Tang 1998) $\widehat{\beta}_i$ is produced that satisfies $\sum_u f_{\widehat{\beta}_i}(u|\mathbf{t}_2) = \frac{1}{2}$, where the sum is over all possible values of T_i, and a Newton-Raphson-type algorithm is used to perform the search.

Likelihood ratio tests based on the conditional pdf are used to test the null H_0: $\beta_i = 0$ against various alternatives. For testing against the alternative H_A: $\beta_i > 0$, the critical region for the UMP test consists of the upper tail of values for T_i in the permutation distribution. Thus, the one-sided significance level $p_G(t_i; 0)$ is the probability of a more extreme (greater) value:

$$p_G(t_i; 0) = \sum_{u\geq t_i} f_0(u|\mathbf{t}_2)$$

The one-sided significance level $p_L(t_i; 0)$ against H_A: $\beta_i < 0$ is

$$p_L(t_i; 0) = \sum_{u\leq t_i} f_0(u|\mathbf{t}_2)$$

The minimum of these one-sided levels is reported when the ONESIDED option is specified. The two-sided significance level $p(t_i; 0)$ against H_A: $\beta_i \neq 0$ is calculated as

$$p(t_i; 0) = 2\min[p_L(t_i; 0), p_G(t_i; 0)]$$

An upper $100(1-2\epsilon)$% confidence limit for $\widehat{\beta}_i$ corresponding to the observed t_i is the solution $\beta_U(t_i)$ of $\epsilon = p_L(t_i, \beta_U(t_i))$, while the lower confidence limit is the solution $\beta_L(t_i)$ of $\epsilon = p_G(t_i, \beta_L(t_i))$. A Newton-Raphson procedure is used to search for the solutions.

OUTDIST= Output Data Set

The OUTDIST= data set contains every exact conditional distribution necessary to process the EXACT statement. For example, the following statements create one distribution for the x1 parameter and another for the x2 parameters, and produces the data set dist shown in the following table:

```
proc logistic;
   class x2 / param=ref;
   model y=x1 x2;
   exact x1 x2 / outdist=dist;
proc print data=dist;
run;
```

Obs	x1	x20	x21	Count	Score	Prob
1	.	0	0	3	5.81151	0.03333
2	.	0	1	15	1.66031	0.16667
3	.	0	2	9	3.12728	0.10000
4	.	1	0	15	1.46523	0.16667
5	.	1	1	18	0.21675	0.20000
6	.	1	2	6	4.58644	0.06667
7	.	2	0	19	1.61869	0.21111
8	.	2	1	2	3.27293	0.02222
9	.	3	0	3	6.27189	0.03333
10	2	.	.	6	3.03030	0.12000
11	3	.	.	12	0.75758	0.24000
12	4	.	.	11	0.00000	0.22000
13	5	.	.	18	0.75758	0.36000
14	6	.	.	3	3.03030	0.06000

The first nine observations in the dist data set contain a permutation distribution for the parameters of the x2 effect (hence the values for the x1 parameter are missing), and the remaining five observations are for the x1 parameter. If a joint distribution was created, there would be observations with values for both the x1 and x2 parameters. For CLASS variables, the corresponding parameters in the dist data set are identified by concatenating the variable name with the appropriate classification level.

The data set contains the possible sufficient statistics of the parameters for the effects specified in the EXACT statement, and the Count variable contains the number of different responses that yield these statistics. For example, there were six possible response vectors $\mathbf{y}$ for which the product $\mathbf{y}'\mathbf{x1}$ was equal to 2, and for which $\mathbf{y}'\mathbf{x20}$, $\mathbf{y}'\mathbf{x21}$, and $\mathbf{y}'\mathbf{1}$ were equal to their actual observed values (displayed in the "Sufficient Statistics" table). When hypothesis tests are performed on the parameters, the Prob variable contains the probability of obtaining that statistic (which is just the count divided by the total count), and the Score variable contains the score for that statistic. For more information, see the "EXACT Statement Examples" section on page 93.

Displayed Output

The displayed exact conditional analysis output of the LOGISTIC procedure includes the following:

- the "Conditional Exact Tests" table provides two tests for the null hypothesis that the parameters for the specified effects are zero: the exact probability test and the exact conditional scores test. For each test, the test statistic, an exact p-value (the probability of obtaining a more extreme statistic than the observed,

assuming the null hypothesis), and a mid p-value (adjusts for the discreteness of the distribution) are displayed.

Individual hypothesis tests for the parameter of each continuous effect, and joint tests for the parameters of classification variables, are generated by default. A joint test for all effects may be requested with the JOINT or JOINTONLY options.

- if you specify the ESTIMATE, ESTIMATE=PARM, or ESTIMATE=BOTH options, the "Exact Parameter Estimates" table displays individual parameter estimates for each parameter conditional on the values of all the other parameters in the model. These are either the exact conditional maximum likelihood estimates (MLE) or, in cases where the conditional MLE does not exist, the median unbiased estimates.

  Also displayed are one-sided or two-sided confidence limits for the estimate, and a one-sided or two-sided p-value for testing that the parameter estimate is zero. The one-sided p-value is the smaller of the left and right tail probabilities for the observed sufficient statistic for the parameter under the null hypothesis that the parameter is zero, while the two-sided p-value is twice the one-sided p-value.
- if you specify the ESTIMATE=ODDS or ESTIMATE=BOTH options, the "Exact Odds Ratios" table displays odds ratios for individual parameters, confidence limits, and a p-value for testing that the odds ratio is 1.
- if you request an OUTDIST= data set, the "Sufficient Statistics" table is displayed before printing any of the exact analysis results. The table lists the parameters and their observed sufficient statistics.

ODS Table Names

PROC LOGISTIC assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. The names for the exact conditional analyses are listed in the following table.

Table 7.1. ODS Tables Produced in PROC LOGISTIC

ODS Table Name	Description	Statement	Option
ExactTests	Conditional Exact Tests	EXACT	default
ExactParmEst	Parameter Estimates	EXACT	ESTIMATE, ESTIMATE=PARM, ESTIMATE=BOTH
ExactOddsRatio	Exact Odds Ratios	EXACT	ESTIMATE=ODDS, ESTIMATE=BOTH
SuffStats	Sufficient Statistics	EXACT	OUTDIST=

Examples

The following examples illustrate different types of exact analysis. Example 7.1 illustrates how to use exact conditional analysis and how to adjust for within-strata correlation. It also discusses how the MLE for the unconditional likelihood analysis may not exist, rendering the asymptotic inference impossible, while the exact conditional inference is still plausible. Example 7.2 is a phase II analysis for the pharmaceutical industry.

Example 7.1. Dose-Response Study

Researchers are interested in analyzing how mortality rates change with respect to dosage of a drug. The data contains life/death outcomes for six levels of drug dosage (0 to 5). One subject in each of three different research centers (A to C) is given a specific dose of the drug, and the Response is recorded as 0 for survival and 1 for death; the Z variable will be used in a later analysis:

```
data dose;
   input Center $ Dose Response @@;
   Z = 2 - Response;
   datalines;
A 0 0    B 0 0    C 0 0
A 1 0    B 1 0    C 1 0
A 2 0    B 2 0    C 2 0
A 3 0    B 3 0    C 3 0
A 4 1    B 4 0    C 4 0
A 5 1    B 5 0    C 5 1
;
```

Since all of the cells have counts that are less than 5, the applicability of large sample theory is questionable. For each subject i receiving dosage x_i, $i = 1, ..., 18$, let $Y_i = 1$ if the subject died, $Y_i = 0$ otherwise, and $\pi_i = \Pr(Y_i = 1|x_i)$. Then the linear logistic model for this problem is $\text{logit}(\pi_i) = \log\left(\frac{\pi_i}{1-\pi_i}\right) = \alpha + x_i\beta$, which fits a common intercept and slope for the i subjects. In the PROC LOGISTIC invocation below, the EXACT statement requests an exact analysis and the ESTIMATE option produces exact parameter estimates:

```
proc logistic data=dose descending;
   model Response = Dose;
   exact Dose / estimate=both;
run;
```

Output 7.1.1 displays some of the unconditional asymptotic results that are produced by default. The likelihood ratio and score tests reject the null hypothesis that β is zero. However, the Wald test does not reject this null hypothesis. The conflicting conclusions of these tests are a telltale sign that the large sample approximation is unreliable. The estimates for the intercept α and the slope β both have p-values greater than 0.05, indicating marginal influence. The confidence limits for the odds

ratio of the dose parameter contains the value 1, from which you could conclude, if you accept the model, that there is no change in mortality with a change in dosage.

Output 7.1.1. Output from Asymptotic Analysis

The LOGISTIC Procedure

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	8.1478	1	0.0043
Score	5.7943	1	0.0161
Wald	2.7249	1	0.0988

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept	1	-9.4745	5.5677	2.8958	0.0888
Dose	1	2.0804	1.2603	2.7249	0.0988

Odds Ratio Estimates

Effect	Point Estimate	95% Wald Confidence Limits	
Dose	8.007	0.677	94.679

Output 7.1.2 shows the results from the exact conditional analysis. The p-values in the "Conditional Exact Tests" table lead to rejecting the null hypothesis that β is zero (no conclusions can be made about α since it is "conditioned" away). The "Exact Parameter Estimates" table shows that the slope β is estimated to be $\widehat{\beta} = 1.8$, and since the 95% confidence interval for the exponential of $\widehat{\beta}$ does not contain 1, the odds of death increase significantly with dosage. Note that the exact tests do not produce standard errors for the estimates.

