SAS/STAT® 15.1 User's Guide
The GAMPL Procedure

This document is an individual chapter from *SAS/STAT® 15.1 User's Guide*.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2018. *SAS/STAT® 15.1 User's Guide*. Cary, NC: SAS Institute Inc.

SAS/STAT® 15.1 User's Guide

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

November 2018

Chapter 46
The GAMPL Procedure

Contents

Overview: GAMPL Procedure

The GAMPL procedure is a high-performance procedure that fits generalized additive models that are based on low-rank regression splines (Wood 2006). This procedure provides powerful tools for nonparametric regression and smoothing.

Generalized additive models are extensions of generalized linear models. They relax the linearity assumption in generalized linear models by allowing spline terms in order to characterize nonlinear dependency structures. Each spline term is constructed by the thin-plate regression spline technique (Wood 2003). A roughness penalty is applied to each spline term by a smoothing parameter that controls the balance between goodness of fit and the roughness of the spline curve. PROC GAMPL fits models for standard distributions in the exponential family, such as normal, Poisson, and gamma distributions.

PROC GAMPL runs in either single-machine mode or distributed mode.

NOTE: Distributed mode requires SAS High-Performance Statistics.

PROC GAMPL Features

PROC GAMPL offers the following basic features:

- estimates the regression parameters of a generalized additive model that has fixed smoothing parameters by using penalized likelihood estimation
- estimates the smoothing parameters of a generalized additive model by using either the performance iteration method or the outer iteration method
- estimates the regression parameters of a generalized linear model by using maximum likelihood techniques
- tests the total contribution of each spline term based on the Wald statistic
- provides model-building syntax in the CLASS statement and effect-based parametric effects in the MODEL statement, which are used in other SAS/STAT analytic procedures (in particular, the GLM, LOGISTIC, GLIMMIX, and MIXED procedures)
- provides response-variable options
- enables you to construct a spline term by using multiple variables
- provides control options for constructing a spline term, such as fixed degrees of freedom, initial smoothing parameter, fixed smoothing parameter, smoothing parameter search range, user-supplied knot values, and so on
- provides multiple link functions for any distribution
- provides a WEIGHT statement for weighted analysis
- provides a FREQ statement for grouped analysis

- provides an OUTPUT statement to produce a data set that has predicted values and other observation-wise statistics
- produces graphs via ODS Graphics

Because the GAMPL procedure is a high-performance analytical procedure, it also does the following:

- enables you to run in distributed mode on a cluster of machines that distribute the data and the computations
- enables you to run in single-machine mode on the server where SAS is installed
- exploits all the available cores and concurrent threads, regardless of execution mode

For more information, see the section "Processing Modes" (Chapter 2, *SAS/STAT User's Guide: High-Performance Procedures*).

PROC GAMPL Contrasted with PROC GAM

Both the GAMPL procedure and the GAM procedure in SAS/STAT software fit generalized additive models. However, the GAMPL procedure uses different approaches for constructing spline basis expansions, fitting generalized additive models, and testing smoothing components. The GAMPL procedure focuses on automatic smoothing parameter selection by using global model-evaluation criteria to find optimal models. The GAM procedure focuses on constructing models by fitting partial residuals against each smoothing term. In general, you should not expect similar results from the two procedures. The following subsections summarize the differences. For more information about the GAM procedure, see Chapter 45, "The GAM Procedure."

Constructing Spline Basis Expansions

The GAMPL procedure uses thin-plate regression splines to construct basis expansions for each spline term, and each term allows multiple variables. The GAM procedure uses univariate or bivariate smoothing splines to construct basis expansions, and each term allows only one or two variables. The thin-plate regression splines that PROC GAMPL uses are low-rank approximations to multivariate smoothing splines. The GAM procedure also allows loess smoothers.

Fitting Generalized Additive Models

The GAMPL procedure fits a generalized additive model that has fixed smoothing parameters by using a global design matrix and a roughness penalty matrix. The GAM procedure uses partial residuals to fit against single smoothing terms. For models that have unknown smoothing parameters, the GAMPL procedure estimates smoothing parameters simultaneously by optimizing global criteria such as generalized cross validation (GCV) and the unbiased risk estimator (UBRE). The GAM procedure estimates each smoothing parameter by optimizing the local criterion GCV one spline term at a time.

Distribution Families and Link Functions

The GAMPL procedure supports all the distribution families and all the link functions that the GAM procedure supports. In addition, PROC GAMPL fits models in the negative binomial and Tweedie families. PROC GAMPL supports any link function for each distribution, whereas PROC GAM supports only the canonical link for each distribution.

Testing Smoothing Components

The GAMPL procedure tests the total contribution for a spline term, including both linear and nonlinear trends. The GAM procedure tests the existence of nonlinearity for a spline term beyond the linear trend.

Model Inference

A global Bayesian posterior covariance matrix is available for models that are fitted by the GAMPL procedure. The confidence limits for prediction of each observation are available, in addition to componentwise confidence limits. For generalized additive models that are fitted by the GAM procedure, only the componentwise confidence limits are available, and they are based on the partial residuals for each smoothing term. The degrees of freedom for generalized additive models that are fitted by the GAMPL procedure is defined as the trace of the degrees-of-freedom matrix. The degrees of freedom for generalized additive models that are fitted by the GAM procedure is approximated by summing the trace of the smoothing matrix for each smoothing term.

Getting Started: GAMPL Procedure

This example concerns the proportions and demographic and geographic characteristics of votes that were cast in 3,107 counties in the United States in the 1980 presidential election. You can use the data set sashelp.Vote1980 directly from the SASHELP library or download it from the StatLib Datasets Archive (Vlachos 1998). For more information about the data set, see Pace and Barry (1997).

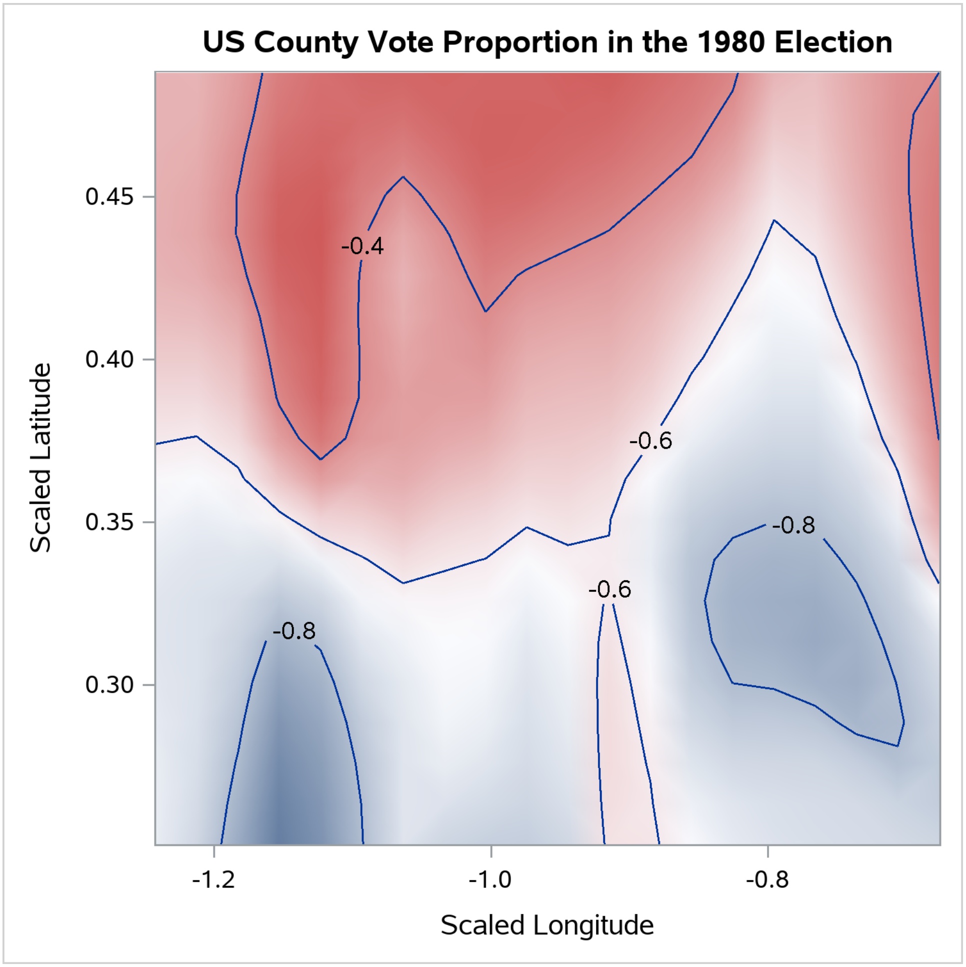
The data set contains 3,107 observations and seven variables. The dependent variable LogVoteRate is the logarithm transformation of the proportion of the county population who voted for any candidate. The six explanatory variables are the number of people in the county 18 years of age or older (Pop), the number of people in the county who have a 12th-grade or higher education (Edu), the number of owned housing units (Houses), the aggregate income (Income), and the scaled longitude and latitude of geographic centroids (Longitude and Latitude).

The following statements produce the plot of LogVoteRate with respect to the geographic locations Longitude and Latitude:

```
%let off0 = offsetmin=0 offsetmax=0
            linearopts=(thresholdmin=0 thresholdmax=0);
proc template;
   define statgraph surface;
      dynamic _title _z;
      begingraph / designwidth=defaultDesignHeight;
         entrytitle _title;
         layout overlay / xaxisopts=(&off0) yaxisopts=(&off0);
            contourplotparm z=_z y=Latitude x=Longitude / gridded=FALSE;
         endlayout;
      endgraph;
   end;
run;

proc sgrender data=sashelp.Vote1980 template=surface;
   dynamic _title = 'US County Vote Proportion in the 1980 Election'
          _z     = 'LogVoteRate';
run;
```

Figure 46.1 shows the map of the logarithm transformation of the proportion of the county population who voted for any candidate in the 1980 US presidential election.

Figure 46.1 US County Vote Proportion in the 1980 Election



The objective is to explore the nonlinear dependency structure between the dependent variable and demographic variables (Pop, Edu, Houses, and Income), in addition to the spatial variations on geographic variables (Longitude and Latitude). The following statements use thin-plate regression splines to fit a generalized additive model:

```
ods graphics on;

proc gampl data=sashelp.Vote1980 plots seed=12345;
   model LogVoteRate = spline(Pop    ) spline(Edu) spline(Houses)
                       spline(Income) spline(Longitude Latitude);
   id Longitude Latitude;
   output out=VotePred;
run;
```

With ODS Graphics enabled by the first statement, the PLOTS option in the PROC GAMPL statement requests a smoothing component panel of fitted spline terms. The SEED option specifies the random seed so that you can reproduce the analysis.

The default output from this analysis is presented in Figure 46.2 through Figure 46.10.

The "Performance Information" table in Figure 46.2 shows that PROC GAMPL executed in single-machine mode (that is, on the server where SAS is installed). When high-performance procedures run in single-machine mode, they use concurrently scheduled threads. In this case, four threads were used.

Figure 46.2 Performance Information

The GAMPL Procedure

Performance Information	
Execution Mode	Single-Machine
Number of Threads	12

Figure 46.3 displays the "Model Information" table. The response variable LogVoteRate is modeled by using a normal distribution whose mean is modeled by an identity link function. The GAMPL procedure uses the performance iteration method and the GCV criterion as the fitting criterion. PROC GAMPL searches for the optimum smoothing parameters by using the Newton-Raphson algorithm to optimize the fitting criterion. The random number seed is set to 12,345. Random number generation is used for sampling from observations to form spline knots and truncated eigendecomposition. Changing the random number seed might yield slightly different model fits.

Figure 46.3 Model Information

Model Information	
Data Source	SASHELP.VOTE1980
Response Variable	LogVoteRate
Distribution	Normal
Link Function	Identity
Fitting Method	Performance Iteration
Fitting Criterion	GCV
Optimization Technique for Smoothing	Newton-Raphson
Random Number Seed	12345

Figure 46.4 displays the "Number of Observations" table. All 3,107 observations in the data set are used in the analysis. For data sets that have missing or invalid values, the number of used observations might be less than the number of observations read.

Figure 46.4 Number of Observations

Number of Observations Read	3107
Number of Observations Used	3107

Figure 46.5 displays the convergence status of the performance iteration method.

Figure 46.5 Convergence Status

```
The performance iteration converged after 3 steps.
```

Figure 46.6 shows the “Fit Statistics” table. The penalized log likelihood and the roughness penalty are displayed. You can use effective degrees of freedom to compare generalized additive models with generalized linear models that do not have spline transformations. Information criteria such as Akaike’s information criterion (AIC), Akaike’s bias-corrected information criterion (AICC), and Schwarz Bayesian information criterion (BIC) can also be used for comparisons. These criteria penalize the –2 log likelihood for effective degrees of freedom. The GCV criterion is used to compare against other generalized additive models or models that are penalized.

Figure 46.6 Fit Statistics

Fit Statistics	
Penalized Log Likelihood	2729.51482
Roughness Penalty	0.25787
Effective Degrees of Freedom	48.70944
Effective Degrees of Freedom for Error	3055.44725
AIC (smaller is better)	-5361.86863
AICC (smaller is better)	-5360.28467
BIC (smaller is better)	-5067.59479
GCV (smaller is better)	0.01042

.

The “Parameter Estimates” table in Figure 46.7 shows the regression parameter and dispersion parameter estimates. In this model, the intercept is the only regression parameter because (1) all variables are characterized by spline terms and no parametric effects are present and (2) the intercept absorbs the constant effect that is extracted from each spline term to make fitted splines identifiable. The dispersion parameter is estimated by maximizing the likelihood, given other model parameters.

Figure 46.7 Regression Parameter Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept	1	-0.576234	0.001803	102119.645	<.0001
Dispersion	1	0.010103	0.014287		

The "Estimates for Smoothing Components" table is shown in Figure 46.8. For each spline term, the effective degrees of freedom, the estimated smoothing parameter, and the corresponding roughness penalty are displayed. The table also displays additional information about spline terms, such as the number of parameters, penalty matrix rank, and number of spline knots.

Figure 46.8 Estimates for Smoothing Components

Estimates for Smoothing Components

Component	Effective DF	Smoothing Parameter	Roughness Penalty	Number of Parameters	Rank of Penalty Matrix	Number of Knots
Spline(Pop)	7.80559	0.0398	0.0114	9	10	2000
Spline(Edu)	7.12453	0.2729	0.0303	9	10	2000
Spline(Houses)	7.20940	0.1771	0.0370	9	10	2000
Spline(Income)	5.92854	0.7498	0.0488	9	10	2000
Spline(Longitude Latitude)	18.64138	0.000359	0.1304	19	20	2000

Figure 46.9 displays the hypothesis testing results for each smoothing component. The null hypothesis for each spline term is whether the total dependency on each variable is 0. The effective degrees of freedom for both fit and test is displayed.

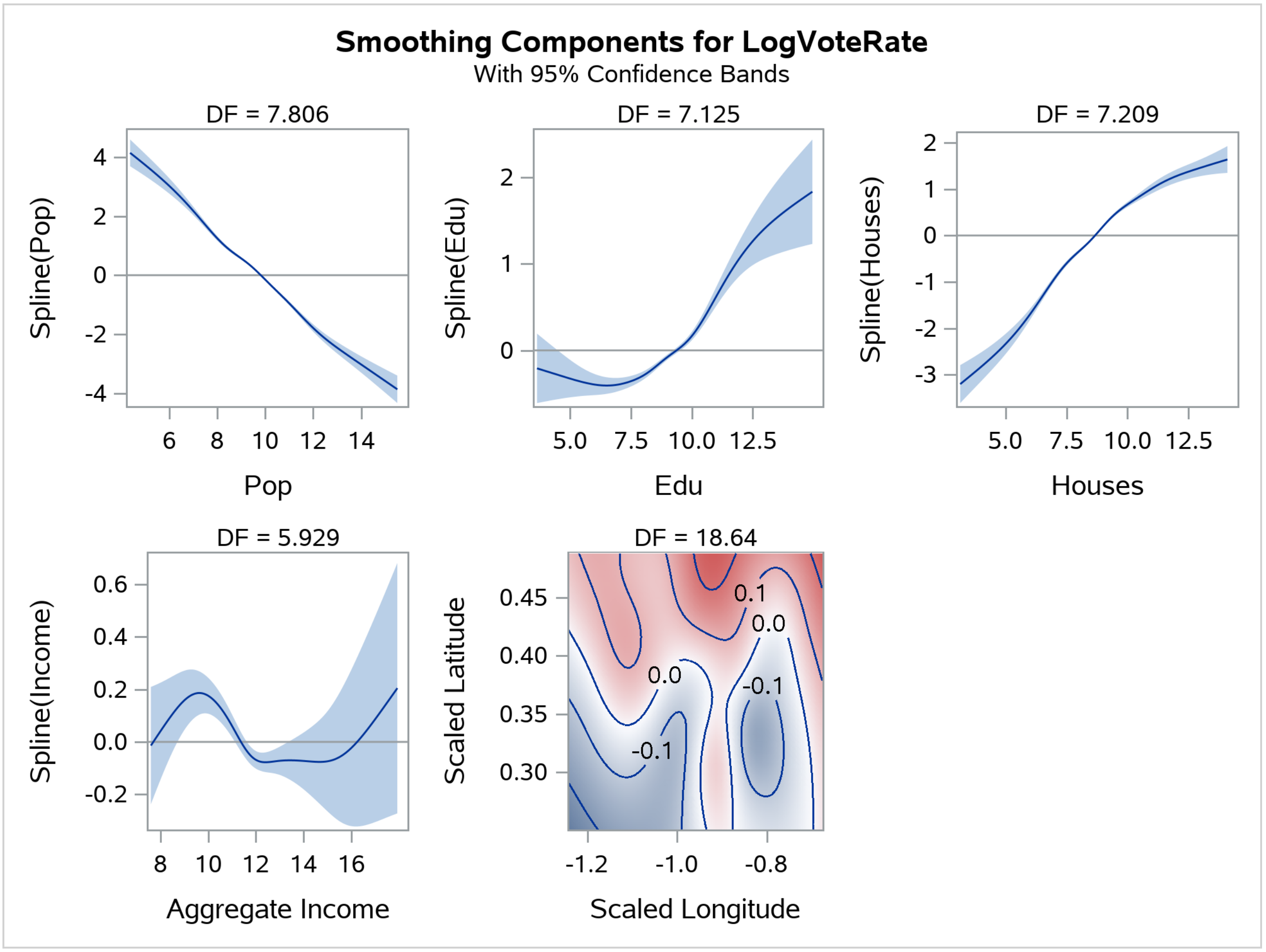
Figure 46.9 Tests for Smoothing Components

Tests for Smoothing Components

Component	Effective DF	Effective DF for Test	F Value	Pr > F
Spline(Pop)	7.80559	8	1443.64	<.0001
Spline(Edu)	7.12453	8	153.94	<.0001
Spline(Houses)	7.20940	8	1213.94	<.0001
Spline(Income)	5.92854	7	43.17	<.0001
Spline(Longitude Latitude)	18.64138	19	1619.15	<.0001

Figure 46.10 displays the "Smoothing Component Panel" for all the spline terms used in the model. It displays predicted spline curves and 95% Bayesian posterior confidence bands for each univariate spline term.

Figure 46.10 Smoothing Component Panel

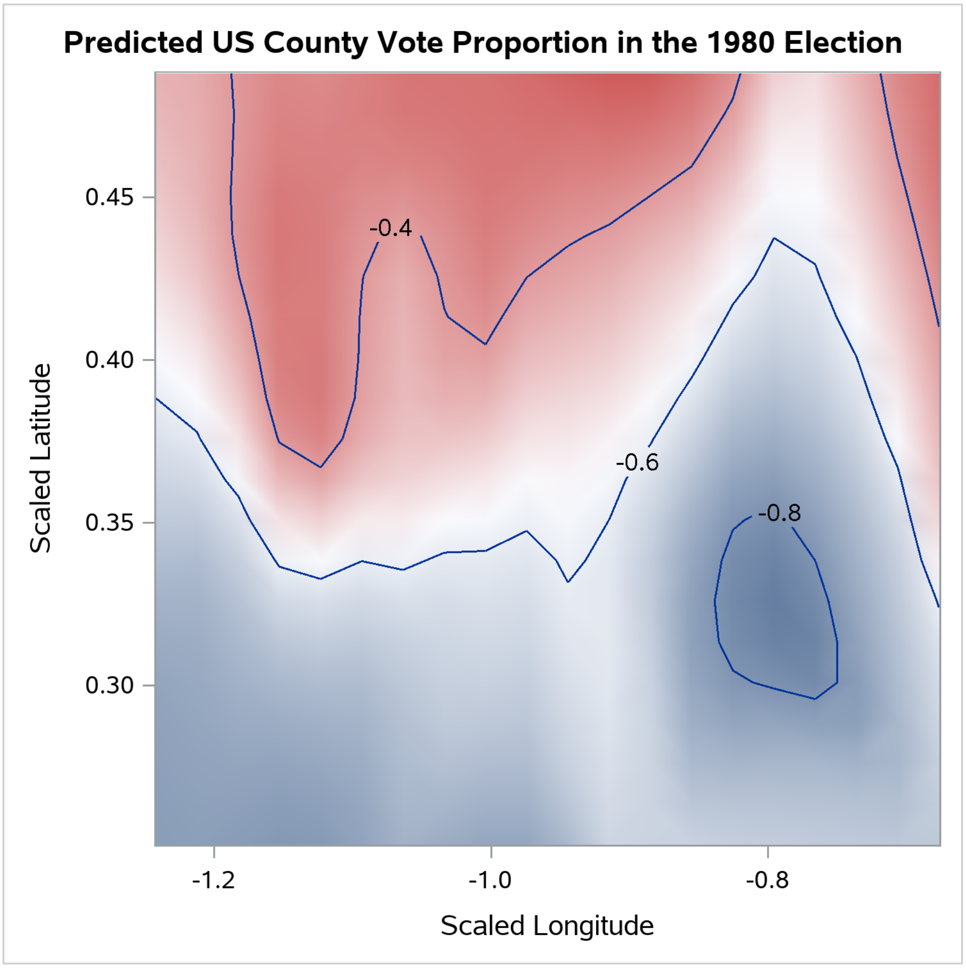


The preceding program contains an ID statement and an OUTPUT statement. The ID statement transfers the specified variables (Longitude and Latitude) from the input data set, sashelp.Vote1980, to the output data set, VotePred. The OUTPUT statement requests the prediction for each observation by default and saves the results in the data set VotePred. The following run of the SGRENDER procedure produces the fitted surface of the log vote proportion in the 1980 presidential election:

```
proc sgrender data=VotePred template=surface;
   dynamic _title='Predicted US County Vote Proportion in the 1980 Election'
           _z    ='Pred';
run;
```

Figure 46.11 shows the map of predictions of the logarithm transformation of the proportion of county population who voted for any candidates in the 1980 US presidential election from the fitted generalized additive model.

Figure 46.11 Predicted US County Vote Proportion in the 1980 Election



Compared to the map of the logarithm transformations of the proportion of votes cast shown in Figure 46.1, the map of the predictions of the logarithm transformations of the proportion of votes cast has a smoother surface.

Syntax: GAMPL Procedure

The following statements are available in the GAMPL procedure:

> **PROC GAMPL** < *options* > ;
> **CLASS** *variable* < **(***options***)** >. . . < *variable* < **(***options***)** > > < / *global-options* > ;
> **MODEL** *response* < **(***response-options***)** > **=** < **PARAM(***effects***)** >
> < *spline-effects* > < / *model-options* > ;
> **MODEL** *events / trials* **=** < **PARAM(***effects***)** > < *spline-effects* > < / *model-options* > ;
> **OUTPUT** < **OUT=***SAS-data-set* > < *keyword* < **=***name* > >. . . < *keyword* < **=***name* > > < / *options* > ;
> **PERFORMANCE** *performance-options* ;
> **FREQ** *variable* ;
> **ID** *variables* ;
> **WEIGHT** *variable* ;

The PROC GAMPL statement and at least one MODEL statement are required. The CLASS statement can appear multiple times. If a CLASS statement is specified, it must precede the MODEL statements. The following sections describe the PROC GAMPL statement and then describe the other statements in alphabetical order.

PROC GAMPL Statement

PROC GAMPL < *options* > ;

The PROC GAMPL statement invokes the procedure. Table 46.1 summarizes the available options in the PROC GAMPL statement by function. The options are then described fully in alphabetical order.

Table 46.1 PROC GAMPL Statement Options

Option	Description
Basic Options	
ALPHA=	Specifies a global significance level
DATA=	Specifies the input data set
NAMELEN=	Limits the length of effect names
SEED=	Sets the seed for pseudorandom number generation
Display Options	
ITDETAILS	Displays the "Iteration History" table
NOCLPRINT	Limits or suppresses the display of classification variable levels
NOPRINT	Suppresses ODS output
PLOTS=	Controls plots that are produced through ODS Graphics
Optimization Subject Options	
PLIKEOPTIONS	Sets optimization parameters for likelihood estimation
SMOOTHOPTIONS	Sets optimization parameters for smoothing parameter estimation

Table 46.1 *continued*

Option	Description
Tolerance Options	
SINGCHOL=	Tunes the singularity criterion for Cholesky decompositions
SINGSWEEP=	Tunes the singularity criterion for the sweep operator
SINGULAR=	Tunes the general singularity criterion
User-Defined Format Options	
FMTLIBXML=	Specifies the file reference for a format stream

You can specify the following *options* in the PROC GAMPL statement.

ALPHA=*number*

specifies a global significance level for the hypothesis testing of smoothing components and the construction of Bayesian confidence bands of predictions. The confidence level is 1 – *number*. The value of *number* must be between 0 and 1; the default is 0.05. You can override this global significance level for Bayesian confidence bands of predictions by specifying the ALPHA= option in the OUTPUT statement.