Output 7.1.2. Output from EXACT Analysis

The LOGISTIC Procedure

Exact Conditional Analysis

Conditional Exact Tests

Effect	Test	Statistic	--- p-Value --- Exact	Mid
Dose	Score	5.4724	0.0245	0.0190
	Probability	0.0110	0.0245	0.0190

Exact Parameter Estimates

Parameter	Estimate	95% Confidence Limits		p-Value
Dose	1.7999	0.1157	5.8665	0.0245

Exact Odds Ratios

Parameter	Estimate	95% Confidence Limits		p-Value
Dose	6.049	1.123	353.000	0.0245

The unconditional asymptotic and conditional exact results produce conflicting conclusions for this example. Stokes, Davis, and Koch (1995) recommend looking at the exact results when sample sizes are small and the approximate p-values are less than 0.10. For this example, the small sample size and the conflicting results for the asymptotic hypothesis tests indicate that an exact analysis is appropriate.

You can also perform a stratified analysis to control for the research centers. The strata are treated as nuisance parameters and a conditional likelihood removes them from the analysis. Your model contains a different intercept term for each stratum:

$$\text{logit}(\pi_{hi}) = \alpha_h + \mathbf{x}_{hi}\boldsymbol{\beta}$$

where h indexes the strata, α_h are the strata intercepts, and i indexes the subjects within the strata.

With PROC LOGISTIC, you can specify a stratification variable by including it in the CLASS statement. For example, a stratification variable that has three levels can be parameterized as

Stratum	Level 1	Level 2
1	1	0
2	0	1
3	0	0

where the usual intercept term represents the last strata level, and the other strata levels are a combination of the intercept and the appropriate level term. This is defined in the CLASS statement with the PARAM=REF option.

You can perform a stratified analysis by using the following statements, where Center is defined to be a classification variable and is conditioned out of the analysis by specifying only Dose in the EXACT statement.

```
proc logistic data=dose descending;
   class Center / param=ref;
   model Response = Center Dose;
   exact Dose / estimate=both;
run;
```

The usual asymptotic analysis indicates that there is complete separation of the data. (This means that unique maximum likelihood estimates do not exist.) You can see that the parameter estimates do not seem to converge if you specify both the ITPRINT and NOCHECK options in the MODEL statement. However, the exact analysis is still valid in this case, and exact tests and estimates for the conditional analysis are computed and displayed in Output 7.1.3.

Output 7.1.3. Stratified Output from EXACT Analysis

The LOGISTIC Procedure

Exact Conditional Analysis

Conditional Exact Tests

Effect	Test	Statistic	--- p-Value --- Exact	Mid
Dose	Score	5.5714	0.0222	0.0167
	Probability	0.0111	0.0222	0.0167

Exact Parameter Estimates

Parameter	Estimate	95% Confidence Limits		p-Value
Dose	1.3204*	0.1354	Infinity	0.0222

NOTE: * indicates a median unbiased estimate.

Exact Odds Ratios

Parameter	Estimate	95% Confidence Limits		p-Value
Dose	3.745*	1.145	Infinity	0.0222

NOTE: * indicates a median unbiased estimate.

The median unbiased estimate is created instead of the conditional MLE because the value of the observed sufficient statistic lies at an extreme of the derived distribution, implying that the conditional MLE does not exist. The confidence interval for the exact odds ratio does not include 1, so you can conclude that the odds of death increases significantly with dosage. Even though the asymptotic results are unreliable, the exact analysis allows you to conclude that there is a significant effect due to Dose.

This exact analysis should be compared to an asymptotic conditional likelihood analysis, which is available with the PHREG procedure. First, define a variable Z to be 1 if the response is an event and 2 if the response is a nonevent. This variable is used as the time variable as well as the censoring indicator (with 2 as the censored value) in the MODEL statement of PROC PHREG. Also specify the TIES=DISCRETE option to request the discrete logistic model, and the STRATA statement to specify the strata on which to condition:

```
proc phreg data=dose;
   strata Center;
   model Z*Z(2)=Dose / ties=discrete;
run;
```

For these data, the PHREG procedure does not converge and the maximum likelihood estimates are not valid. Generally, the conditional score statistics for testing the overall null hypothesis should be the same for both the asymptotic conditional analysis in PROC PHREG and the exact analysis in PROC LOGISTIC. However, PROC PHREG computes the p-value by comparing the value of the conditional score statistic to a chi-squared distribution, while PROC LOGISTIC derives its p-value from the exact conditional distribution. Also, inference on individual parameters is often not the same.

Example 7.2. Crossover Clinical Trial

One common use of conditional logistic regression is in a crossover clinical trial. In this example, the subjects are given a sequence of drugs and their response to each drug is recorded. Each subject is considered to be a separate stratum. The goal is to determine if the drugs have the same effect, adjusting for period and carryover effects. In this example, researchers give 15 different subjects three different drugs (A, B, P=placebo) in three consecutive periods (P1, P2, P3), and their response in each period is 1 for improvement and 0 for no improvement. The carryover effect is a classification variable indicating which drug was given in the preceding period:

```
data Crossover (drop=P1 P2 P3);
   input Subject P1$ P2$ P3$ Improve @@;
   Period=1; Drug=P1; Carry='0'; output;
   input Improve @@;
   Period=2; Drug=P2; Carry=P1;  output;
   input Improve @@;
   Period=3; Drug=P3; Carry=P2;  output;
   datalines;
1  A B P 0 0 0    8  B P A 0 0 1
2  A B P 1 1 0    9  B P A 1 0 1
3  A B P 0 1 1    10 B P A 0 1 0
4  A P B 1 0 1    11 P A B 0 1 0
5  A P B 1 0 0    12 P B A 1 0 1
6  B A P 0 0 0    13 P B A 0 0 1
7  B A P 1 1 0    14 P B A 0 1 0
                  15 P B A 0 1 1
;
```

The model to be fit is

$$\begin{aligned}\text{logit}(\pi_{hi}) &= \alpha_h + I(\text{Drug}_{hi} = \text{A})\beta_1 + I(\text{Drug}_{hi} = \text{B})\beta_2 \\ &\quad + I(i = 1)\beta_3 + I(i = 2)\beta_4\end{aligned}$$

where h indexes the subject, α_h are the subject intercepts, i indexes the period, and the $I(\cdot)$ are indicator variables taking the value 1 when the condition is true. Note that this model ignores carryover effects. The following statements perform the analysis:

```
proc logistic data=Crossover descending exactonly;
   class Subject Drug Period/ param=ref;
   model Improve=Subject Drug Period;
   exact Drug Period/ joint;
run;
```

The exact conditional score p-value for the test of significance of all the parameters is 0.1835; hence, you cannot reject the null hypothesis. However, the exact conditional score p-value for the test of no drug effects, $\beta_1 = \beta_2 = 0$, is 0.0583, while the p-value for the test of no period effects, $\beta_3 = \beta_4 = 0$, is 0.8605, which suggests that the period term should be dropped from this model.

References

Agresti, Alan (1990), *Categorical Data Analysis*, New York: John Wiley & Sons, Inc.

Agresti, Alan (1992), "A Survey of Exact Inference for Contingency Tables," *Statistical Science*, 7, 131–177.

Cox, D.R. (1970), *Analysis of Binary Data*, New York: Chapman and Hall.

Cox, D.R. and Snell, E.J. (1989), *Analysis of Binary Data*, Second Edition, New York: Chapman and Hall.

Derr, Robert E. (2000), "Performing Exact Logistic Regression with the SAS System," *Proceedings of the Twenty-Fifth Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc.

Hirji, Karim F., Mehta, Cyrus R., and Patel, Nitin R. (1987), "Computing Distributions for Exact Logistic Regression," *Journal of the American Statistical Association*, 82, 1110–1117.

Hirji, Karim F. and Tang, Man-Lai (1998), "A Comparison of Tests for Trend," *Communications in Statistics—Theory and Methods*, 27, 943–963.

Hirji, Karim F., Tsiatis, Anastasios A., and Mehta, Cyrus R. (1989), "Median Unbiased Estimation for Binary Data," *American Statistician*, 43, 7–11.

Lancaster, H. O., (1961), "Significance Tests in Discrete Distributions," *Journal of the American Statistical Association*, 56, 223–234.

Mehta, Cyrus R. and Patel, Nitin R. (1995), "Exact Logistic Regression: Theory and Examples," *Statistics in Medicine*, 14, 2143–2160.

Stokes, Maura E., Davis, Charles S., and Koch, Gary G. (1995), *Categorical Data Analysis Using the SAS System*, Cary, NC: SAS Institute Inc.

Vollset, Stein E., Hirji, Karim F., and Afifi, Abdelmonem A. (1991), "Evaluation of Exact and Asymptotic Interval Estimators in Logistic Analysis of Matched Case-Control Studies," *Biometrics*, 47, 1311–1325.

Chapter 8
The MIXED Procedure

Chapter Table of Contents

DETAILS 111
Default Output 111
REFERENCES 111

Chapter 8
The MIXED Procedure

Details

Default Output

Fit Statistics

AIC and BIC are now printed in smaller-is-better forms in the "Fit Statistics" table. A finite-sample corrected version of AIC (AICC) is also included. When you specify METHOD=ML, these criteria now incorporate the effective number of fixed-effects parameters (the rank of the X matrix) in addition to the number of estimated covariance parameters. Refer to Burnham and Anderson (1998) for additional details.

References

Burnham, K.P. and Anderson, D.R. (1998), *Model Selection and Inference: A Practical Information-Theoretic Approach*, New York: Springer-Verlag.

Chapter 9
The MODECLUS Procedure

Chapter Table of Contents

DETAILS 115
Density Estimation 115

Chapter 9
The MODECLUS Procedure

Details

Density Estimation

The MODECLUS procedure now provides a default smoothing parameter if none of the options (DR=, CR=, R=, DK=, CK=, and K=) is specified. The formula for computing this default value is given by

$$\left[\frac{2^{v+2}(v+2)\Gamma(.5v+1)}{nv^2}\right]^{1/(v+4)}\sqrt{\sum_{l=1}^{v}s_l^2}$$

If the data are distances, the factor $\sqrt{\sum {s_l}^2}$ can be replaced by an average root-mean-square Euclidean distance divided by $\sqrt{2}$.

Chapter 10
The MULTTEST Procedure

Chapter Table of Contents

SYNTAX 119
PROC MULTTEST Statement 119
STRATA Statement 119

DETAILS 119
p-Value Adjustments 119

REFERENCES 120

Chapter 10
The MULTTEST Procedure

Syntax

PROC MULTTEST Statement

Two new p-value adjustment methods are available: Fisher combination and Hommel. You can obtain these adjustments by specifying the FISHER_C and HOMMEL options, respectively.

STRATA Statement

The WEIGHT= option for the STRATA statement specifies the type of strata weighting to use when computing the Freeman-Tukey and t-tests for the mean. Valid values for the WEIGHT= option in the STRATA statement are SAMPLESIZE, HARMONIC, and EQUAL. SAMPLESIZE requests weights proportional to the within-stratum sample sizes, and is the default method. HARMONIC sets up weights equal to the harmonic mean of the non-missing within-stratum CLASS sizes, and is similar to a Type 2 analysis in PROC GLM. EQUAL specifies equal weights, and is similar to a Type 3 analysis in PROC GLM.

Details

p-Value Adjustments

The FISHER_C option requests adjusted p-values using closed tests, based on the idea of Fisher's combination test. The Fisher combination test for a joint test of any set of S hypotheses with p-values uses the Chi-Square statistic $\chi^2 = -2\sum \log(p_i)$, with $2S$ degrees of freedom. The FISHER_C adjusted p-value for test j is the maximum of all p-values for the combination tests, taken over all joint tests that include j as one of their components. Independence of p-values is absolutely required for this method.

Hommel's (1988) method is a closed testing procedure based on Simes' (1986) test. The Simes p-value for a joint test of any set of S hypotheses with p-values $p_1 \leq p2 \leq \ldots \leq p_S$ is $\min((S/1)p_1, (S/2)p_2, \ldots, (S/S)p_S)$. The Hommel adjusted p-value for test j is the maximum of all such Simes p-values, taken over all joint tests that include j as one of their components. Hochberg adjusted p-values are always as large or larger than Hommel adjusted p-values. Sarkar and Chang (1997) showed that Simes' method is valid under independent or positively dependent p-values, so Hommel's and Hochberg's methods also are valid in such cases by the closure principle.

Westfall et al. (1999) and Westfall and Wolfinger (2000) are new references dealing with multiplicity issues and PROC MULTTEST.

References

Hommel, G. (1988), "A Comparison of Two Modified Bonferroni Procedures," *Biometrika,* 75, 383–386.

Sarkar, S. and Chang, C.K. (1997), "Simes Method for Multiple Hypothesis Testing with Positively Dependent Test Statistics," *Journal of the American Statistical Association,* 92, 1601–1608.

Simes, R.J. (1986), "An Improved Bonferroni Procedure for Multiple Tests of Significance," *Biometrika,* 73, 751–754.

Westfall, P.H., Tobias, R.D., Rom, D., Wolfinger, R.D., and Hochberg, Y. (1999), *Multiple Comparisons and Multiple Tests using the SAS System,* SAS Institute Inc., Cary, NC.

Westfall, P.H. and Wolfinger, R.D. (2000), "Closed Multiple Testing Procedures and PROC MULTTEST," *Observations,* SAS Institute Inc., to appear.

Chapter 11
The NLMIXED Procedure

Chapter Table of Contents

SYNTAX 123
MODEL Statement 123
DETAILS 123
Displayed Output 123
REFERENCES 123

Chapter 11
The NLMIXED Procedure

Syntax

MODEL Statement

The gamma and negative binomial distributions are now available in the MODEL statement. They are specified as *gamma(a,b)* and *negbin(n,p)*.

Details

Displayed Output

Fitting Information

Only the smaller-is-better forms of AIC and BIC are now printed in the "Fit Statistics" table. A finite-sample corrected version of AIC (AICC) is also included. The criteria are computed as follows:

$$\begin{aligned} AIC &= 2f(\hat{\theta}) + 2p \\ AICC &= 2f(\hat{\theta}) + 2pn/(n-p-1) \\ BIC &= 2f(\hat{\theta}) + p\log(s) \end{aligned}$$

where $f()$ is the marginal likelihood function, $\hat{\theta}$ is the vector of parameter estimates, p is the number of parameters, n is the number of observations, and s is the number of subjects. Refer to Burnham and Anderson (1998) for additional details.

References

Burnham, K.P. and Anderson, D.R. (1998), *Model Selection and Inference: A Practical Information-Theoretic Approach*, New York: Springer-Verlag.

Chapter 12
The PHREG Procedure

Chapter Table of Contents

OVERVIEW 127

SYNTAX 127
PROC PHREG Statement 127

DETAILS 127
Robust Estimate of the Covariance Matrix 127
Modified Score Statistic 128
Fitting a Null Model 128

REFERENCES 128

Chapter 12
The PHREG Procedure

Overview

The PHREG procedure now offers the Wei and Lin (1989) robust estimate of the covariance matrix. Also in this release, you can fit a null model by not specifying any explanatory variables in the MODEL statement.

Syntax

PROC PHREG Statement

COVSANDWICH < (AGGREGATE) >
COVS < (AGGREGATE) >

requests the robust sandwich estimate of Lin and Wei (1989) for the covariance matrix. When this option is specified, this robust sandwich estimate is used in the Wald tests for testing the global null hypothesis, null hypotheses of individual parameters, and the hypotheses in the TEST statements. In addition, a modified score test is computed in the testing of the global null hypothesis, and the parameter estimates table has an additional StdErrRatio column, which contains the ratios of the robust estimate of the standard error relative to the corresponding model-based estimate. Optionally, you can specify the keyword AGGREGATE enclosed in parentheses after the COVSANDWICH (or COVS) option, which requests a summing up of the score residuals for each distinct ID pattern in the computation of the robust sandwich covariance estimate. This AGGREGATE option has no effects if the ID statement is not specified.

Details

Robust Estimate of the Covariance Matrix

The robust variance of Lin and Wei (1989) is a sandwich estimate given by

$$\hat{\boldsymbol{V}}^{s} = \boldsymbol{I}^{-1}(\boldsymbol{U}'\boldsymbol{U})\boldsymbol{I}^{-1}$$

where $\boldsymbol{I}$ is the information matrix evaluated at the maximum likelihood estimate $\hat{\beta}$ and $\boldsymbol{U}$ is the $n \times p$ matrix of score residuals. Since the matrix of DFBETA residuals $\boldsymbol{D}$ can be written as $\boldsymbol{D} = \boldsymbol{U}\boldsymbol{I}^{-1}$, the robust sandwich variance estimate $\hat{\boldsymbol{V}}^{s}$ can be computed as

$$\hat{\boldsymbol{V}}^{s} = \boldsymbol{D}'\boldsymbol{D}$$

Modified Score Statistic

Let $\boldsymbol{U}_0$ be the $n \times p$ matrix of efficient scores and $\boldsymbol{I}_0$ be the $p \times p$ information matrix, both evaluated at $\beta = \mathbf{0}$; let $\mathbf{1}$ be a column n-vector of 1's. Let $\hat{\beta}_1$ be the one-step estimate of β; that is,

$$\hat{\beta}_1 = \boldsymbol{I}_0^{-1}(\boldsymbol{U}_0'\mathbf{1})$$

the covariance matrix is estimated by $\hat{\boldsymbol{V}}(\beta_1) = \boldsymbol{I}_0^{-1}$.

The score statistic for testing H_0:β=$\mathbf{0}$ can be expressed as a Wald test statistic (Therneau and Grambsch [2000]):

$$\begin{aligned}(\boldsymbol{U}_0'\mathbf{1})'\boldsymbol{I}_0^{-1}(\boldsymbol{U}_0'\mathbf{1}) &= [\boldsymbol{I}_0^{-1}(\boldsymbol{U}_0'\mathbf{1})]'\boldsymbol{I}_0[\boldsymbol{I}_0^{-1}(\boldsymbol{U}_0'\mathbf{1})] \\ &= \hat{\beta}_1'[\hat{\boldsymbol{V}}(\beta_1)]^{-1}\hat{\beta}_1\end{aligned}$$

The modified score test statistic for testing H_0:β=$\mathbf{0}$ is obtained by replacing $\hat{\boldsymbol{V}}(\beta_1)$ in the score statistic by the robust sandwich estimate $\hat{\boldsymbol{V}}_0^s = \boldsymbol{D}_0'\boldsymbol{D}_0$ where $\boldsymbol{D}_0 = \boldsymbol{U}_0\boldsymbol{I}_0^{-1}$:

$$\begin{aligned}\hat{\beta}_1(\hat{\boldsymbol{V}}_0^s)^{-1}\hat{\beta}_1 &= [\boldsymbol{I}_0^{-1}(\boldsymbol{U}_0'\mathbf{1})]'(\boldsymbol{D}_0'\boldsymbol{D}_0)^{-1}[\boldsymbol{I}_0^{-1}(\boldsymbol{U}_0'\mathbf{1})] \\ &= (\boldsymbol{U}_0'\mathbf{1})'(\boldsymbol{U}_0'\boldsymbol{U}_0)^{-1}(\boldsymbol{U}_0'\mathbf{1})\end{aligned}$$

Fitting a Null Model

In some situations you may want to fit a null model. For instance, by fitting a null model, you can trick PROC PHREG into providing the Kaplan-Meier estimate of the survivor function for a set of survival times with right censoring and left truncation. In the following SAS program, t2 is the survival time, t1 is the left truncation time, and c is the censoring indicator with value 1 indicating censored observations. By not specifying any explanatory variables in the MDOEL statement, you are fitting a null model. The sdf variable in the output data set out1 contains the product-limit estimates.

```
proc phreg;
  model t2*c(1)= /entrytime=t1 ;
  output out=out1 survival= sdf;
  run;
```

References

Lin, D.Y. and Wei, L.J. (1989), "The Robust Inference for the Proportional Hazards Model," *Journal of the American Statistical Association*, 84, 1074–1078.