DATA=*SAS-data-set*

names the input SAS data set for PROC GAMPL to use. The default is the most recently created data set.

If the procedure executes in distributed mode, the input data are distributed to memory on the appliance nodes and analyzed in parallel, unless the data are already distributed in the appliance database. In that case, PROC GAMPL reads the data alongside the distributed database. For information about the various execution modes, see the section “Processing Modes” (Chapter 2, *SAS/STAT User’s Guide: High-Performance Procedures*). For information about the alongside-the-database model, see the section “Alongside-the-Database Execution” (Chapter 2, *SAS/STAT User’s Guide: High-Performance Procedures*).

FMTLIBXML=*file-ref*

specifies the file reference for the XML stream that contains user-defined format definitions. User-defined formats are handled differently in a distributed computing environment than they are in other SAS products. For information about how to generate an XML stream for your formats, see the section “Working with Formats” (Chapter 2, *SAS/STAT User’s Guide: High-Performance Procedures*).

ITDETAILS

adds to the “Iteration History” table the current values of the parameter estimates and their gradients. If the optimization algorithm is used to determine at least one smoothing parameter, the table lists values for smoothing parameters. If all smoothing parameters are fixed or a parametric generalized linear model is specified, the table lists values for regression parameters. These quantities are reported only for parameters that participate in the optimization.

NAMELEN=*number*

specifies the length to which long effect names are shortened. The value of *number* must be an integer greater than or equal to 20. By default, NAMELEN=20.

NOCLPRINT< **=***number* >

suppresses the display of the “Class Level Information” table if you do not specify *number*. If you specify *number*, the values of the classification variables are displayed for only those variables whose number of levels is less than *number*. Specifying a *number* helps reduce the size of the “Class Level Information” table if some classification variables have a large number of levels.

NOPRINT

suppresses the generation of ODS output.

PLIKEOPTIONS(*optimization-parameters***)**

sets optimization parameters for either maximum or penalized likelihood estimation. For more information about how to specify *optimization-parameters*, see the section “Optimization Parameters” on page 3286.

PLOTS < **(***global-plot-option***)** > < **=** *plot-requests* < **(***option***)** > >

controls the plots that are produced through ODS Graphics. When ODS Graphics is enabled, PROC GAMPL produces by default a panel of plots of partial prediction curves or surfaces of smoothing components.

ODS Graphics must be enabled before plots can be requested. For example:

```
ods graphics on;

proc gampl plots;
   model y=spline(x1) spline(x2);
run;

ods graphics off;
```

For more information about enabling and disabling ODS Graphics, see the section “Enabling and Disabling ODS Graphics” on page 623 in Chapter 21, “Statistical Graphics Using ODS.”

You can specify the following *global-plot-option*, which applies to the smoothing component plots that the GAMPL procedure generates:

UNPACK | UNPACKPANEL

suppresses paneling. By default, multiple smoothing component plots can appear in some output panels. Specify UNPACK to get each plot individually.

You can specify the following *plot-requests*:

ALL

requests that all default plots be produced.

COMPONENTS < **(***component-option***)** >

plots a panel of smoothing components of the fitted model. You can specify the following *component-option*:

COMMONAXES

requests that the smoothing component plots use a common vertical axis except for bivariate contour plots. This option enables you to visually judge the relative effect size.

NONE
suppresses all plots.

SEED=*number*
specifies an integer that is used to start the pseudorandom number generator for subset sampling from observations to form knots if necessary and for truncated eigendecomposition. If you do not specify this option, or if *number* ≤ 0, the seed is generated from the time of day, which is read from the computer's clock.

SINGCHOL=*number*
tunes the singularity criterion in Cholesky decompositions. The default is 1E4 times the machine epsilon; this product is approximately 1E–12 on most computers.

SINGSWEEP=*number*
tunes the singularity criterion in sweep operations that determine collinearity between spline basis expansions. The default is 1E–8.

SINGULAR=*number*
tunes the general singularity criterion that is applied in sweep and inversion operations. The default is 1E4 times the machine epsilon; this product is approximately 1E–12 on most computers.

SMOOTHOPTIONS(*optimization-parameters***)**
specifies optimization parameters for smoothing parameter estimation. For more information about how to specify *optimization-parameters*, see the section "Optimization Parameters" on page 3286.

Optimization Parameters

You can specify *optimization-parameters* for both the PLIKEOPTIONS and SMOOTHOPTIONS options. Depending on the modeling context, some optimization parameters might have no effect. For parametric generalized linear models or generalized additive models that have fixed smoothing parameters, any optimization parameters that you specify in the SMOOTHOPTIONS option are ignored. For the performance iteration method, only the ABSFCONV=, FCONV=, and MAXITER= options are effective for PLIKEOPTIONS. The optimization algorithm is considered to have converged when any one of the convergence criteria that are specified in *optimization-parameters* is satisfied. Table 46.2 lists the available optimization parameters for both the PLIKEOPTIONS and SMOOTHOPTIONS options.

Table 46.2 Optimization Parameters

Option	Description
ABSCONV=	Tunes the absolute function convergence criterion
ABSFCONV=	Tunes the absolute function difference convergence criterion
ABSGCONV=	Tunes the absolute gradient convergence criterion
FCONV=	Tunes the relative function difference convergence criterion
GCONV=	Tunes the relative gradient convergence criterion
MAXFUNC=	Specifies the maximum number of function evaluations in any optimization
MAXITER=	Chooses the maximum number of iterations in any optimization
MAXTIME=	Specifies the upper limit of CPU time (in seconds) for any optimization
MINITER=	Specifies the minimum number of iterations in any optimization
TECHNIQUE=	Selects the optimization technique

You can specify the following *optimization-parameters*:

ABSCONV=*r*

ABSTOL=*r*

specifies an absolute function convergence criterion. For minimization, termination requires $f(\boldsymbol{\psi}^{(k)}) \leq r$, where $\boldsymbol{\psi}$ is the vector of parameters in the optimization and $f(\cdot)$ is the objective function. The default value of *r* is the negative square root of the largest double-precision value, which serves only as a protection against overflow.

ABSFCONV=*r* <*n*>

ABSFTOL=*r* <*n*>

specifies an absolute function difference convergence criterion. For all techniques except NMSIMP, termination requires a small change of the function value in successive iterations,

$$|f(\boldsymbol{\psi}^{(k-1)}) - f(\boldsymbol{\psi}^{(k)})| \leq r$$

where $\boldsymbol{\psi}$ denotes the vector of parameters that participate in the optimization and $f(\cdot)$ is the objective function. The same formula is used for the NMSIMP technique, but $\boldsymbol{\psi}(k)$ is defined as the vertex that has the lowest function value and $\boldsymbol{\psi}^{(k-1)}$ is defined as the vertex that has the highest function value in the simplex. The optional integer value *n* specifies the number of successive iterations for which the criterion must be satisfied before the process terminates. By default, ABSFCONV=0.

ABSGCONV=*r* <*n*>

ABSGTOL=*r* <*n*>

specifies an absolute gradient convergence criterion. Termination requires the maximum absolute gradient element to be small,

$$\max_j |g_j(\boldsymbol{\psi}^{(k)})| \leq r$$

where $\boldsymbol{\psi}$ denotes the vector of parameters that participate in the optimization and $g_j(\cdot)$ is the gradient of the objective function with respect to the *j*th parameter. This criterion is not used by the NMSIMP technique. The default value is *r* = 1E–8. The optional integer value *n* specifies the number of successive iterations for which the criterion must be satisfied before the process terminates.

FCONV=*r* <*n*>

FTOL=*r* <*n*>

specifies a relative function difference convergence criterion. For all techniques except NMSIMP, termination requires a small relative change of the function value in successive iterations,

$$\frac{|f(\boldsymbol{\psi}^{(k)}) - f(\boldsymbol{\psi}^{(k-1)})|}{|f(\boldsymbol{\psi}^{(k-1)})|} \leq r$$

where $\boldsymbol{\psi}$ denotes the vector of parameters that participate in the optimization and $f(\cdot)$ is the objective function. The same formula is used for the NMSIMP technique, but $\boldsymbol{\psi}^{(k)}$ is defined as the vertex that has the lowest function value and $\boldsymbol{\psi}^{(k-1)}$ is defined as the vertex that has the highest function value in the simplex.

The default value is *r* = $2\times\epsilon$, where ϵ is the machine precision. The optional integer value *n* specifies the number of successive iterations for which the criterion must be satisfied before the process terminates.

GCONV=*r* < *n* >

GTOL=*r* < *n* >

specifies a relative gradient convergence criterion. For all techniques except CONGRA and NMSIMP, termination requires that the normalized predicted function reduction be small,

$$\frac{g(\psi^{(k)})'[\mathbf{H}^{(k)}]^{-1}g(\psi^{(k)})}{|f(\psi^{(k)})|} \leq r$$

where ψ denotes the vector of parameters that participate in the optimization, $f(\cdot)$ is the objective function, and $g(\cdot)$ is the gradient. For the CONGRA technique (where a reliable Hessian estimate, $\mathbf{H}$, is not available), the following criterion is used:

$$\frac{\| g(\psi^{(k)}) \|_2^2 \quad \| s(\psi^{(k)}) \|_2}{\| g(\psi^{(k)}) - g(\psi^{(k-1)}) \|_2 \, |f(\psi^{(k)})|} \leq r$$

This criterion is not used by the NMSIMP technique. The default value is *r* = 1E–8. The optional integer value *n* specifies the number of successive iterations for which the criterion must be satisfied before the process terminates.

MAXFUNC=*n*

MAXFU=*n*

specifies the maximum number of function calls in the optimization process. The default values are as follows, depending on the optimization technique that you specify in the TECHNIQUE= option:

- TRUREG, NRRIDG, NEWRAP: *n* = 125
- QUANEW, DBLDOG: *n* = 500
- CONGRA: *n* = 1,000
- NMSIMP: *n* = 3,000

The optimization can terminate only after completing a full iteration. Therefore, the number of function calls that are actually performed can exceed *n*.

MAXITER=*n*

MAXIT=*n*

specifies the maximum number of iterations in the optimization process. The default values are as follows, depending on the optimization technique that you specify in the TECHNIQUE option:

- TRUREG, NRRIDG, NEWRAP: *n* = 50
- QUANEW, DBLDOG: *n* = 200
- CONGRA: *n* = 400
- NMSIMP: *n* = 1,000

These default values also apply when *n* is specified as a missing value.

MAXTIME=*r*

specifies an upper limit of *r* seconds of CPU time for the optimization process. The default value is the largest floating-point double representation of your computer. The time that you specify in this option is checked only once at the end of each iteration. Therefore, the actual running time can be longer than *r*.

MINITER=*n*

MINIT=*n*

specifies the minimum number of iterations. If you request more iterations than are actually needed for convergence to a stationary point, the optimization algorithms might behave strangely. For example, the effect of rounding errors can prevent the algorithm from continuing for the required number of iterations. By default, MINITER=0.

TECHNIQUE=*keyword*

TECH=*keyword*

specifies the optimization technique for obtaining smoothing parameter estimates and regression parameter estimates. You can choose from the following techniques:

CONGRA	performs a conjugate-gradient optimization.
DBLDOG	performs a version of double-dogleg optimization.
NEWRAP	performs a Newton-Raphson optimization with line search.
NMSIMP	performs a Nelder-Mead simplex optimization.
NONE	performs no optimization.
NRRIDG	performs a Newton-Raphson optimization with ridging.
QUANEW	performs a dual quasi-Newton optimization.
TRUREG	performs a trust-region optimization

By default, TECHNIQUE=NEWRAP for the performance iteration (METHOD=PERFORMANCE), and TECHNIQUE=QUANEW for the outer iteration (METHOD=OUTER).

For more information, see the section "Choosing an Optimization Technique" on page 3312.

CLASS Statement

CLASS *variable* < **(***options***)** >. . . < *variable* < **(***options***)** > > < / *global-options* > **;**

The CLASS statement names the classification variables to be used as explanatory variables in the analysis. The CLASS statement must precede the MODEL statement. You can list the response variable for binary models in the CLASS statement, but this is not necessary.

The CLASS statement is documented in the section "CLASS Statement" (Chapter 3, *SAS/STAT User's Guide: High-Performance Procedures*).

FREQ Statement

FREQ *variable* **;**

The *variable* in the FREQ statement identifies a numeric variable in the data set that contains the frequency of occurrence of each observation. PROC GAMPL treats each observation as if it appeared f times, where the frequency value f is the value of the FREQ variable for the observation. If f is not an integer, then f is truncated to an integer. If f is less than 1 or missing, the observation is not used in the analysis. When you do not specify the FREQ statement, each observation is assigned a frequency of 1.

ID Statement

ID *variables* **;**

The ID statement lists one or more variables from the input data set that are to be transferred to the output data set that you specify in the OUTPUT statement.

For more information, see the section "ID Statement" (Chapter 3, *SAS/STAT User's Guide: High-Performance Procedures*).