Therneau, T. M. and Grambsch, P.M. (2000), *Modeling Survival Data: Extending the Cox Model.* New York: Springer-Verlag.

Chapter 13
The SURVEYMEANS Procedure

Chapter Table of Contents

OVERVIEW 131

GETTING STARTED 131
Stratified Sampling 131

SYNTAX 133
DOMAIN Statement 133

DETAILS 134
Domain Analysis 134
Missing Values 134
Statistical Computations 136
Displayed Output 139
ODS Table Names 139

EXAMPLES 139
Example 13.1 Stratified Cluster Sample Design 140
Example 13.2 Domain Analysis 143
Example 13.3 Analyze Survey Data with Missing Values 146

REFERENCES 149

Chapter 13
The SURVEYMEANS Procedure

Overview

The new DOMAIN statement in the SURVEYMEANS procedure enables you to perform domain analysis for survey data.

Examples and computational details are revised and improved.

Getting Started

Stratified Sampling

This example illustrates how you can use PROC SURVEYMEANS to estimate population means and proportions from sample survey data. The study population is a junior high school with a total of 4,000 students in grades 7, 8, and 9. Researchers want to know how much these students spend weekly for ice cream, on average, and what percentage of students spend at least $10 weekly for ice cream.

Suppose that the sample of students was selected using stratified random sampling. In stratified sampling, the study population is divided into nonoverlapping strata, and samples are selected from each stratum independently.

The list of students in this junior high school was stratified by grade, yielding three strata: grades 7, 8, and 9. A simple random sample of students was selected from each grade. Table 13.1 shows the total number of students in each grade.

Table 13.1. Number of Students by Grade

Grade	Number of Students
7	1,824
8	1,025
9	1,151
Total	4,000

A sample of 40 students was selected from the entire student population. Each student selected for the sample was asked how much he or she spends for ice cream per week, on average. The SAS data set named IceCream saved the responses of the 40 students:

```
data IceCream;
   input Grade Spending @@;
   if (Spending < 10) then Group='less';
     else Group='more';
   datalines;
```

```
7 7  7  7  8 12  9 10  7  1  7 10  7  3  8 20  8 19  7 2
7 2  9 15  8 16  7  6  7  6  7  6  9 15  8 17  8 14  9 8
9 8  9  7  7  3  7 12  7  4  9 14  8 18  9  9  7  2  7 1
7 4  7 11  9  8  8 10  8 13  7  2  9  6  9 11  7  2  7 9
;
```

The variable Grade contains a student's grade. The variable Spending contains a student's response on how much was spent per week for ice cream, in dollars. The variable Group is created to indicate whether a student spends at least $10 weekly for ice cream: Group='more' if a student spends at least $10, or Group='less' if a student spends less than $10.

To analyze this stratified sample, you need to provide the population totals for each stratum to PROC SURVEYMEANS. The SAS data set named StudentTotals contains the information from Table 13.1:

```
data StudentTotals;
   input Grade _total_; datalines;
7 1824
8 1025
9 1151
;
```

The variable Grade is the stratum identification variable, and the variable _TOTAL_ contains the total number of students for each stratum. PROC SURVEYMEANS requires you to use the variable name _TOTAL_ for the stratum population totals.

The procedure uses the stratum population totals to adjust variance estimates for the effects of sampling from a finite population. If you do not provide population totals or sampling rates, then the procedure assumes that the proportion of the population in the sample is very small, and the computation does not involve a finite population correction.

In a stratified sample design, when the sampling rates in the strata are unequal, you need to use sampling weights to reflect this information in order to produce an unbiased mean estimator. In this example, the appropriate sampling weights are reciprocals of the probabilities of selection. You can use the following data step to create the sampling weights:

```
data IceCream;
   set IceCream;
   if Grade=7 then Prob=20/1824;
   if Grade=8 then Prob=9/1025;
   if Grade=9 then Prob=11/1151;
   Weight=1/Prob;
```

If you use PROC SURVEYSELECT to select your sample, it creates these sampling weights for you.

The following SAS statements perform the stratified analysis of the survey data:

```
title1 'Analysis of Ice Cream Spending';
title2 'Stratified Simple Random Sample Design';
proc surveymeans data=IceCream total=StudentTotals;
   stratum Grade / list;
   var Spending Group;
   weight Weight;
run;
```

The PROC SURVEYMEANS statement invokes the procedure. The DATA= option names the SAS data set IceCream as the input data set to be analyzed. The TOTAL= option names the data set StudentTotals as the input data set containing the stratum population totals. Notice that the TOTAL=StudentTotals option is used here instead of the TOTAL=4000 option. In this stratified sample design, the population totals are different for different strata, and so you need to provide them to PROC SURVEYMEANS in a SAS data set.

The STRATA statement identifies the stratification variable Grade. The LIST option in the STRATA statement requests that the procedure display stratum information. The WEIGHT statement tells the procedure that the variable Weight contains the sampling weights.

Syntax

The following statement is available in PROC SURVEYMEANS.

DOMAIN *variables* < *variable***variable*
*variable***variable***variable* . . . > **;**

DOMAIN Statement

DOMAIN | **SUBGROUP** *variables* < *variable***variable*
*variable***variable***variable* . . . > **;**

The DOMAIN statement requests analysis for subpopulations, or domains, in addition to analysis for the entire study population. The DOMAIN statement names the variables that identify domains, which are called domain variables.

It is common practice to compute statistics for domains. The formation of these domains may be unrelated to the sample design. Therefore, the sample sizes for the domains are random variables. In order to incorporate this variability into the variance estimation, you should use a DOMAIN statement.

Note that a DOMAIN statement is different from a BY statement. In a BY statement, you treat the sample sizes as fixed in each subpopulation, and you perform analysis within each BY group independently. See the section "Domain Analysis" on page 136 for more details.

A domain variable can be either character or numeric. However, the procedure treats domain variables as categorical variables. If a variable appears by itself in a

DOMAIN statement, each level of this variable determines a domain in the study population. If two or more variables are joined by asterisks (*), then every possible combination of levels of the variables determines a domain. The procedure performs a descriptive analysis within each domain defined by the domain variables.

The formatted values of the domain variables determine the categorical variable levels. Thus, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *SAS Procedures Guide* and to the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*.

Details

Domain Analysis

It is common practice to compute statistics for subpopulations, or domains, in addition to computing statistics for the entire study population. Analysis for domains using the entire sample is called *domain analysis* (or subgroup analysis, subpopulation analysis, subdomain analysis). The formation of these subpopulations of interest may be unrelated to the sample design. Therefore, the sample sizes for the subpopulations may actually be random variables.

In order to incorporate this variability into the variance estimation, you should use a DOMAIN statement. Note that using a BY statement provides completely separate analyses of the BY groups. It does not provide a statistically valid subpopulation or domain analysis, where the total number of units in the subpopulation is not known with certainty. For more detailed information about domain analysis, refer to Kish (1965) and Statistical Laboratory (1989).

Missing Values

When computing statistics for an analysis variable, PROC SURVEYMEANS omits observations with missing values for that variable. The procedure bases statistics for each variable only on observations that have nonmissing values for that variable. If you specify the MISSING option in the PROC SURVEYMEANS statement, the procedure treats missing values of a categorical variable as a valid category.

An observation is also excluded if it has a missing value for any STRATA or CLUSTER variable, unless the MISSING option is used.

If an observation has a missing value or a nonpositive value for the WEIGHT variable, then PROC SURVEYMEANS excludes that observation from the analysis.

The procedure performs univariate analysis and analyzes each VAR variable separately. Thus, the number of missing observations may be different for different variables. You can specify the keyword NMISS in the PROC SURVEYMEANS statement to display the number of missing values for each analysis variable in the "Statistics" table.

If you have missing values in your survey data for any reason (such as nonresponse), this can compromise the quality of your survey results. An observation without

missing values is called a complete respondent, and an observation with missing values is called an incomplete respondent. If the complete respondents are different from the incomplete respondents with regard to a survey effect or outcome, then survey estimates will be biased and will not accurately represent the survey population. There are a variety of techniques in sample design and survey operations that can reduce nonresponse. Once data collection is complete, you can use imputation to replace missing values with acceptable values, and you can use sampling weight adjustments to compensate for nonresponse. You should complete this data preparation and adjustment before you analyze your data with PROC SURVEYMEANS. Refer to Cochran (1977), Kalton and Kaspyzyk (1986), and Brick and Kalton (1996) for more details.

If there is evidence indicating that complete respondents are different from incomplete respondents for your study, you can use the DOMAIN statement to compute the descriptive statistics "among complete respondents" from your survey data without imputation on incomplete respondents. See Example 13.3 on page 146.

If missing values result in empty strata in the sample, then they will have an impact on the statistical computation, which uses the total number of strata. If all the observations in a stratum have missing weights or missing values for the current analysis variable, this stratum is an *empty stratum*. For example,

```
data new;
   input stratum y z w;
   datalines;
1 . 13 40
1 2  9  .
1 .  5 25
2 5 10 20
2 8 60 15
;
proc surveymeans df mean nobs nmiss;
   strata stratum;
   var y z;
   weight w;
run;
```

You analyze variable Y and Z, with weight variable W and stratum variable STRATUM. For variable Y, all observations have missing values or missing weights in STRATUM=1, therefore, the analysis for variable Y uses only observations in STRATUM=2. Thus, for variable Y, STRATUM=1 is an empty stratum and STRATUM=2 is a non-empty stratum. Note, however, that STRATUM=1 is a non-empty stratum for variable Z.

If your sample design contains stratification, PROC SURVEYMEANS analyzes only the data in non-empty strata. Therefore, the total number of strata for an analysis variable means the total number of *non-empty* strata. In this example, the total number of strata for Y and Z is one and two, respectively.

Statistical Computations

t Test for the Mean

If you specify the keyword T, PROC SURVEYMEANS computes the *t* value for testing that the population mean equals zero, $H_0 : \bar{Y} = 0$. The test statistic equals

$$t(\widehat{\bar{Y}}) \quad = \quad \widehat{\bar{Y}} \;/\; \text{StdErr}(\widehat{\bar{Y}})$$

The two-sided *p*-value for this test is

$$\text{Prob}(\; |T| > |t(\widehat{\bar{Y}})| \;)$$

where T is a random variable with the *t* distribution with *df* degrees of freedom.

PROC SURVEYMEANS calculates the degrees of freedom for the *t* test as the number of clusters minus the number of strata. If there are no clusters, then *df* equals the number of observations minus the number of strata. If the design is not stratified, then *df* equals the number of clusters minus one. The procedure displays *df* for the *t* test if you specify the keyword DF in the PROC SURVEYMEANS statement.

If missing values or missing weights are present in your data, the number of strata, the number of observations, and the number of clusters are counted based on the observations in non-empty strata. See the section "Missing Values" on page 134 for details. For degrees of freedom in domain analysis, see the section "Domain Analysis" on page 136.

Domain Analysis

When you use a DOMAIN statement to request a domain analysis, the procedure computes the requested statistics for each domain.

For a domain D, let I_D be the corresponding indicator variable:

$$I_D(h,i,j) \quad = \begin{cases} 1 & \text{if observation } (h,i,j) \text{ belongs to } D \\ 0 & \text{otherwise} \end{cases}$$

Let

$$z_{hij} \quad = \quad y_{hij} I_D(h,i,j) \quad = \begin{cases} y_{hij} & \text{if observation } (h,i,j) \text{ belongs to } D \\ 0 & \text{otherwise} \end{cases}$$

The requested statistics for variable y in domain D are computed based on the values of z.

Domain Mean The estimated mean of y in the domain D is

$$\widehat{\bar{Y}_D} \quad = \quad \left(\sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} v_{hij}\, z_{hij} \right) / \; v_{...}$$

where

$$v_{hij} \quad = \quad w_{hij} I_D(h,i,j) = \begin{cases} w_{hij} & \text{if observation } (h,i,j) \text{ belongs to } D \\ 0 & \text{otherwise} \end{cases}$$

$$v_{...} = \sum_{h=1}^{H}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} v_{hij}$$

The variance of $\widehat{\bar{Y}_D}$ is estimated by

$$\widehat{V}(\widehat{\bar{Y}_D}) = \sum_{h=1}^{H} \frac{n_h(1-f_h)}{n_h-1} \sum_{i=1}^{n_h} (r_{hi\cdot} - \bar{r}_{h\cdot\cdot})^2$$

where

$$r_{hi\cdot} = \left(\sum_{j=1}^{m_{hi}} v_{hij}\ (z_{hij} - \widehat{\bar{Y}_D}) \right) / \ v_{...}$$

$$\bar{r}_{h\cdot\cdot} = \left(\sum_{i=1}^{n_h} r_{hi\cdot} \right) / \ n_h$$

Domain Total The estimated total in domain D is

$$\widehat{Y}_D = \sum_{h=1}^{H}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} v_{hij}\ z_{hij}$$

and its estimated variance is

$$\widehat{V}(\widehat{Y}_D) = \sum_{h=1}^{H} \frac{n_h(1-f_h)}{n_h-1} \sum_{i=1}^{n_h} (z_{hi\cdot} - \bar{z}_{h\cdot\cdot})^2$$

where

$$z_{hi\cdot} = \sum_{j=1}^{m_{hi}} v_{hij}\ z_{hij}$$

$$\bar{z}_{h\cdot\cdot} = \left(\sum_{i=1}^{n_h} z_{hi\cdot} \right) \ / \ n_h$$

Degrees of Freedom For domain analysis, PROC SURVEYMEANS computes the degrees of freedom for *t* tests as the number of clusters in the non-empty strata minus the number of non-empty strata. When the sample design has no clusters, the degrees of freedom equals the number of observations in non-empty strata minus the number of non-empty strata. As discussed in the section "Missing Values" on page 134, missing values and missing weights can result in empty strata. In domain analysis, an empty stratum can also occur when the stratum contains no observations in the specified domain. If no observations in a whole stratum belong to a domain, then this stratum is called an empty stratum for that domain.

For example,

```
data new;
   input str clu y w d;
   datalines;
1 1 . 40 9
1 2 2  . 9
1 3 . 25 9
2 4 5 20 9
2 5 8 15 9
3 6 5 30 7
3 7 9 89 7
3 8 6 23 7
;
proc surveymeans df nobs nclu nmiss;
   strata str;
   cluster clu;
   var y;
   weight w;
   domain d;
run;
```

Table 13.2. Calculations of df for Y

	Domain D=**7**	**Domain** D=**9**
Non Empty Strata	STR=3	STR=2
Clusters Used in the Analysis	CLU=6, CLU=7, and CLU=8	CLU=4 and CLU=5
df	$3-1=2$	$2-1=1$

Although there are three strata in the data set, STR=1 is an empty stratum for variable Y because of missing values and missing weights. In addition, no observations in stratum STR=3 belong to domain D=9. Therefore, STR=3 becomes an empty stratum as well for variable Y in domain D=9. As a result, the total number of non-empty strata for domain D=9 is one. The non-empty stratum for domain D=9 and variable Y is stratum STR=2. The total number of clusters for domain D=9 is two, which belong to stratum STR=2. Thus, for variable Y in domain D=9, the degrees of freedom for the *t* tests of the domain mean is $df = 2-1 = 1$. Similarly, for domain D=7, strata STR=1 and STR=2 are both empty strata, so the total number of strata is one (STR=3), and the total number of clusters is three (CLU=6, CLU=7, and CLU=8). Table 13.2 illustrates how domains affect the total number of clusters and total number of strata in the df calculation. Figure 13.1 shows the df computed by the procedure.

The SURVEYMEANS Procedure

Domain Analysis: d

d	Variable	N	N Miss	Clusters	DF
7	y	3	0	3	2
9	y	2	2	2	1

Figure 13.1. Degrees of Freedoms in Domain Analysis

Displayed Output

Domain Analysis

If you use a DOMAIN statement, the procedure displays statistics in each domain in a "Domain Analysis" table. A "Domain Analysis" table contains all the columns in the "Statistics" table, plus columns of domain variable values.

Note that depending on how you define the domains with domain variables, the procedure may produce more than one "Domain Analysis" table. For example, in the following DOMAIN statement

```
domain A B*C*D A*C C;
```

you use four definitions to define domains:

- A: all the levels of A
- C: all the levels of C
- A*C: all the interactive levels of A and C
- B*C*D: all the interactive levels of B, C, and D

The procedure displays four "Domain Analysis" tables, one for each domain definition.

ODS Table Names

PROC SURVEYMEANS assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table.

Table 13.3. ODS Tables Produced in PROC SURVEYMEANS

ODS Table Name	Description	Statement	Option
ClassVarInfo	Class level information	CLASS	default
Domain	Statistics in domains	DOMAIN	default
Statistics	Statistics	PROC	default
StrataInfo	Stratum information	STRATA	LIST
Summary	Data summary	PROC	default

Examples

The "Getting Started" section on page 131 contains an example of analyzing data from stratified simple random sample designs. This section provides more examples that illustrate how to use PROC SURVEYMEANS.

Example 13.1. Stratified Cluster Sample Design

Consider the example in the section "Stratified Sampling" on page 131. The study population is a junior high school with a total of 4,000 students in grades 7, 8, and 9. Researchers want to know how much these students spend weekly for ice cream, on the average, and what percentage of students spend at least $10 weekly for ice cream.

The example in the section "Stratified Sampling" on page 131 assumes that the sample of students was selected using a stratified simple random sample design. This example shows analysis based on a more complex sample design.

Suppose that every student belongs to a study group and that study groups are formed within each grade level. Each study group contains between two and four students. Table 13.4 shows the total number of study groups for each grade.

Table 13.4. Study Groups and Students by Grade

Grade	Number of Study Groups	Number of Students
7	608	1,824
8	252	1,025
9	403	1,151
Total	617	4,000

It is quicker and more convenient to collect data from students in the same study group than to collect data from students individually. Therefore, this study uses a stratified clustered sample design. The primary sampling units, or clusters, are study groups. The list of all study groups in the school is stratified by grade level. From each grade level, a sample of study groups is randomly selected, and all students in each selected study group are interviewed. The sample consists of eight study groups from the 7th grade, three groups from the 8th grade, and five groups from the 9th grade.

The SAS data set named IceCreamStudy saves the responses of the selected students:

```
data IceCreamStudy;
   input Grade StudyGroup Spending @@;
   if (Spending < 10) then Group='less';
     else Group='more';
   datalines;
7  34  7     7  34  7    7 412  4     9  27 14
7  34  2     9 230 15    9  27 15     7 501  2
9 230  8     9 230  7    7 501  3     8  59 20
7 403  4     7 403 11    8  59 13     8  59 17
8 143 12     8 143 16    8  59 18     9 235  9
8 143 10     9 312  8    9 235  6     9 235 11
9 312 10     7 321  6    8 156 19     8 156 14
7 321  3     7 321 12    7 489  2     7 489  9
7  78  1     7  78 10    7 489  2     7 156  1
7  78  6     7 412  6    7 156  2     9 301  8
;
```

In the data set IceCreamStudy, the variable Grade contain a student's grade. The variable StudyGroup identifies a student's study group. It is possible for students from different grades to have the same study group number because study groups are sequentially numbered within each grade. The variable Spending contains a student's response to how much he or she spends per week for ice cream, in dollars. The variable GROUP indicates whether a student spends at least $10 weekly for ice cream. It is not necessary to store the data in order of grade and study group.