MODEL Statement

MODEL *response* < **(***response-options***)** > **=** < **PARAM(***effects***)** > < *spline-effects* > < */ model-options* > **;**

MODEL *events / trials* **=** < **PARAM(***effects***)** > < *spline-effects* > < */ model-options* > **;**

The MODEL statement specifies the response (dependent or target) variable and the predictor (independent or explanatory) effects of the model. You can specify the response in the form of a single variable or in the form of a ratio of two variables, which are denoted *events/trials*. The first form applies to all distribution families; the second form applies only to summarized binomial response data. When you have binomial data, the *events* variable contains the number of positive responses (or events) and the *trials* variable contains the number of trials. The values of both *events* and (*trials* – *events*) must be nonnegative, and the value of *trials* must be positive. If you specify a single *response* variable that is in a CLASS statement, then the response is assumed to be binary.

You can specify parametric effects that are constructed from variables in the input data set and include the effects in the parentheses of a PARAM( ) option, which can appear multiple times. For information about constructing the model effects, see the section "Specification and Parameterization of Model Effects" (Chapter 3, *SAS/STAT User's Guide: High-Performance Procedures*).

You can specify *spline-effects* by including independent variables inside the parentheses of the SPLINE( ) option. Only continuous variables (not classification variables) can be specified in *spline-effects*. Each *spline-effect* can have at least one variable and optionally some *spline-options*. You can specify any number of *spline-effects*. The following table shows some examples.

Spline Effect Specification	Meaning
`Spline(x)`	Constructs the univariate spline with x and uses the observed data points as knots. The maximum degrees of freedom is 10. PROC GAMPL uses an optimization algorithm to determine the optimal smoothing parameter.
`Spline(x1/knots=list(1 to 10))`	Constructs the univariate spline by using x1 and a supplied list of knots from 1 to 10. PROC GAMPL uses an optimization algorithm to determine the optimal smoothing parameter.
`Spline(x2 x3/smooth=0.3)`	Constructs the bivariate spline by using x2 and x3 and a fixed smoothing parameter 0.3.
`Spline(x4 x5 x6/maxdf=40)`	Constructs the trivariate spline by using x4, x5, and x6 and a maximum of 40 degrees of freedom. PROC GAMPL uses an optimization algorithm to determine the optimal smoothing parameter.

Both parametric effects and spline effects are optional. If none are specified, a model that contains only an intercept is fitted. If only parametric effects are present, PROC GAMPL fits a parametric generalized linear model by using the terms inside the parentheses of all PARAM( ) terms. If only spline effects are present, PROC GAMPL fits a nonparametric additive model. If both types of effects are present, PROC GAMPL fits a semiparametric model by using the parametric effects as the linear part of the model.

There are three sets of options in the MODEL statement. The *response-options* determine how the GAMPL procedure models probabilities for binary data. The *spline-options* controls how each spline term forms basis expansions. The *model-options* control other aspects of model formation and inference. Table 46.4 summarizes these options.

Table 46.4 MODEL Statement Options

Option	Description
Response Variable Options for Binary Models	
DESCENDING	Reverses the response categories
EVENT=	Specifies the event category
ORDER=	Specifies the sort order
REF=	Specifies the reference category
Smoothing Options for Spline Effects	
DETAILS	Requests detailed spline information
DF=	Specifies the fixed degrees of freedom
INITSMOOTH=	Specifies the starting value for the smoothing parameter
KNOTS=	Specifies the knots to be used for constructing the spline
M=	Specifies polynomial orders for constructing the spline
MAXDF=	Specifies the maximum degrees of freedom
MAXKNOTS=	Specifies the maximum number of knots to be used for constructing the spline
MAXSMOOTH=	Specifies the upper bound for the smoothing parameter

Table 46.4 *continued*

Option	Description
MINSMOOTH=	Specifies the lower bound for the smoothing parameter
SMOOTH=	Specifies a fixed smoothing parameter
Model Options	
ALLOBS	Requests all nonmissing values of spline variables for constructing spline basis functions regardless of other model variables
CRITERION=	Specifies the model evaluation criterion
DISPERSION | PHI=	Specifies the fixed dispersion parameter
DISTRIBUTION | DIST=	Specifies the response distribution
FDHESSIAN	Requests a finite-difference Hessian for smoothing parameter selection
INITIALPHI=	Specifies the starting value of the dispersion parameter
LINK=	Specifies the link function
NORMALIZE	Requests normalized spline basis functions for model fitting
MAXPHI=	Specifies the upper bound for searching the dispersion parameter
METHOD=	Specifies the algorithm for selecting smoothing parameters
MINPHI=	Specifies the lower bound for searching the dispersion parameter
OFFSET=	Specifies the offset variable
RIDGE=	Specifies the ridge parameter
SCALE=	Specifies the method for estimating the dispersion parameter

Response Variable Options

Response variable options determine how the GAMPL procedure models probabilities for binary data.

You can specify the following *response-options* by enclosing them in parentheses after the *response* variable.

DESCENDING

DESC

reverses the order of the response categories. If you specify both the DESCENDING and ORDER= options, PROC GAMPL orders the response categories according to the ORDER= option and then reverses that order.

EVENT=‘*category*’ | FIRST | LAST

specifies the event category for the binary response model. PROC GAMPL models the probability of the event category. The EVENT= option has no effect when there are more than two response categories.

You can specify any of the following:

‘*category*’

specifies that observations whose value matches *category* (formatted, if a format is applied) in quotation marks represent events in the data. For example, the following statements specify that observations that have a formatted value of ‘1’ represent events in the data. The probability that is modeled by the GAMPL procedure is thus the probability that the variable def takes on the (formatted) value ‘1’.

```
proc gampl data=MyData;
   class A B C;
   model def(event ='1') = param(A B C) spline(x1 x2 x3);
run;
```

FIRST
designates the first ordered category as the event.

LAST
designates the last ordered category as the event.

By default, EVENT=FIRST.

ORDER=DATA | FORMATTED | INTERNAL

ORDER=FREQ | FREQDATA | FREQFORMATTED | FREQINTERNAL

specifies the sort order for the levels of the *response* variable. When ORDER=FORMATTED (the default) for numeric variables for which you have supplied no explicit format (that is, for which there is no corresponding FORMAT statement in the current PROC GAMPL run or in the DATA step that created the data set), the levels are ordered by their internal (numeric) value. Table 46.5 shows the interpretation of the ORDER= option.

Table 46.5 Sort Order

ORDER=	Levels Sorted By
DATA	Order of appearance in the input data set
FORMATTED	External formatted value, except for numeric variables that have no explicit format, which are sorted by their unformatted (internal) value
FREQ	Descending frequency count (levels that have the most observations come first in the order)
FREQDATA	Order of descending frequency count; within counts by order of appearance in the input data set when counts are tied
FREQFORMATTED	Order of descending frequency count; within counts by formatted value when counts are tied
FREQINTERNAL	Order of descending frequency count; within counts by unformatted value when counts are tied
INTERNAL	Unformatted value

By default, ORDER=FORMATTED. For the FORMATTED and INTERNAL orders, the sort order is machine-dependent.

For more information about sort order, see the chapter about the SORT procedure in *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

REF=‘*category*’ **| FIRST | LAST**

specifies the reference category for the binary response model. Specifying one response category as the reference is the same as specifying the other response category as the event category. You can specify any of the following:

‘*category*’

specifies that observations whose value matches *category* (formatted, if a format is applied) are designated as the reference.

FIRST

designates the first ordered category as the reference

LAST

designates the last ordered category as the reference.

By default, REF=LAST.

Spline Options

DETAILS

requests a detailed spline specification information table.

DF=*number*

specifies a fixed degrees of freedom. When you specify this option, no smoothing parameter selection is performed on the spline term. If *number* is not an integer, then *number* is truncated to an integer.

INITSMOOTH=*number*

specifies the starting value for a smoothing parameter. The *number* must be nonnegative.

KNOTS=*method*

specifies the *method* for supplying user-defined knot values instead of using data values for constructing basis expansions. You can use the following *methods* for supplying the knots:

LIST(*list-of-values***)**

specifies a list of values as knots for the spline construction. For a multivariate spline term, the listed values are taken as multiple row vectors, where each vector has values that are ordered by specified variables. If the last row vector of knots contains fewer values than the number of variables, then the last row vector is ignored. For example, the following specification of a spline term produces two actual knot vectors ($\boldsymbol{k}_1$ and $\boldsymbol{k}_2$) and the value 5 is ignored.

```
spline(x1 x2/knots=list(1 2 3 4 5))
```

Table 46.6 Knot Values for a Bivariate Spline with a Supplied List

	x1	x2
$\boldsymbol{k}_1$	1	2
$\boldsymbol{k}_2$	3	4

EQUAL(*n*)

specifies the number of equally spaced interior knots for every variable in a spline term. Two boundary knots are automatically added to the knot list for each variable such that the total number of knots is $(n + 2)^d$, where d is the number of variables in the spline term. For a multivariate spline term, knot values for each variable are determined independently from the corresponding boundary values. For example, if the boundary points for x1 are 1 and 5 and the boundary points for x2 are 2 and 6, then the following specification of a spline term produces nine actual knots ($\boldsymbol{k}_1$ — $\boldsymbol{k}_9$), which consist of two boundary knots and one interior knot for each variable.

```
spline(x1 x2/knots=equal(1))
```

Table 46.7 Knot Values for a Bivariate Spline with One Interior Knot

	x1	x2
$\boldsymbol{k}_1$	1	2
$\boldsymbol{k}_2$	1	4
$\boldsymbol{k}_3$	1	6
$\boldsymbol{k}_4$	3	2
$\boldsymbol{k}_5$	3	4
$\boldsymbol{k}_6$	3	6
$\boldsymbol{k}_7$	5	2
$\boldsymbol{k}_8$	5	4
$\boldsymbol{k}_9$	5	6

M=*number*

specifies the order of the derivative in the penalty term. The *number* must be a positive integer. The default is $\max(2, \text{int}(d/2) + 1)$, where d is the number of variables in the spline term.

MAXDF=*number*

specifies the maximum *number* of degrees of freedom. When a thin-plate regression spline is formed, the specified *number* is used for constructing a low-rank penalty matrix to approximate the penalty matrix via the truncated eigendecomposition. The *number* must be greater than $\binom{m+d-1}{d}$, where m is the derivative order that is specified in the M= option. The default is $10 \times d$. For more information, see the section "Thin-Plate Regression Splines" on page 3301.

MAXKNOTS=*number*

specifies the maximum *number* of knots if data points are used to form knots. If KNOTS=LIST(*list-of-values*) is not specified, PROC GAMPL forms knots from unique data points. If the number of unique data points is greater than *number*, a subset of size *number* is formed by random sampling from all unique data points. The *number* cannot exceed the largest integer that can be stored on the given machine. By default, MAXKNOTS=2000.

MAXSMOOTH=*number*

specifies the upper bound for the smoothing parameter. The default is the largest double-precision value.

MINSMOOTH=*number*

specifies the lower bound for the smoothing parameter. By default, MINSMOOTH=0.

SMOOTH=*number*

specifies a fixed smoothing parameter. When you specify this option, no smoothing parameter selection is performed on the spline term.

Model Options

ALLOBS

requests that all nonmissing values of the variables be specified in a spline term for constructing the spline basis functions, regardless of whether other model variables are missing.

CRITERION=*criterion*

specifies the model evaluation criterion for selecting smoothing parameters for *spline-effects*. You can specify the following values:

GACV< (FACTOR=*number* **| GAMMA=***number***) >**

uses the generalized approximate cross validation (GACV) criterion to evaluate models.

GCV< (FACTOR=*number* **| GAMMA=***number***) >**

uses the generalized cross validation (GCV) criterion to evaluate models.

UBRE< (FACTOR=*number* **| GAMMA=***number***) >**

uses the unbiased risk estimator (UBRE) criterion to evaluate models.

The default *criterion* depends on the value of the DISTRIBUTION= option. For distributions that involve dispersion parameters, GCV is the default. For distributions without dispersion parameters, UBRE is the default. For all three criteria, you can optionally use the FACTOR= option to specify an extra tuning parameter in order to penalize more for model roughness. The value of *number* must be greater than or equal to 1. For more information about the model evaluation criteria, see "Model Evaluation Criteria" on page 3306.

DISPERSION=*number*

PHI=*number*

specifies a fixed dispersion parameter for distributions that have a dispersion parameter. The dispersion parameter that is used in all computations is fixed at *number*; it is not estimated.

DISTRIBUTION=*keyword*

specifies the response distribution for the model. The *keywords* and their associated distributions are shown in Table 46.8.

Table 46.8 Built-In Distribution Functions

DISTRIBUTION=	Distribution Function
BINARY	Binary
BINOMIAL	Binary or binomial
GAMMA	Gamma
INVERSEGAUSSIAN | IG	Inverse Gaussian
NEGATIVEBINOMIAL | NB	Negative binomial
NORMAL | GAUSSIAN	Normal
POISSON	Poisson
TWEEDIE< (*Tweedie-options***) >**	Tweedie

When DISTRIBUTION=TWEEDIE, you can specify the following *Tweedie-options*:

INITIALP=

specifies a starting value for iterative estimation of the Tweedie power parameter.

P=

requests a fixed Tweedie power parameter.

If you do not specify a link function in the LINK= option, a default link function is used. The default link function for each distribution is shown in Table 46.9. You can use any link function shown in Table 46.10 by specifying the LINK= option. Other commonly used link functions for each distribution are shown in Table 46.9.

Table 46.9 Default and Commonly Used Link Functions

DISTRIBUTION=	Default Link Function	Other Commonly Used Link Functions
BINARY	Logit	Probit, complementary log-log, log-log
BINOMIAL	Logit	Probit, complementary log-log, log-log
GAMMA	Reciprocal	Log
INVERSEGAUSSIAN | IG	Reciprocal square	Log
NEGATIVEBINOMIAL | NB	Log	
NORMAL | GAUSSIAN	Identity	Log
POISSON	Log	
TWEEDIE	Log	

FDHESSIAN

requests that the second-order derivatives (Hessian) be computed using finite-difference approximations based on evaluation of the first-order derivatives (gradient). This option might be useful if the analytical Hessian takes a long time to compute.

INITIALPHI=*number*
specifies a starting value for iterative maximum likelihood estimation of the dispersion parameter for distributions that have a dispersion parameter.

LINK=*keyword*
specifies the link function for the model. The *keywords* and the associated link functions are shown in Table 46.10. Default and commonly used link functions for the available distributions are shown in Table 46.9.

Table 46.10 Built-In Link Functions

LINK=	Link Function	$g(\mu) = \eta =$
CLOGLOG | CLL	Complementary log-log	$\log(-\log(1-\mu))$
IDENTITY | ID	Identity	μ
INV | RECIP	Reciprocal	$1/\mu$
INV2	Reciprocal square	$1/\mu^2$
LOG	Logarithm	$\log(\mu)$
LOGIT	Logit	$\log(\mu/(1-\mu))$
LOGLOG	Log-log	$-\log(-\log(\mu))$
PROBIT	Probit	$\Phi^{-1}(\mu)$

$\Phi^{-1}(\cdot)$ denotes the quantile function of the standard normal distribution.

MAXPHI=*number*
specifies an upper bound for maximum likelihood estimation of the dispersion parameter for distributions that have a dispersion parameter.

METHOD=OUTER | PERFORMANCE
specifies the algorithm for selecting smoothing parameters for *spline-effects*. You can specify the following values:

OUTER specifies the outer iteration method for selecting smoothing parameters. For more information about the method, see the section "Outer Iteration" on page 3307.

PERFORMANCE specifies the performance iteration method for selecting smoothing parameters. For more information about the method, see the section "Performance Iteration" on page 3308.

By default, METHOD=PERFORMANCE.

MINPHI=*number*
specifies a lower bound for maximum likelihood estimation of the dispersion parameter for distributions that have a dispersion parameter.

NORMALIZE
requests normalized spline basis functions for model fitting. After the regression spline basis functions are computed, each column is standardized to have a unit standard error. The corresponding penalty matrix is also scaled accordingly. This option might be helpful when you have badly scaled data.

OFFSET=*variable*

specifies a *variable* to be used as an offset to the linear predictor. An offset plays the role of an effect whose coefficient is known to be 1. The offset variable cannot appear in the CLASS statement or elsewhere in the MODEL statement. Observations that have missing values for the offset variable are excluded from the analysis.

RIDGE=*number*

allows a ridge parameter such that a diagonal matrix $\mathbf{H} : \mathbf{H}_{ii} = number$ is added to the optimization problem with respect to regression parameters:

$$\min(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \boldsymbol{\beta}'\mathbf{S}_{\boldsymbol{\lambda}}\boldsymbol{\beta} + \boldsymbol{\beta}'\mathbf{H}\boldsymbol{\beta} \quad \text{with respect to} \quad \boldsymbol{\beta}$$

By default, RIDGE=0. Specifying a small ridge parameter might be helpful if the model matrix $\mathbf{X}'\mathbf{X} + \mathbf{S}_{\boldsymbol{\lambda}}$ is close to singular.

SCALE=DEVIANCE | MLE | PEARSON

specifies the method for estimating the scale and dispersion parameters. You can specify the following values:

DEVIANCE

estimates the dispersion parameter by using the deviance statistic.

MLE

computes the dispersion parameter by maximizing the likelihood or penalized likelihood.

PEARSON

estimates the dispersion parameter by using Pearson's statistic.

By default, SCALE=MLE.

OUTPUT Statement

OUTPUT < **OUT=***SAS-data-set*>
< *keyword* < **=***name*> >. . . < *keyword* < **=***name*> > < / *options*> **;**

The OUTPUT statement creates a data set that contains observationwise statistics that are computed after the model is fitted. The variables in the input data set are *not* included in the output data set in order to avoid data duplication for large data sets; however, variables that are specified in the ID statement are included.

If the input data set is distributed (so that accessing data in a particular order cannot be guaranteed), the GAMPL procedure copies the distribution or partition key to the output data set so that its contents can be joined with the input data.

The computation of the output statistics is based on the final parameter estimates. If the model fit does not converge, missing values are produced for the quantities that depend on the estimates.

You can specify the following syntax elements in the OUTPUT statement before the slash (/).

OUT=*SAS-data-set*

DATA=*SAS-data-set*

specifies the name of the output data set. If you omit this option, PROC GAMPL uses the DATA*n* convention to name the output data set.

keyword **<** **=***name* **>**

specifies a statistic to include in the output data set and optionally assigns a *name* to the variable. If you do not provide a *name*, the GAMPL procedure assigns a default name based on the *keyword*.

You can specify the following *keywords* for adding statistics to the OUTPUT data set:

LINP | XBETA

requests the linear predictor $\eta = \mathbf{x}'\boldsymbol{\beta}$. For observations in which only the response variable is missing, values of the linear predictor are computed even though these observations do not affect the model fit. The default *name* is Xbeta.

LOWER

requests a lower Bayesian confidence band value for the predicted value. The default *name* is Lower.

PEARSON | PEARS | RESCHI

requests the Pearson residual, $(y - \mu)/\sqrt{\mathrm{V}(\mu)}$, where μ is the estimate of the predicted response mean and $\mathrm{V}(\mu)$ is the response distribution variance function. The default *name* is Pearson.

PREDICTED | PRED | P

requests predicted values for the response variable. For observations in which only the response variable is missing, the predicted values are computed even though these observations do not affect the model fit. The default *name* is Pred.

RESIDUAL | RESID | R

requests the raw residual, $y - \mu$, where μ is the estimate of the predicted mean. The default *name* is Residual.

STD

requests a standard error for the linear predictor. The default *name* is Std.

UPPER

requests an upper Bayesian confidence band value for the predicted value. The default *name* is Upper.

You can specify the following *options* in the OUTPUT statement after the slash (/):

ALPHA=*number*

specifies the significance level for the construction of Bayesian confidence bands in the OUTPUT data set. The confidence level is $1 -$ *number*.

COMPONENT

requests componentwise statistics for all spline terms if LINP, LOWER, STD, or UPPER is specified as a *keyword*.

PERFORMANCE Statement

PERFORMANCE < *performance-options* > **;**

You can use the PERFORMANCE statement to control whether the procedure executes in single-machine or distributed mode. The default is single-machine mode.

You can also use this statement to define performance parameters for multithreaded and distributed computing, and you can request details about performance results.

The PERFORMANCE statement is documented in the section “PERFORMANCE Statement” (Chapter 2, *SAS/STAT User’s Guide: High-Performance Procedures*).

WEIGHT Statement

WEIGHT *variable* **;**

The *variable* in the WEIGHT statement is used as a weight to perform a weighted analysis of the data. Observations that have nonpositive or missing weights are not included in the analysis. If a WEIGHT statement is not included, then all observations that are used in the analysis are assigned a weight of 1.

Details: GAMPL Procedure

Missing Values

Any observation that has missing values for the response, frequency, weight, offset, or explanatory variables is excluded from the analysis; however, missing values are valid for response and explanatory variables if you specify the MISSING option in the CLASS statement. Observations that have a nonpositive weight or a frequency less than 1 are also excluded. For Poisson, negative binomial and Tweedie distributions, observations that have a negative response value are excluded. For gamma and inverse Gaussian distributions, observations that have a nonpositive response value are excluded.

The estimated linear predictor and the fitted means are not computed for any observation that has missing offset or explanatory variable values. However, if only the response value is missing, the linear predictor and the fitted means can be computed and output to a data set by using the OUTPUT statement.

Thin-Plate Regression Splines

The GAMPL procedure uses thin-plate regression splines (Wood 2003) to construct spline basis expansions. The thin-plate regression splines are based on thin-plate smoothing splines (Duchon 1976, 1977). Compared to thin-plate smoothing splines, thin-plate regression splines produce fewer basis expansions and thus make direct fitting of generalized additive models possible.

Thin-Plate Smoothing Splines

Consider the problem of estimating a smoothing function f of $\mathbf{x}$ with d covariates from n observations. The model assumes

$$y_i = f(\mathbf{x}_i) + \epsilon_i, i = 1, \dots, n$$

Then the thin-plate smoothing splines estimate the smoothing function f by minimizing the penalized least squares function:

$$\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i))^2 + \lambda J_{m,d}(f)$$

The penalty term $\lambda J_{m,d}(f)$ includes the function that measures roughness on the f estimate:

$$J_{m,d}(f) = \int \cdots \int \sum_{\alpha_1+\cdots+\alpha_d=m} \frac{m!}{\alpha_1! \cdots \alpha_d!} \left(\frac{\partial^m f}{\partial {x_1}^{\alpha_1} \cdots \partial {x_d}^{\alpha_d}} \right)^2 \mathrm{d}x_1 \cdots \mathrm{d}x_d$$

The parameter m (which corresponds to the M= option for a spline effect) specifies how the penalty is applied to the function roughness. Function derivatives whose order is less than m are not penalized. The relation $2m > d$ must be satisfied.

The penalty term also includes the smoothing parameter $\lambda \in [0, \infty)$, which controls the trade-off between the model's fidelity to the data and the function smoothness of f. When $\lambda = 0$, the function estimate corresponds to an interpolation. When $\lambda \to \infty$, the function estimate becomes the least squares fit. By using the defined penalized least squares criterion and a fixed λ value, you can explicitly express the estimate of the smooth function f in the following form:

$$f_\lambda(\mathbf{x}) = \sum_{j=1}^{M} \theta_j \phi_j(\mathbf{x}) + \sum_{i}^{n} \delta_i \eta_{m,d}(\|\mathbf{x} - \mathbf{x}_i\|)$$

In the expression of $f_\lambda(\mathbf{x})$, δ_i and θ_j are coefficients to be estimated. The functions $\phi_j(\mathbf{x})$ correspond to unpenalized polynomials of $\mathbf{x}$ with degrees up to $m - 1$. The total number of these polynomials is $M = \binom{m+d-1}{d}$. The function $\eta_{m,d}$ models the extra nonlinearity besides the polynomials and is a function of the Euclidean distance r between any $\mathbf{x}$ value and an observed $\mathbf{x}_i$ value:

$$\eta_{m,d}(r) = \begin{cases} \frac{(-1)^{m+1+d/2}}{2^{2m-1}\pi^{d/2}(m-1)!(m-d/2)!} r^{2m-d} \log(r) & \text{if } d \text{ is even} \\ \frac{\Gamma(d/2-m)}{2^{2m}\pi^{d/2}(m-1)!} r^{2m-d} & \text{if } d \text{ is odd} \end{cases}$$

Define the penalty matrix $\mathbf{E}$ such that each entry $E_{ij} = \eta_{m,d}(\|\mathbf{x}_i - \mathbf{x}_j\|)$, let $\mathbf{y}$ be the vector of the response, let $\mathbf{T}$ be the matrix where each row is formed by $\phi_j(\mathbf{x})$, and let $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$ be vectors of coefficients θ_j and δ_i. Then you can obtain the function estimate f_λ from the following minimization problem:

$$\min \|\mathbf{y} - \mathbf{T}\boldsymbol{\theta} - \mathbf{E}\boldsymbol{\delta}\|^2 + \lambda \boldsymbol{\delta}'\mathbf{E}\boldsymbol{\delta} \quad \text{subject to} \quad \mathbf{T}'\boldsymbol{\delta} = \mathbf{0}$$

For more information about thin-plate smoothing splines, see Chapter 122, “The TPSPLINE Procedure.”

Low-Rank Approximation

Given the representation of the thin-plate smoothing spline, the estimate of f involves as many parameters as the number of unique data points. Solving $(\boldsymbol{\theta}, \boldsymbol{\delta})$ with an optimum λ becomes difficult for large problems.