The SAS data set StudyGroup is created to provide PROC SURVEYMEANS with the sample design information shown in Table 13.4:

```
data StudyGroups;
   input Grade _total_; datalines;
7 608
8 252
9 403
;
```

The variable Grade identifies the strata, and the variable _TOTAL_ contains the total number of study groups in each stratum. The population totals stored in the variable _TOTAL_ should be expressed in terms of the primary sampling units (PSUs), which are study groups in this example. Therefore, the variable _TOTAL_ contains the total number of study groups for each grade, rather than the total number of students.

In order to obtain unbiased estimates, you create sampling weights using the following SAS statements:

```
data IceCreamStudy;
   set IceCreamStudy;
   if Grade=7 then Prob=8/608;
   if Grade=8 then Prob=3/252;
   if Grade=9 then Prob=5/403;
   Weight=1/Prob;
```

The sampling weights are the reciprocals of the probabilities of selections. The variable Weight contains the sampling weights. Because the sampling design is clustered, and all students from each selected cluster are interviewed, the sampling weights equal the inverse of the cluster (or study group) selection probabilities.

The following SAS statements perform the analysis for this sample design:

```
title1 'Analysis of Ice Cream Spending';
title2 'Stratified Clustered Sample Design';
proc surveymeans data=IceCreamStudy total=StudyGroups;
   stratum Grade / list;
   cluster StudyGroup;
   var Spending Group;
   weight Weight;
run;
```

Output 13.1.1. Data Summary and Class Information

```
                    Analysis of Ice Cream Spending
                  Stratified Clustered Sample Design

                      The SURVEYMEANS Procedure

                             Data Summary

                  Number of Strata                   3
                  Number of Clusters                16
                  Number of Observations            40
                  Sum of Weights                 3162.6


                       Class Level Information

                  Class
                  Variable      Levels    Values

                  Group              2    less more
```

Output 13.1.1 provides information on the sample design and the input data set. There are 3 strata in the sample design, and the sample contains 16 clusters and 40 observations. The variable Group has two levels, 'less' and 'more'.

Output 13.1.2. Stratum Information

```
                    Analysis of Ice Cream Spending
                  Stratified Clustered Sample Design

                      The SURVEYMEANS Procedure

                         Stratum Information

Stratum            Population  Sampling
 Index      Grade       Total      Rate     N Obs  Variable           N  Clusters
-------------------------------------------------------------------------------
   1            7         608    1.32%         20  Spending          20         8
                                                   Group=less        17         8
                                                   Group=more         3         3
   2            8         252    1.19%          9  Spending           9         3
                                                   Group=less         0         0
                                                   Group=more         9         3
   3            9         403    1.24%         11  Spending          11         5
                                                   Group=less         6         4
                                                   Group=more         5         4
-------------------------------------------------------------------------------
```

Output 13.1.2 displays information for each stratum. Since the primary sampling units in this design are study groups, the population totals shown in Output 13.1.2 are the total numbers of study groups for each stratum or grade. Output 13.1.2 also displays the number of clusters for each stratum and analysis variable.

Output 13.1.3. Statistics

Analysis of Ice Cream Spending
Stratified Clustered Sample Design

The SURVEYMEANS Procedure

Statistics

Variable	N	Mean	Std Error of Mean	Lower 95% CL for Mean	Upper 95% CL for Mean
Spending	40	8.923860	0.650859	7.517764	10.329957
Group=less	23	0.561437	0.056368	0.439661	0.683213
Group=more	17	0.438563	0.056368	0.316787	0.560339

Output 13.1.3 displays the estimates of the average weekly ice cream expense and the percentage of students spending at least $10 weekly for ice cream.

Example 13.2. Domain Analysis

Suppose that you are studying profiles of the 800 top-performing companies to provide information on their impact on the economy. You are also interested in the company profiles within each market type. A sample of 66 companies is selected with unequal probability across market types. However, market type is not included in the sample design. Thus, the number of companies within each market type is a random variable in your sample. To obtain statistics within each market type, you should use domain analysis. The data of the 66 companies are saved in the following data set:

```
data Company;
   length Type $14;
   input Type$ Asset Sale Value Profit Employee Weight;
   datalines;
Other            2764.0  1828.0  1850.3   144.0   18.7   9.6
Energy          13246.2  4633.5  4387.7   462.9   24.3  42.6
Finance          3597.7   377.8    93.0    14.0    1.1  12.2
Transportation   6646.1  6414.2  2377.5   348.2   47.1  21.8
HiTech           1068.4  1689.8  1430.2    72.9    4.6   4.3
Manufacturing    1125.0  1719.4  1057.5    98.1   20.4   4.5
Other            1459.0  1241.4   452.7    24.5   20.1   5.5
Finance          2672.3   262.5   296.2    23.1    2.2   9.3
Finance           311.0   566.2   932.0    52.8    2.7   1.9
Energy           1148.6  1014.6   485.1    60.6    4.0   4.5
Finance          5327.0   572.4   372.9    25.2    4.2  17.7
Energy           1602.7   678.4   653.0    75.6    2.8   6.0
Energy           5808.8  1288.4  2007.0   318.8    5.9  19.2
Medical           268.8   204.4   820.9    45.6    3.7   1.8
Transportation   5222.6  2627.8  1910.0   245.6   22.8  17.4
Other             872.7  1419.4   939.3    69.7   12.2   3.7
Retail           4461.7  8946.8  4662.7   289.0  132.1  15.0
HiTech           6719.2  6942.0  8240.2   381.3   85.8  22.1
Retail            833.4  1538.8  1090.3    64.9   15.4   3.5
Finance           415.9   167.3  1126.8    56.8    0.7   2.2
```

```
HiTech            442.4  1139.9  1039.9    57.6   22.7   2.3
Other             801.5  1157.0   664.2    56.9   15.5   3.4
Finance          4954.8   468.8   366.4    41.7    3.0  16.5
Finance          2661.9   257.9   181.1    21.2    2.1   9.3
Finance          5345.8   530.1   337.4    36.4    4.3  17.8
Energy           3334.3  1644.7  1407.8   157.6    6.4  11.4
Manufacturing    1826.6  2671.7   483.2    71.3   25.3   6.7
Retail            618.8  2354.7   767.7    58.6   19.0   2.9
Retail           1529.1  6534.0   826.3    58.3   65.8   5.7
Manufacturing    4458.4  4824.5  3132.1    28.9   67.0  15.0
HiTech           5831.7  6611.1  9464.7   459.6   86.7  19.3
Medical          6468.3  4199.2  3170.4   270.1   59.5  21.3
Energy           1720.7   473.1   811.1    86.6    1.6   6.3
Energy           1679.7  1379.9   721.1    91.8    4.5   6.2
Retail           4018.2 16823.4  2038.3   178.1  162.0  13.6
Other             227.1   575.8  1083.8    62.6    1.9   1.6
Finance          3872.8   362.0   209.3    27.6    2.4  13.1
Retail           3359.3  4844.7  2651.4   224.1   75.6  11.5
Energy           1295.6   356.9   180.8   162.3    0.6   5.0
Energy           1658.0   626.6   688.0   126.0    3.5   6.1
Finance         12156.7  1345.5   680.7   106.6    9.4  39.2
HiTech           3982.6  4196.0  3946.8   313.9   64.3  13.5
Finance          8760.7   886.4  1006.9    90.0    7.5  28.5
Manufacturing    2362.2  3153.3  1080.0   137.0   25.2   8.4
Transportation   2499.9  3419.0   992.6    47.2   25.3   8.8
Energy           1430.4  1610.0   664.3    77.7    3.5   5.4
Energy          13666.5 15465.4  2736.7   411.4   26.6  43.9
Manufacturing    4069.3  4174.7  2907.6   289.2   38.2  13.7
Energy           2924.7   711.9  1067.8   146.7    3.4  10.1
Transportation   1262.1  1716.0   364.3    71.2   14.5   4.9
Medical           684.4   672.9   287.4    61.8    6.0   3.1
Energy           3069.3  1719.0  1439.0   196.4    4.9  10.6
Medical           246.5   318.8   924.1    43.8    3.1   1.7
Finance         11562.2  1128.5   580.4    64.2    6.7  37.3
Finance          9316.0  1059.4   816.5    95.9    8.0  30.2
Retail           1094.3  3848.0   563.3    29.4   44.7   4.4
Retail           1102.1  4878.3   932.4    65.2   47.3   4.4
HiTech            466.4   675.8   845.7    64.5    5.2   2.4
Manufacturing   10839.4  5468.7  1895.4   232.8   47.8  35.0
Manufacturing     733.5  2135.3    96.6    10.9    2.7   3.2
Manufacturing   10354.2 14477.4  5607.2   321.9  188.5  33.5
Energy           1902.1  2697.9   329.3    34.2    2.2   6.9
Other            2245.2  2132.2  2230.4   198.9    8.0   8.0
Transportation    949.4  1248.3   298.9    35.4   10.4   3.9
Retail           2834.4  2884.6   458.2    41.2   49.8   9.8
Retail           2621.1  6173.8  1992.7   183.7  115.1   9.2
;
```

For each company in your sample,

- The variable Type identifies the type of market for the company.
- The variable Asset contains the company's assets in millions of dollars.

- The variable Sale contains sales in millions of dollars.
- The variable Value contains the market value of the company in millions of dollars.
- The variable Profit contains the profit in millions of dollars.
- The variable Employee stores the number of employees in thousands.
- The variable Weight contains the sampling weight.

The following SAS statements use PROC SURVEYMEANS to perform the domain analysis, estimating means and other statistics for the overall population and also for the subpopulations (or domain) defined by market type. The DOMAIN statement specifies Type as the domain variable:

```
title1 'Top Companies Profile Study';
proc surveymeans data=Company total=800 mean sum;
   var Asset Sale Value Profit Employee;
   weight Weight;
   domain Type;
run;
```

Output 13.2.1. Company Profile Study

```
                    Top Companies Profile Study

                     The SURVEYMEANS Procedure

                            Data Summary

                  Number of Observations            66
                  Sum of Weights                 799.8


                              Statistics

                              Std Error
Variable            Mean         of Mean             Sum         Std Dev
------------------------------------------------------------------------
Asset         6523.488510      720.557075         5217486         1073829
Sale          4215.995799      839.132506         3371953          847885
Value         2145.935121      342.531720         1716319          359609
Profit         188.788210       25.057876          150993           30144
Employee        36.874869        7.787857           29493     7148.003298
------------------------------------------------------------------------
```

Output 13.2.1 shows that there are 66 observations in the sample. The sum of the sampling weights equals 799.8, which is close to the total number of companies in the study population.