Because the matrix $\mathbf{E}$ is symmetric and nonnegative definite, the eigendecomposition can be taken as $\mathbf{E} = \mathbf{UDU}'$, where $\mathbf{D}$ is the diagonal matrix of eigenvalues d_i of $\mathbf{E}$, and $\mathbf{U}$ is the matrix of eigenvectors that corresponds to d_i. The truncated eigendecomposition forms $\tilde{\mathbf{E}}_k$, which is an approximation to $\mathbf{E}$ such that

$$\tilde{\mathbf{E}}_k = \mathbf{U}_k\mathbf{D}_k\mathbf{U}_k'$$

where $\mathbf{D}_k$ is a diagonal matrix that contains the k most extreme eigenvalues in descending order of absolute values: $|\tilde{d}_1| > \cdots > |\tilde{d}_k|$. $\mathbf{U}_k$ is the matrix that is formed by columns of eigenvectors that correspond to the eigenvalues in $\mathbf{D}_k$.

The approximation $\tilde{\mathbf{E}}_k$ not only reduces the dimension from $n \times n$ of $\mathbf{E}$ to $n \times k$ but also is optimal in two senses. First, $\tilde{\mathbf{E}}_k$ minimizes the spectral norm $\|\mathbf{E} - \mathbf{F}_k\|_2$ between $\mathbf{E}$ and all rank k matrices $\mathbf{F}_k$. Second, $\tilde{\mathbf{E}}_k$ also minimizes the worst possible change that is introduced by the eigenspace truncation as defined by

$$\max_{\boldsymbol{\delta}\neq\mathbf{0}} \frac{\boldsymbol{\delta}'(\mathbf{E} - \mathbf{G}_k)\boldsymbol{\delta}}{\|\boldsymbol{\delta}\|^2}$$

where $\mathbf{G}_k$ is formed by any k eigenvalues and corresponding eigenvectors. For more information, see Wood (2003).

Now given $\mathbf{E} \approx \tilde{\mathbf{E}}_k$ and $\tilde{\mathbf{E}}_k = \mathbf{U}_k\mathbf{D}_k\mathbf{U}_k'$, and letting $\boldsymbol{\delta}_k = \mathbf{U}_k'\boldsymbol{\delta}$, the minimization problem becomes

$$\min \|\mathbf{y} - \mathbf{T}\boldsymbol{\theta} - \mathbf{U}_k\mathbf{D}_k\boldsymbol{\delta}_k\|^2 + \lambda\boldsymbol{\delta}_k'\mathbf{D}_k\boldsymbol{\delta}_k \quad \text{subject to} \quad \mathbf{T}'\mathbf{U}_k\boldsymbol{\delta}_k = \mathbf{0}$$

You can turn the constrained optimization problem into an unconstrained one by using any orthogonal column basis $\mathbf{Z}$. One way to form $\mathbf{Z}$ is via the QR decomposition of $\mathbf{U}_k'\mathbf{T}$:

$$\mathbf{U}_k'\mathbf{T} = [\mathbf{Q}_1\mathbf{Q}_2]\begin{bmatrix}\mathbf{R}\\ \mathbf{0}\end{bmatrix}$$

Let $\mathbf{Z} = \mathbf{Q}_2$. Then it is verified that

$$\mathbf{T}'\mathbf{U}_k\mathbf{Z} = \mathbf{R}'\mathbf{Q}_1'\mathbf{Q}_2 = \mathbf{0}$$

So for $\boldsymbol{\delta}_k$ such that $\mathbf{T}'\mathbf{U}_k\boldsymbol{\delta}_k = \mathbf{0}$, it is true that $\boldsymbol{\delta}_k = \mathbf{Z}\tilde{\boldsymbol{\delta}}$. Now the problem becomes the unconstrained optimization,

$$\min \|\mathbf{y} - \mathbf{T}\boldsymbol{\theta} - \mathbf{U}_k\mathbf{D}_k\mathbf{Z}\tilde{\boldsymbol{\delta}}\|^2 + \lambda\tilde{\boldsymbol{\delta}}'\mathbf{Z}'\mathbf{D}_k\mathbf{Z}\tilde{\boldsymbol{\delta}}$$

Let

$$\boldsymbol{\beta} = \begin{bmatrix}\boldsymbol{\theta}\\ \tilde{\boldsymbol{\delta}}\end{bmatrix}, \quad \mathbf{X} = [\mathbf{T} : \mathbf{U}_k\mathbf{D}_k\mathbf{Z}], \quad \text{and} \quad \mathbf{S} = \begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{Z}'\mathbf{D}_k\mathbf{Z}\end{bmatrix}$$

The optimization is simplified as

$$\min \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\boldsymbol{\beta}'\mathbf{S}\boldsymbol{\beta} \quad \text{with respect to} \quad \boldsymbol{\beta}$$

Generalized Additive Models

Generalized Linear Models

All probability distributions that the GAMPL procedure fits are members of an exponential family of distributions. For the specification of an exponential family, see the section "Generalized Linear Models Theory" in Chapter 48, "The GENMOD Procedure."

Table 46.11 lists and defines some common notation that is used in the context of generalized linear models and generalized additive models.

Table 46.11 Common Notation

Notation	Meaning
ℓ	Log-likelihood
ℓ_p	Penalized log-likelihood
D	Deviance
D_p	Penalized deviance
g	Link function
g^{-1}	Inverse link function
$\boldsymbol{\mu}$	Response mean $\boldsymbol{\mu} = g^{-1}(\boldsymbol{\eta})$
$\boldsymbol{\eta}$	Linear predictor $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta}$
ϕ	Dispersion parameter
$\mathbf{z}$	Column of adjusted response variable
$\boldsymbol{\nu}$	Column of response variance
ω_i	Prior weight for each observation
w_i	Adjusted weight for each observation
$\mathbf{W}$	Diagonal matrix of adjusted weights

The GAMPL procedure supports the following distributions: binary, binomial, gamma, inverse Gaussian, negative binomial, normal (Gaussian), Poisson and Tweedie.

For forms of log-likelihood functions for each of the probability distributions, see the section "Log-Likelihood Functions" in Chapter 48, "The GENMOD Procedure." For forms of the deviance for each of the probability distributions, see the section "Goodness of Fit" in Chapter 48, "The GENMOD Procedure."

Generalized Additive Models

Generalized additive models are extensions of generalized linear models (Nelder and Wedderburn 1972). For each observation that has a response Y_i and a row vector of the model matrix $\mathbf{x}_i$, both generalized additive models and generalized linear models assume the model additivity

$$g(\mu_i) = f_1(\mathbf{x}_{i1}) + \cdots + f_p(\mathbf{x}_{ip})$$

where $\mu_i = E(Y_i)$ and Y_i is independently distributed in some exponential family. Generalized linear models further assume model linearity by $f_j(\mathbf{x}_{ij}) = \mathbf{x}_{ij}\beta_j$ for $j = 1, \ldots, p$. Generalized additive models relax the linearity assumption by allowing some smoothing functions f_j to characterize the dependency. The GAMPL procedure constructs the smoothing functions by using thin-plate regression splines. For more information about generalized additive models and other type of smoothing functions, see Chapter 45, "The GAM Procedure."

Consider a generalized additive model that has some linear terms $\mathbf{X}_L$ with coefficients $\boldsymbol{\beta}_L$ and p smoothing functions f_j. Each smoothing function can be constructed by thin-plate regression splines with a smoothing parameter λ_j. Using the notations described in the section "Low-Rank Approximation" on page 3303, you can characterize each smoothing function by

$$\boldsymbol{\beta}_j = \begin{bmatrix} \boldsymbol{\theta}_j \\ \tilde{\boldsymbol{\delta}}_j \end{bmatrix}, \quad \mathbf{X}_j = [\mathbf{T}_j : \mathbf{U}_{kj}\mathbf{D}_{kj}\mathbf{Z}_j], \quad \text{and} \quad \mathbf{S}_j = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}'_j\mathbf{D}_{kj}\mathbf{Z}_j \end{bmatrix}$$

Notice that each smoothing function representation contains a zero-degree polynomial that corresponds to a constant. Having multiple constant terms makes the smoothing functions unidentifiable. The solution is to include a global constant term (that is, the intercept) in the model and enforce the centering constraint to each smoothing function. You can write the constraint as

$$\mathbf{1}'\mathbf{X}_j\boldsymbol{\beta}_j = 0$$

By using a similar approach as the linear constraint for thin-plate regression splines, you obtain the orthogonal column basis $\mathbf{V}_j$ via the QR decomposition of $\mathbf{X}'_j\mathbf{1}$ such that $\mathbf{1}'\mathbf{X}_j\mathbf{V}_j = 0$. Each smoothing function can be reparameterized as $\tilde{\mathbf{X}}_j = \mathbf{X}_j\mathbf{V}_j$.

Let $\mathbf{X} = [\mathbf{X}_L : \tilde{\mathbf{X}}_1 : \cdots : \tilde{\mathbf{X}}_p]$ and $\boldsymbol{\beta}' = [\boldsymbol{\beta}'_L : \boldsymbol{\beta}'_1 : \cdots : \boldsymbol{\beta}'_p]$. Then the generalized additive model can be represented as $g(\boldsymbol{\mu}) = \mathbf{X}\boldsymbol{\beta}$. The roughness penalty matrix is represented as a block diagonal matrix:

$$\mathbf{S}_{\boldsymbol{\lambda}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & & \cdots & & \mathbf{0} \\ \mathbf{0} & \lambda_1\mathbf{S}_1 & & \cdots & & \mathbf{0} \\ & & \ddots & & & \\ \vdots & \vdots & & \lambda_j\mathbf{S}_j & & \vdots \\ & & & & \ddots & \\ \mathbf{0} & \mathbf{0} & & \cdots & & \lambda_p\mathbf{S}_p \end{bmatrix}$$

Then the roughness penalty is measured in the quadratic form $\boldsymbol{\beta}'\mathbf{S}_{\boldsymbol{\lambda}}\boldsymbol{\beta}$.

Penalized Likelihood Estimation

Given a fixed vector of smoothing parameters, $\boldsymbol{\lambda} = [\lambda_i \ldots \lambda_p]'$, you can fit the generalized additive models by the penalized likelihood estimation. In contrast to the maximum likelihood estimation, penalized likelihood estimation obtains an estimate for $\boldsymbol{\beta}$ by maximizing the penalized log likelihood,

$$\ell_p(\boldsymbol{\beta}) = \ell(\boldsymbol{\beta}) - \frac{1}{2}\boldsymbol{\beta}'\mathbf{S}_{\boldsymbol{\lambda}}\boldsymbol{\beta}$$

Any optimization technique that you can use for maximum likelihood estimation can also be used for penalized likelihood estimation. If first-order derivatives are required for the optimization technique, you can compute the gradient as

$$\frac{\partial \ell_p}{\partial \boldsymbol{\beta}} = \frac{\partial \ell}{\partial \boldsymbol{\beta}} - \mathbf{S}_{\boldsymbol{\lambda}}\boldsymbol{\beta}$$

If second-order derivatives are required for the optimization technique, you can compute the Hessian as

$$\frac{\partial^2 \ell_p}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}'} = \frac{\partial^2 \ell}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}'} - \mathbf{S}_{\boldsymbol{\lambda}}$$

In the gradient and Hessian forms, $\partial \ell / \partial \boldsymbol{\beta}$ and $\partial^2 \ell / (\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}')$ are the corresponding gradient and Hessian, respectively, for the log-likelihood for generalized linear models. For more information about optimization techniques, see the section "Choosing an Optimization Technique" on page 3312.

Model Evaluation Criteria

Given a fixed set of smoothing parameters $\boldsymbol{\lambda}$ in which each λ_i controls the smoothness of each spline term, you can fit a generalized additive model by the penalized likelihood estimation. There are infinitely many sets of smoothing parameters. In order to search optimum models, some model evaluation criteria need to be defined to quantify the model goodness-of-fit. The GAMPL procedure uses the following model evaluation criteria:

- generalized cross validation (GCV), $\mathcal{V}_g$ (Craven and Wahba 1979)
- unbiased risk estimator (UBRE), $\mathcal{V}_u$ (Craven and Wahba 1979)
- generalized approximate cross validation (GACV), $\mathcal{V}_a$ (Xiang and Wahba 1996)

Consider the optimization problem

$$\min(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \boldsymbol{\beta}'\mathbf{S}_{\boldsymbol{\lambda}}\boldsymbol{\beta} \quad \text{with respect to} \quad \boldsymbol{\beta}$$

The parameter estimate for $\boldsymbol{\beta}$ can be represented as

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X} + \mathbf{S}_{\boldsymbol{\lambda}})^{-1}\mathbf{X}'\mathbf{y}$$

And the smoothing matrix (also called the influence matrix or hat matrix) is thus represented as

$$\mathbf{H}_{\boldsymbol{\lambda}} = \mathbf{X}(\mathbf{X}'\mathbf{X} + \mathbf{S}_{\boldsymbol{\lambda}})^{-1}\mathbf{X}'$$

With the defined smoothing matrix, you can form the model evaluation criteria as follows:

$$\begin{aligned}
\mathcal{V}_g(\boldsymbol{\lambda}) &= \frac{n\|\mathbf{y} - \mathbf{H}_{\boldsymbol{\lambda}}\mathbf{y}\|^2}{(\mathrm{tr}(\mathbf{I} - \gamma\mathbf{H}_{\boldsymbol{\lambda}}))^2} \\
\mathcal{V}_u(\boldsymbol{\lambda}) &= \frac{1}{n}\|\mathbf{y} - \mathbf{H}_{\boldsymbol{\lambda}}\mathbf{y}\|^2 - \frac{2}{n}\sigma^2\mathrm{tr}(\mathbf{I} - \gamma\mathbf{H}_{\boldsymbol{\lambda}}) + \sigma^2 \\
\mathcal{V}_a(\boldsymbol{\lambda}) &= \frac{1}{n}\|\mathbf{y} - \mathbf{H}_{\boldsymbol{\lambda}}\mathbf{y}\|^2 \left(1 + 2\gamma\frac{\mathrm{tr}(\mathbf{H}_{\boldsymbol{\lambda}})}{\mathrm{tr}(\mathbf{I} - \mathbf{H}_{\boldsymbol{\lambda}})}\right)
\end{aligned}$$

In the equations, $\gamma \geq 1$ (which corresponds to the GAMMA= suboption of the CRITERION= option) is the tuning parameter that is sometimes used to enforce smoother models.

The GAMPL procedure uses fitting algorithms that involve minimizing the model evaluation criterion with respect to unknown smoothing parameters $\boldsymbol{\lambda}$.

Fitting Algorithms

For models that assume a normally distributed response variable, you can minimize the model evaluation criteria directly by searching for optimal smoothing parameters. For models that have nonnormal distributions, you cannot use the model evaluation criteria directly because the involved statistics keep changing between iterations. The GAMPL procedure enables you to use either of two fitting approaches to search for optimum models: the outer iteration method and the performance iteration method. The outer iteration method modifies the model evaluation criteria so that a global objective function can be minimized in order to find the best smoothing parameters. The performance iteration method minimizes a series of objective functions in an iterative fashion and then obtains the optimum smoothing parameters at convergence. For large data sets, the performance iteration method usually converges faster than the outer iteration method.

Outer Iteration

The outer iteration method is outlined in Wood (2006). The method uses modified model evaluation criteria, which are defined as follows:

$$\begin{aligned}
\mathcal{V}_g^o(\boldsymbol{\lambda}) &= \frac{nD_{\boldsymbol{\lambda}}(\boldsymbol{\mu})}{(n-\gamma\mathrm{tr}(\mathbf{H}_{\boldsymbol{\lambda}}))^2} \\
\mathcal{V}_u^o(\boldsymbol{\lambda}) &= \frac{D_{\boldsymbol{\lambda}}(\boldsymbol{\mu})}{n} - \frac{2}{n}\sigma^2\mathrm{tr}(\mathbf{I}-\gamma\mathbf{H}_{\boldsymbol{\lambda}}) + \sigma^2 \\
\mathcal{V}_a^o(\boldsymbol{\lambda}) &= \frac{D_{\boldsymbol{\lambda}}(\boldsymbol{\mu})}{n} + \frac{2\gamma\mathrm{tr}(\mathbf{H}_{\boldsymbol{\lambda}})P_{\boldsymbol{\lambda}}}{n\mathrm{tr}(\mathbf{I}-\mathbf{H}_{\boldsymbol{\lambda}})}
\end{aligned}$$

By replacing $\|\mathbf{y}-\mathbf{H}_{\boldsymbol{\lambda}}\mathbf{y}\|^2$ with model deviance $D_{\boldsymbol{\lambda}}(\boldsymbol{\mu})$, the modified model evaluation criteria relate to the smoothing parameter $\boldsymbol{\lambda}$ in a direct way such that the analytic gradient and Hessian are available in explicit forms. The Pearson's statistic $P_{\boldsymbol{\lambda}}$ is used in the GACV criterion $\mathcal{V}_a^o(\boldsymbol{\lambda})$ (Wood 2008). The algorithm for the outer iteration is thus as follows:

1. Initialize smoothing parameters by taking one step of performance iteration based on adjusted response and adjusted weight except for spline terms with initial values that are specified in the INITSMOOTH= option.

2. Search for the best smoothing parameters by minimizing the modified model evaluation criteria. The optimization process stops when any of the convergence criteria that are specified in the SMOOTHOPTIONS option is met. At each optimization step:

   a) Initialize by setting initial regression parameters $\boldsymbol{\beta} = \{g(\bar{\mathbf{y}}), 0, \ldots, 0\}'$. Set the initial dispersion parameter if necessary.

   b) Search for the best regression parameters $\boldsymbol{\beta}$ by minimizing the penalized deviance D_p (or maximizing the penalized likelihood ℓ_p for negative binomial distribution). The optimization process stops when any of the convergence criteria that are specified in the PLIKEOPTIONS option is met.

   c) At convergence, evaluate derivatives of the model evaluation criteria with respect to $\boldsymbol{\lambda}$ by using $\partial D_p/\partial\boldsymbol{\beta}$, $\partial^2 D_p/(\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}')$, $\partial\boldsymbol{\beta}/\partial\lambda_j$, and $\partial^2\boldsymbol{\beta}/(\partial\lambda_j\partial\lambda_k)$.

Step 2b usually converges quickly because it is essentially penalized likelihood estimation given that $D_p = 2\phi(\ell_{\max} - \ell) + \boldsymbol{\beta}\mathbf{S}_{\boldsymbol{\lambda}}\boldsymbol{\beta}'$. Step 2c contains involved computation by using the chain rule of derivatives. For more information about computing derivatives of $\mathcal{V}_g^o$ and $\mathcal{V}_u^o$, see Wood (2008, 2011).

Performance Iteration

The performance iteration method is proposed by Gu and Wahba (1991). Wood (2004) modifies and stabilizes the algorithm for fitting generalized additive models. The algorithm for the performance iteration method is as follows:

1. Initialize smoothing parameters $\boldsymbol{\lambda} = \{1, \ldots, 1\}$, except for spline terms whose initial values are specified in the INITSMOOTH= option. Set the initial regression parameters $\boldsymbol{\beta} = \{g(\bar{y}), 0, \ldots, 0\}'$. Set the initial dispersion parameter if necessary.
2. Repeat the following steps until any of these three conditions is met: (1) the absolute function change in penalized likelihood is sufficiently small; (2) the absolute relative function change in penalized likelihood is sufficiently small; (3) the number of iterations exceeds the maximum iteration limit.
   a) Form the adjusted response and adjusted weight from $\boldsymbol{\mu} = g^{-1}(\boldsymbol{\eta})$. For each observation,

   $$z_i = \eta_i + (y_i - \mu_i)/\mu_i', \quad w_i = \omega_i \mu_i'^2 / \nu_i$$

   b) Search for the best smoothing parameters for the current iteration based on valid adjusted response values and adjusted weight values.
   c) Use the smoothing parameters to construct the linear predictor and the predicted means.
   d) Obtain an estimate for the dispersion parameter if necessary.

In step 2b, you can use different optimization techniques to search for the best smoothing parameters. The Newton-Raphson optimization is efficient in finding the optimum $\boldsymbol{\lambda}$ where the first- and second-order derivatives are available. For more information about computing derivatives of $\mathcal{V}_g$ and $\mathcal{V}_u$ with respect to $\boldsymbol{\lambda}$, see Wood (2004).

Degrees of Freedom

Let $\mathbf{W}$ be the adjusted weight matrix at convergence, and let $\mathbf{S}_{\boldsymbol{\lambda}}$ be the roughness penalty matrix with selected smoothing parameters. The degrees of freedom matrix is defined as in Wood (2006):

$$\mathbf{F} = (\mathbf{X}'\mathbf{W}\mathbf{X} + \mathbf{S}_{\boldsymbol{\lambda}})^{-1}\mathbf{X}'\mathbf{W}\mathbf{X}$$

Given the adjusted response $\mathbf{z}$, the parameter estimate is shown to be $\tilde{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\mathbf{z}$ for the model without penalization, and the parameter estimate is $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{W}\mathbf{X} + \mathbf{S}_{\boldsymbol{\lambda}})^{-1}\mathbf{X}'\mathbf{W}\mathbf{z} = \mathbf{F}\tilde{\boldsymbol{\beta}}$ with penalization. $\mathbf{F}$ is thus the matrix that projects or maps the unpenalized parameter estimates to the penalized ones.

The model degrees of freedom is given as

$$\mathrm{df} = \mathrm{tr}(\mathbf{F})$$

And the degrees of freedom for error is given as

$$\mathrm{df}_r = n - 2\mathrm{df} + \mathrm{tr}(\mathbf{FF})$$

For the *j*th spline term, the degrees of freedom for the component is defined to be the trace of the submatrix of **F** that corresponds to parameter estimates $\boldsymbol{\beta}_j$:

$$\mathrm{df}_j = \mathrm{tr}(\mathbf{F}_j)$$

The degrees of freedom for the smoothing component test of the *j*th term is defined similarly as

$$\mathrm{df}^t_j = 2\mathrm{df}_j - \mathrm{tr}((\mathbf{FF})_j)$$

Model Inference

Wahba (1983) proposes a Bayesian covariance matrix for parameter estimates $\boldsymbol{\beta}$ by interpreting a smoothing spline as a posterior mean. Nychka (1988) shows that the derived Bayesian posterior confidence limits work well from frequentist viewpoints. The Bayesian posterior covariance matrix for the parameters is

$$\mathbf{V}_{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{WX} + \mathbf{S}_{\boldsymbol{\lambda}})^{-1}\sigma^2$$

The posterior distribution for $\boldsymbol{\beta}$ is thus

$$\boldsymbol{\beta}|\mathbf{y} \sim N(\hat{\boldsymbol{\beta}}, \mathbf{V}_{\boldsymbol{\beta}})$$

For a particular point whose design row is vector **x**, the prediction is $\mathbf{x}\hat{\boldsymbol{\beta}}$ and the standard error is $\sqrt{\mathbf{x}\mathbf{V}_{\boldsymbol{\beta}}\mathbf{x}'}$. The Bayesian posterior confidence limits are thus

$$\left(\mathbf{x}\hat{\boldsymbol{\beta}} \pm z_{\alpha/2}\sqrt{\mathbf{x}\mathbf{V}_{\boldsymbol{\beta}}\mathbf{x}'}\right)$$

where $z_{\alpha/2}$ is the $1-\alpha/2$ quantile of the standard normal distribution.

For the *j*th spline term, the prediction for the component is formed by $\mathbf{x}_j\hat{\boldsymbol{\beta}}$, where $\mathbf{x}_j$ is a row vector of zeros except for columns that correspond to basis expansions of the *j*th spline term. And the standard error for the component is $\sqrt{\mathbf{x}_j\mathbf{V}_{\boldsymbol{\beta}}\mathbf{x}'_j}$.

Dispersion Parameter

Some distribution families (Gaussian, gamma, inverse Gaussian, negative binomial, and Tweedie) have a dispersion parameter that you can specify in the DISPERSION= option in the MODEL statement or you can estimate from the data. The following three suboptions for the SCALE= option in the MODEL statement correspond to three ways to estimate the dispersion parameter:

DEVIANCE

estimates the dispersion parameter by the deviance, given the regression parameter estimates:

$$\hat{\phi} = \frac{\sum_i D_i(y_i, \mu_i)}{n - \mathrm{df}}$$

MLE

estimates the dispersion parameter by maximizing the penalized likelihood, given the regression parameter estimates:

$$\hat{\phi} = \underset{\phi}{\text{argmax}}\ \ell_p(\hat{\boldsymbol{\beta}}, \phi)$$

The MLE option is the only option that you can use to estimate the dispersion parameter for the negative binomial and Tweedie distributions.

PEARSON

estimates the dispersion parameter by Pearson's statistic, given the regression parameter estimates:

$$\hat{\phi} = \frac{\sum_i \omega_i (y_i - \mu_i)^2 / \nu_i}{n - \text{df}}$$

If the dispersion parameter is estimated, it contributes one additional degree of freedom to the fitted model.

Tests for Smoothing Components

The GAMPL procedure performs a smoothing component test on the null hypotheses $f_j = 0$ for the jth component. In contrast to the analysis of deviance that is used in PROC GAM (which tests existence of nonlinearity for each smoothing component), the smoothing component test used in PROC GAMPL tests for the existence of a contribution for each smoothing component.