The "Statistics" table in Output 13.2.1 displays the estimates of the mean and total for all analysis variables for the entire 800 companies, while Output 13.2.2 shows the mean and total estimates for each company type.

Output 13.2.2. Domain Analysis for Company Profile Study

Top Companies Profile Study

The SURVEYMEANS Procedure

Domain Analysis: Type

Type	Variable	Mean	Std Error of Mean	Sum	Std Dev
Energy	Asset	7868.302932	1941.699163	1449341	785962
	Sale	5419.679099	2416.214417	998305	673373
	Value	2249.297177	520.295162	414321	213580
	Profit	289.564658	52.512141	53338	25927
	Employee	14.151194	3.974697	2606.650000	1481.777769
Finance	Asset	7890.190264	1057.185336	1855773	704506
	Sale	829.210502	115.762531	195030	74436
	Value	565.068197	76.964547	132904	48156
	Profit	63.716837	10.099341	14986	5801.108513
	Employee	5.806293	0.811555	1365.640000	519.658410
HiTech	Asset	5031.959781	732.436967	321542	183302
	Sale	5464.292019	731.296997	349168	196013
	Value	6707.828482	1194.160584	428630	249154
	Profit	346.407042	42.299004	22135	12223
	Employee	70.766980	8.683595	4522.010000	2524.778281
Manufacturing	Asset	7403.004250	1454.921083	888361	492577
	Sale	7207.638833	2112.444703	864917	501679
	Value	2986.442750	799.121544	358373	196979
	Profit	211.933583	39.993255	25432	13322
	Employee	83.314333	31.089019	9997.720000	6294.309490
Medical	Asset	5046.570609	1218.444638	140799	131942
	Sale	3313.219713	758.216303	92439	85655
	Value	2561.614695	530.802245	71469	64663
	Profit	218.682796	44.051447	6101.250000	5509.560969
	Employee	46.518996	11.135955	1297.880000	1213.651734
Other	Asset	1850.250000	338.128984	58838	31375
	Sale	1620.784906	168.686773	51541	24593
	Value	1432.820755	297.869828	45564	24204
	Profit	115.089937	27.970560	3659.860000	2018.201371
	Employee	14.306604	2.313733	454.950000	216.327710
Retail	Asset	2939.845750	393.692369	235188	94605
	Sale	7395.453500	1746.187580	591636	263263
	Value	2103.863125	529.756409	168309	78304
	Profit	157.171875	31.734253	12574	5478.281027
	Employee	93.624000	15.726743	7489.920000	3093.832061
Transportation	Asset	4712.047359	888.954411	267644	163516
	Sale	4030.233275	1015.555708	228917	142669
	Value	1703.330282	313.841326	96749	58947
	Profit	224.762324	56.168925	12767	8287.585418
	Employee	30.946303	6.786270	1757.750000	1066.586615

Example 13.3. Analyze Survey Data with Missing Values

As described in the section "Missing Values" on page 134, the SURVEYMEANS procedure excludes an observation from the analysis if it has a missing value for the analysis variable or a nonpositive value for the WEIGHT variable.

However, if there is evidence indicating that the nonrespondents are different from the respondents for your study, you can use the DOMAIN statement to compute descriptive statistics "among respondents" from your survey data without imputation for nonrespondents.

Consider the ice cream example in the section "Stratified Sampling" on page 131. Suppose that some of the students failed to provide the amounts spent on ice cream, as shown in the following data set IceCream:

```
data IceCream;
   input Grade Spending @@; datalines;
7 7  7  7  8  .  9 10  7  .  7 10  7  3  8 20  8 19  7 2
7 .  9 15  8 16  7  6  7  6  7  6  9 15  8 17  8 14  9 .
9 8  9  7  7  3  7 12  7  4  9 14  8 18  9  9  7  2  7 1
7 4  7 11  9  8  8  .  8 13  7  .  9  .  9 11  7  2  7 9
;
data StudentTotals;
   input Grade _total_; datalines;
7 1824
8 1025
9 1151
;
```

Considering the possibility that those students who didn't respond spend differently than those students who did respond, you can create an indicator variable to identify the respondents and non-respondents using the following SAS DATA step:

```
data IceCream;
   set IceCream;
   if Spending=. then Indicator='Nonrespondent';
   else do;
      Indicator='Respondent';
      if (Spending < 10) then Group='less';
         else Group='more';
      end;
   if Grade=7 then Prob=20/1824;
   if Grade=8 then Prob=9/1025;
   if Grade=9 then Prob=11/1151;
   Weight=1/Prob;
```

Variable Indicator identifies a student in the data set as either a respondent or a non-respondent. Variable Group specifies whether a student spent more than $10 among the respondents.

The following SAS statements analyze the incomplete ice cream data:

```
title1 'Analysis of Ice Cream Spending';
proc surveymeans data=IceCream total=StudentTotals mean sum;
   stratum Grade / list;
   var Spending Group;
   weight Weight;
   domain Indicator;
run;
```

Output 13.3.1. Analyze Incomplete Ice Cream Data Excluding Observations with Missing Values

Analysis of Ice Cream Spending

The SURVEYMEANS Procedure

Data Summary

Number of Strata	3
Number of Observations	40
Sum of Weights	4000

Statistics

Variable	Mean	Std Error of Mean	Sum	Std Dev
Spending	9.770542	0.541381	32139	1780.792065
Group=less	0.515404	0.067092	1695.345455	220.690305
Group=more	0.484596	0.067092	1594.004040	220.690305

Output 13.3.1 shows the mean and total estimates excluding those students who failed to provide the spending amount on ice cream.

Output 13.3.2. Analyze Incomplete Ice Cream Data Treating Respondents as a Domain

Analysis of Ice Cream Spending

The SURVEYMEANS Procedure

Domain Analysis: Indicator

Indicator	Variable	Mean	Std Error of Mean	Sum	Std Dev
Nonrespondent	Spending	.	.	.	.
	Group=less	.	.	.	.
	Group=more	.	.	.	.
Respondent	Spending	9.770542	0.652347	32139	3515.126876
	Group=less	0.515404	0.067257	1695.345455	221.232029
	Group=more	0.484596	0.067257	1594.004040	221.232029

Output 13.3.2 shows the mean and total estimates treating respondents as a domain in the student population. Compared to the estimates in Output 13.3.1, the point estimates are the same, but the variance estimates are slightly higher.

References

Brick, J.M. and Kalton, G. (1996), "Handling Missing Data in Survey Research," *Statistical Methods in Medical Research*, 5, 215–238.

Cochran, W.G. (1977), *Sampling Techniques*, Third Edition, New York: John Wiley & Sons, Inc.

Fuller, W.A., Kennedy, W., Schnell, D., Sullivan, G., and Park, H.J. (1989), *PC CARP*, Ames, IA: Statistical Laboratory, Iowa State University.

Hidiroglou, M.A., Fuller, W.A., and Hickman, R.D. (1980), *SUPER CARP*, Ames, IA: Statistical Laboratory, Iowa State University.

Kalton, G. and Kaspyzyk, D. (1986), "The Treatment of Missing Survey Data," *Survey Methodology*, 12, 1–16.

Kish, L. (1965), *Survey Sampling*, New York: John Wiley & Sons, Inc.

Subject Index

B

biquartimax method, 19–20
biquartimin method, 19

C

CATMOD procedure
  iterative proportional fitting, 3
  maximum likelihood estimation, 3
  missing values, 6
  syntax, 3
  zeros, structural and sampling, 6
confidence intervals, FACTOR procedure, 21
confidence limits, FACTOR procedure, 21
covarimin method, 20
coverage displays
  FACTOR procedure, 23
Crawford-Ferguson method, 19–20

D

degrees of freedom
  SURVEYMEANS procedure, 136–137
domain analysis
  SURVEYMEANS procedure, 134, 136, 139
domain mean
  SURVEYMEANS procedure, 136
domain total
  SURVEYMEANS procedure, 137

E

empty stratum
  SURVEYMEANS procedure, 135, 137
equamax method, 19–20
exact logistic regression
  LOGISTIC procedure, 92, 94

F

factor parsimax method, 19–20
FACTOR procedure
  coverage displays, 23
  simplicity functions, 18, 23

G

GAM procedure
  comparing PROC GAM with PROC TPSPLINE, 61
  generalized additive model with binary data, 54
  ODS table names, 54
generalized Crawford-Ferguson method, 19–20

H

Harris-Kaiser method, 20

I

iterative proportional fitting
  estimation (CATMOD), 3
  formulas (CATMOD), 6

L

LOESS procedure
  automatic smoothing parameter selection, 78
  kd trees and blending, 77
  output table names, 81
LOGISTIC procedure
  displayed output, 99
  exact logistic regression, 92, 94
  ODS table names, 100
  output data sets, 98

M

maximum likelihood estimation
  CATMOD procedure, 3
missing values
  SURVEYMEANS procedure, 134, 146
ML factor analysis
  and confidence intervals, 21
MODECLUS procedure
  smoothing parameter, default, 115

O

oblimin method, 20
orthomax method, 19
output data sets
  LOGISTIC procedure, 98
output table names
  SURVEYMEANS procedure, 139

P

parsimax method, 19–20
Procrustes rotation, 20
promax method, 20

Q

quartimax method, 19–20
quartimin method, 21

S

salience of loadings, FACTOR procedure, 21
sampling zeros
  and structural zeros (CATMOD), 6
simplicity functions
  FACTOR procedure, 18, 23
smoothing parameter, default
  MODECLUS procedure, 115
statistical computation
  SURVEYMEANS procedure, 136
stratified cluster sample
  SURVEYMEANS procedure, 140
stratified sampling
  SURVEYMEANS procedure, 131
subdomain analysis
  SURVEYMEANS procedure, 134
subgroup analysis
  SURVEYMEANS procedure, 134
subpopulation analysis
  SURVEYMEANS procedure, 134
SURVEYMEANS procedure
  degrees of freedom, 136–137
  domain analysis, 134, 136, 139
  domain mean, 136
  domain total, 137
  domain variables, 133
  empty stratum, 135, 137
  missing values, 134, 146
  ODS table names, 139
  output table names, 139
  statistical computation, 136
  stratified cluster sample, 140
  stratified sampling, 131
  subdomain analysis, 134
  subgroup analysis, 134
  subpopulation analysis, 134
  t test, 136

T

t test
  SURVEYMEANS procedure, 136

V

varimax method, 19–20

Z

zeros, structural and sampling
  CATMOD procedure, 6

Syntax Index

A

ALPHA= option
  EXACT statement (LOGISTIC), 92
  MODEL statement (GAM), 45
  PROC FACTOR statement, 17

B

BY statement
  GAM procedure, 42

C

CATMOD procedure, MODEL statement, 3
  MISS= option, 5
  MISSING= option, 5
  ML option, 3
  ZERO= option, 5
  ZEROES= option, 5
  ZEROS= option, 5
CLASS statement
  GAM procedure, 43
CONTENTS= option
  TABLES statement (FREQ), 33
COVER= option
  PROC FACTOR statement, 17
COVS option
  PROC PHREG statement, 127
COVSANDWICH option
  PROC PHREG statement, 127

D

DATA= option
  PROC GAM statement, 42
  SCORE statement (GAM), 46
DETAILS option
  MODEL statement (LOESS), 75
DIST = option
  MODEL statement (GAM), 45
DOMAIN statement
  SURVEYMEANS procedure, 133

E

EPSILON = option
  MODEL statement (GAM), 45
ESTIMATE option
  EXACT statement (LOGISTIC), 92
EXACT statement
  LOGISTIC procedure, 92
EXACTONLY option
  PROC LOGISTIC statement, 91
EXACTOPTIONS option
  PROC LOGISTIC statement, 91

F

FACTOR procedure, PROC FACTOR statement, 17
  ALPHA= option, 17
  COVER= option, 17
  HKPOWER= option, 18
  PREROTATE= option, 18
  RCONVERGE= option, 18
  RITER= option, 18
  ROTATE= option, 18
  SE option, 21
FORMAT= option
  TABLES statement (FREQ), 33
FREQ procedure
  syntax, 33
FREQ procedure, TABLES statement, 33
  CONTENTS= option, 33
  FORMAT= option, 33
  OUTCUM option, 34
FREQ statement
  GAM procedure, 43

G

GAM procedure, 42
  syntax, 42
GAM procedure, BY statement, 42
GAM procedure, CLASS statement, 43
GAM procedure, FREQ statement, 43
GAM procedure, ID statement, 44
GAM procedure, MODEL statement
  ALPHA= option, 45
  DIST= option, 45
  EPSILON= option, 45
  MAXITER= option, 45
  METHOD= option, 45
GAM procedure, OUTPUT statement, 45
  OUT= option, 45
GAM procedure, PROC GAM statement, 42
  DATA= option, 42
GAM procedure, SCORE statement, 46
  DATA= option, 46
  OUT= option, 46

H

HKPOWER= option
  PROC FACTOR statement, 18

I

ID statement
  GAM procedure, 44
INTERP= option
  MODEL statement (LOESS), 76

J

JOINT option
  EXACT statement (LOGISTIC), 92
JOINTONLY option
  EXACT statement (LOGISTIC), 93

L

LOESS procedure, MODEL statement
  DETAILS option, 75
  INTERP= option, 76
  SELECT= option, 76
  SMOOTH= option, 77
  TRACEL option, 76
LOESS procedure, SCORE statement
  STEPS option, 77
LOGISTIC procedure, 91
  syntax, 91
LOGISTIC procedure, EXACT statement, 92
  ALPHA= option, 92
  ESTIMATE option, 92
  JOINT option, 92
  JOINTONLY option, 93
  ONESIDED option, 93
  OUTDIST= option, 93
LOGISTIC procedure, PROC LOGISTIC statement, 91
  EXACTONLY option, 91
  EXACTOPTIONS option, 91

M

MAXITER = option
  MODEL statement (GAM), 45
METHOD= option
  MODEL statement (GAM), 45
MISS= option
  MODEL statement (CATMOD), 5
MISSING= option
  MODEL statement (CATMOD), 5
ML option
  MODEL statement (CATMOD), 3
MODEL statement
  CATMOD procedure, 3
  TPSPLINE procedure, 44

O

ONESIDED option
  EXACT statement (LOGISTIC), 93
OUT= option
  OUTPUT statement (GAM), 45
  SCORE statement (GAM), 46
OUTCUM option
  TABLES statement (FREQ), 34
OUTDIST= option
  EXACT statement (LOGISTIC), 93
OUTPUT statement
  GAM procedure, 45

P

PHREG procedure, PROC PHREG statement
  COVS option, 127
  COVSANDWICH option, 127
PREROTATE= option
  PROC FACTOR statement, 18
PROC FACTOR statement
  See FACTOR procedure
PROC GAM statement
  See GAM procedure
PROC LOGISTIC statement
  See LOGISTIC procedure

R

RCONVERGE= option
  PROC FACTOR statement, 18
RITER= option
  PROC FACTOR statement, 18
ROTATE= option
  PROC FACTOR statement, 18

S

SCORE statement, GAM procedure, 46
SE option
  PROC FACTOR statement, 21
SELECT= option
  MODEL statement (LOESS), 76
SMOOTH= option
  MODEL statement (LOESS), 77
STEPS option
  SCORE statement (LOESS), 77
SUBGROUP statement
  SURVEYMEANS procedure, 133
SURVEYMEANS procedure
  syntax, 133
SURVEYMEANS procedure, DOMAIN statement, 133

T

TABLES statement
  FREQ procedure, 33
TPSPLINE procedure, MODEL statement, 44
TRACEL option
  MODEL statement (LOESS), 76

Z

ZERO= option
  MODEL statement (CATMOD), 5
ZEROES= option
  MODEL statement (CATMOD), 5
ZEROS= option
  MODEL statement (CATMOD), 5

Your Turn

If you have comments or suggestions about *SAS/STAT® Software: Changes and Enhancements, Release 8.1*, please send them to us on a photocopy of this page or send us electronic mail.

For comments about this book, please return the photocopy to

SAS Institute
Publications Division
SAS Campus Drive
Cary, NC 27513
email: yourturn@sas.com

For suggestions about the software, please return the photocopy to

SAS Institute
Technical Support Division
SAS Campus Drive
Cary, NC 27513
email: suggest@sas.com

*Welcome * Bienvenue * Willkommen * Yohkoso * Bienvenido*

SAS® Institute Publishing Is Easy to Reach

Visit our Web page located at www.sas.com/pubs

You will find product and service details, including

- **sample chapters**
- **tables of contents**
- **author biographies**
- **book reviews**

Learn about

- **regional user-group conferences**
- **trade-show sites and dates**
- **authoring opportunities**
- **custom textbooks**

Explore all the services that SAS Institute Publishing has to offer!

Your Listserv Subscription Automatically Brings the News to You

Do you want to be among the first to learn about the latest books and services available from SAS Institute Publishing? Subscribe to our listserv **newdocnews-l** and, once each month, you will automatically receive a description of the newest books and which environments or operating systems and SAS release(s) that each book addresses.

To subscribe,

1. Send an e-mail message to **listserv@vm.sas.com.**
2. Leave the "Subject" line blank.
3. Use the following text for your message:

   subscribe NEWDOCNEWS-L *your-first-name your-last-name*

   For example: subscribe NEWDOCNEWS-L John Doe

Create Customized Textbooks Quickly, Easily, and Affordably

SelecText® offers instructors at U.S. colleges and universities a way to create custom textbooks for courses that teach students how to use SAS software.

For more information, see our Web page at **www.sas.com/selectext**, or contact our SelecText coordinators by sending e-mail to **selectext@sas.com**.

You're Invited to Publish with SAS Institute's User Publishing Program

If you enjoy writing about SAS software and how to use it, the User Publishing Program at SAS Institute offers a variety of publishing options. We are actively recruiting authors to publish books, articles, and sample code. Do you find the idea of writing a book or an article by yourself a little intimidating? Consider writing with a co-author. Keep in mind that you will receive complete editorial and publishing support, access to our users, technical advice and assistance, and competitive royalties. Please contact us for an author packet. E-mail us at **sasbbu@sas.com** or call 919-677-8000, then press 1-6479. See the SAS Institute Publishing Web page at **www.sas.com/pubs** for complete information.

See *Observations*®, Our Online Technical Journal

Feature articles from *Observations*®: *The Technical Journal for SAS® Software Users* are now available online at **www.sas.com/obs**. Take a look at what your fellow SAS software users and SAS Institute experts have to tell you. You may decide that you, too, have information to share. If you are interested in writing for *Observations*, send e-mail to **sasbbu@sas.com** or call 919-677-8000, then press 1-6479.

Book Discount Offered at SAS Public Training Courses!

When you attend one of our SAS Public Training Courses at any of our regional Training Centers in the U.S., you will receive a 15% discount on book orders that you place during the course. Take advantage of this offer at the next course you attend!

SAS Institute
SAS Campus Drive
Cary, NC 27513-2414
Fax 919-677-4444

E-mail: sasbook@sas.com
Web page: www.sas.com/pubs
To order books, call Fulfillment Services at 800-727-3228*
For other SAS Institute business, call 919-677-8000*

*** Note:** Customers outside the U.S. should contact their local SAS office.