The hypothesis test is based on the Wald statistic. Define $\mathbf{X}_j$ as the matrix of all zeros except for columns that correspond to basis expansions of the jth spline term. Then the column vector of predictions is $\hat{\mathbf{f}}_j = \mathbf{X}_j \hat{\boldsymbol{\beta}}$, and the covariance matrix for the predictions is $\mathbf{V}_j = \mathbf{X}_j \mathbf{V}_{\boldsymbol{\beta}} \mathbf{X}_j'$. The Wald statistic for testing is

$$T_r = \hat{\mathbf{f}}_j' \mathbf{V}_j^{r-} \hat{\mathbf{f}}_j = \hat{\boldsymbol{\beta}}' \mathbf{X}_j' (\mathbf{X}_j \mathbf{V}_{\boldsymbol{\beta}} \mathbf{X}_j')^{r-} \mathbf{X}_j \hat{\boldsymbol{\beta}}$$

where $\mathbf{V}_j^{r-}$ is the rank-r pseudo-inverse of $\mathbf{V}_j$. If $\mathbf{R}_j$ is the Cholesky root for $\mathbf{X}_j'\mathbf{X}_j$ such that $\mathbf{R}_j'\mathbf{R} = \mathbf{X}_j'\mathbf{X}_j$, then the test statistic can be written as

$$T_r = \hat{\boldsymbol{\beta}}' \mathbf{R}_j' (\mathbf{R}_j \mathbf{V}_{\boldsymbol{\beta}} \mathbf{R}_j')^{r-} \mathbf{R}_j \hat{\boldsymbol{\beta}}$$

Wood (2012) proposes using the df_j^t degrees of freedom for test (which is defined in the section "Degrees of Freedom" on page 3308) as the rank r. Because spline terms in fitted models often have noninteger degrees of freedom, the GAMPL procedure uses a rounded value of df_j^t as the rank:

$$r = \begin{cases} \lfloor \text{df}_j^t \rfloor & \text{if } \text{df}_j^t - \lfloor \text{df}_j^t \rfloor <= 0.05 \text{ or } \text{df}_j^t < 1 \\ \lceil \text{df}_j^t \rceil & \text{otherwise} \end{cases}$$

Let K be a symmetric and nonnegative definite matrix, and let its eigenvalues be sorted as $d_1 > d_2 > \cdots$; then the rank-r pseudo-inverse of K is formed by

$$K^{r-} = U_k \begin{bmatrix} d_1^{-1} & & \\ & \ddots & \\ & & d_r^{-1} \end{bmatrix} U_k'$$

where U_k are formed by columns of eigenvectors that correspond to the r eigenvalues.

Under the null hypothesis, the Wald statistic T_r approximately follows the chi-square distribution $T_r \sim \chi_r^2$. For an observed test statistic t_r, the p-value for rejecting the null hypothesis is computed as $P(\chi_r^2 > t_r)$ if the dispersion parameter is constant, or $P(F_{r,\mathrm{df}_r} > t_r)$ with df_r error degrees of freedom if the dispersion parameter is estimated.

Be cautious when you interpret the results of the smoothing component test because p-values are computed by approximation and the test does not take the smoothing parameter selection process into account.

Computational Method: Multithreading

Threading is the organization of computational work into multiple tasks (processing units that can be scheduled by the operating system). Each task is associated with a thread. Multithreading is the concurrent execution of threads. When multithreading is possible, you can realize substantial performance gains compared to the performance you get from sequential (single-threaded) execution.

The number of threads that the GAMPL procedure spawns is determined by the number of CPUs on a machine and can be controlled in the following ways:

- You can specify the number of CPUs in the CPUCOUNT= SAS system option. For example, if you specify the following statement, the GAMPL procedure determines threading as if it were executing on a system that had four CPUs, regardless of the actual CPU count:

  ```
  options cpucount=4;
  ```

- You can specify the NTHREADS= option in the PERFORMANCE statement to control the number of threads. This specification overrides the CPUCOUNT= system option. Specify NTHREADS=1 to force single-threaded execution.

The GAMPL procedure allocates one thread per CPU by default.

The tasks that are multithreaded by the GAMPL procedure are primarily defined by dividing the data that are processed on a single machine among the threads—that is, the GAMPL procedure implements multithreading through a data-parallel model. For example, if the input data set has 1,000 observations and PROC GAMPL is running with four threads, then 250 observations are associated with each thread. All operations that require access to the data are then multithreaded. These operations include the following:

- variable levelization
- effect levelization
- formation of the initial crossproducts matrix
- truncated eigendecomposition
- formation of spline basis expansions
- objective function calculation
- gradient calculation

- Hessian calculation
- scoring of observations
- computing predictions for smoothing component plots

In addition, operations on matrices such as sweeps can be multithreaded if the matrices are of sufficient size to realize performance benefits from managing multiple threads for the particular matrix operation.

Choosing an Optimization Technique

First- or Second-Order Techniques

The factors that affect how you choose an optimization technique for a particular problem are complex. Although the default technique works well for most problems, you might occasionally benefit from trying several different techniques.

For many optimization problems, computing the gradient takes more computer time than computing the function value. Computing the Hessian sometimes takes *much* more computer time and memory than computing the gradient, especially when there are many decision variables. Unfortunately, optimization techniques that do not use some kind of Hessian approximation usually require many more iterations than techniques that use a Hessian matrix; as a result, the total run time of these techniques is often longer. Techniques that do not use the Hessian also tend to be less reliable. For example, they can terminate more easily at stationary points than at global optima.

Table 46.12 shows which derivatives are required for each optimization technique.

Table 46.12 Derivatives Required

Technique	First-Order	Second-Order
TRUREG	x	x
NEWRAP	x	x
NRRIDG	x	x
QUANEW	x	-
DBLDOG	x	-
CONGRA	x	-
NMSIMP	-	-

The second-order derivative techniques (TRUREG, NEWRAP, and NRRIDG) are best for small problems for which the Hessian matrix is not expensive to compute. Sometimes the NRRIDG technique can be faster than the TRUREG technique, but TRUREG can be more stable. The NRRIDG technique requires only one matrix with $p(p+1)/2$ double words, where p denotes the number of parameters in the optimization. TRUREG and NEWRAP require two such matrices.

The QUANEW and DBLDOG first-order derivative techniques are best for medium-sized problems for which the objective function and the gradient can be evaluated much faster than the Hessian. In general, the QUANEW and DBLDOG techniques require more iterations than TRUREG, NRRIDG, and NEWRAP, but each iteration can be much faster. The QUANEW and DBLDOG techniques require only the gradient to update an approximate Hessian, and they require slightly less memory than TRUREG or NEWRAP.

The CONGRA first-order derivative technique is best for large problems for which the objective function and the gradient can be computed much faster than the Hessian and for which too much memory is required to store the (approximate) Hessian. In general, the CONGRA technique requires more iterations than QUANEW or DBLDOG, but each iteration can be much faster. Because CONGRA requires only a factor of p double-word memory, many large applications can be solved only by CONGRA.

The no-derivative technique NMSIMP is best for small problems for which derivatives are not continuous or are very difficult to compute.

Each optimization technique uses one or more convergence criteria that determine when it has converged. A technique is considered to have converged when any one of the convergence criteria is satisfied. For example, under the default settings, the QUANEW technique converges if the value of the ABSGCONV= option is less than 1E–5, the value of the FCONV= option is less than $2 \times \epsilon$, or the value of the GCONV= option is less than 1E–8.

By default, the GAMPL procedure applies the NEWRAP technique to optimization for selecting smoothing parameters by using the performance iteration method. For the outer iteration method, the GAMPL procedure applies the QUANEW technique for selecting smoothing parameters.

Technique Descriptions

The following subsections provide details about each optimization technique and follow the same order as Table 46.12.

Trust Region Optimization (TRUREG)

The trust region technique (TRUREG) uses the gradient $\mathbf{g}(\boldsymbol{\psi}^{(k)})$ and the Hessian matrix $\mathbf{H}(\boldsymbol{\psi}^{(k)})$; thus, it requires that the objective function $f(\boldsymbol{\psi})$ have continuous first- and second-order derivatives inside the feasible region.

The trust region technique iteratively optimizes a quadratic approximation to the nonlinear objective function within a hyperelliptic trust region that has radius Δ and constrains the step size that corresponds to the quality of the quadratic approximation. The trust region technique that PROC GAMPL uses is based on Dennis, Gay, and Welsch (1981); Gay (1983); Moré and Sorensen (1983).

The trust region technique performs well for small- to medium-sized problems, and it does not need many function, gradient, and Hessian calls. However, if the computation of the Hessian matrix is computationally expensive, either the QUANEW technique or the CONGRA technique might be more efficient.

Newton-Raphson Optimization with Line Search (NEWRAP)

The Newton-Raphson optimization with line search (NEWRAP) technique uses the gradient $\mathbf{g}(\boldsymbol{\psi}^{(k)})$ and the Hessian matrix $\mathbf{H}(\boldsymbol{\psi}^{(k)})$; thus, it requires that the objective function have continuous first- and second-order derivatives inside the feasible region.

If second-order derivatives are computed efficiently and precisely, the NEWRAP technique can perform well for medium-sized to large problems, and it does not need many function, gradient, and Hessian calls.

This technique uses a pure Newton step when the Hessian is positive-definite and when the Newton step reduces the value of the objective function successfully. Otherwise, a combination of ridging and line search is performed to compute successful steps. If the Hessian is not positive-definite, a multiple of the identity matrix is added to the Hessian matrix to make it positive-definite (Eskow and Schnabel 1991).

In each iteration, a line search is performed along the search direction to find an approximate optimum of the objective function. The line-search technique uses quadratic interpolation and cubic extrapolation.

Newton-Raphson Ridge Optimization (NRRIDG)

The Newton-Raphson ridge optimization (NRRIDG) technique uses the gradient $\mathbf{g}(\boldsymbol{\psi}^{(k)})$ and the Hessian matrix $\mathbf{H}(\boldsymbol{\psi}^{(k)})$; thus, it requires that the objective function have continuous first- and second-order derivatives inside the feasible region.

This technique uses a pure Newton step when the Hessian is positive-definite and when the Newton step reduces the value of the objective function successfully. If at least one of these two conditions is not satisfied, a multiple of the identity matrix is added to the Hessian matrix.

Because the NRRIDG technique uses an orthogonal decomposition of the approximate Hessian, each iteration of NRRIDG can be slower than an iteration of the NEWRAP technique, which works with a Cholesky decomposition. However, NRRIDG usually requires fewer iterations than NEWRAP.

The NRRIDG technique performs well for small- to medium-sized problems, and it does not require many function, gradient, and Hessian calls. However, if the computation of the Hessian matrix is computationally expensive, either the QUANEW technique or the CONGRA technique might be more efficient.

Dual Quasi-Newton Optimization (QUANEW)

The dual quasi-Newton (QUANEW) technique uses the gradient $\mathbf{g}(\boldsymbol{\psi}^{(k)})$, and it does not need to compute second-order derivatives, because they are approximated. It works well for medium-sized to moderately large optimization problems, where the objective function and the gradient can be computed much faster than the Hessian. However, the QUANEW technique requires more iterations in general than the TRUREG, NEWRAP, and NRRIDG techniques, which compute second-order derivatives. The QUANEW technique provides an appropriate balance between the speed and stability that are required for most generalized linear model applications.

The QUANEW technique that PROC GAMPL uses is the dual quasi-Newton technique, which updates the Cholesky factor of an approximate Hessian.

In each iteration, a line search is performed along the search direction to find an approximate optimum. The line-search technique uses quadratic interpolation and cubic extrapolation to obtain a step size α that satisfies the Goldstein conditions (Fletcher 1987). One of the Goldstein conditions can be violated if the feasible region defines an upper limit of the step size. Violating the left-side Goldstein condition can affect the positive-definiteness of the quasi-Newton update. In that case, either the update is skipped or the iterations are restarted by using an identity matrix, resulting in the steepest descent or ascent search direction.

Double-Dogleg Optimization (DBLDOG)

The double-dogleg optimization (DBLDOG) technique combines the ideas of the quasi-Newton and trust region techniques. In each iteration, the double-dogleg technique computes the step $\mathrm{s}^{(k)}$ as the linear combination of the steepest descent or ascent search direction $\mathrm{s}_1^{(k)}$ and a quasi-Newton search direction $\mathrm{s}_2^{(k)}$:

$$\mathrm{s}^{(k)} = \alpha_1 \mathrm{s}_1^{(k)} + \alpha_2 \mathrm{s}_2^{(k)}$$

The step must remain within a prespecified trust region radius (Fletcher 1987, p. 107). Thus, the DBLDOG subroutine uses the dual quasi-Newton update but does not perform a line search.

The double-dogleg optimization technique works well for medium-sized to moderately large optimization problems, where the objective function and the gradient can be computed much faster than the Hessian. The implementation is based on Dennis and Mei (1979) and Gay (1983), but it is extended for dealing with boundary and linear constraints. The DBLDOG technique generally requires more iterations than the TRUREG, NEWRAP, and NRRIDG techniques, which require second-order derivatives; however, each of the DBLDOG iterations is computationally cheap. Furthermore, the DBLDOG technique requires only gradient calls for updating of the Cholesky factor of an approximate Hessian.

Conjugate Gradient Optimization (CONGRA)

Second-order derivatives are not required by the CONGRA technique and are not even approximated. The CONGRA technique can be expensive in function and gradient calls, but it requires only $O(p)$ memory for unconstrained optimization. In general, the technique must perform many iterations to obtain a precise solution, but each of the CONGRA iterations is computationally cheap.

The CONGRA technique should be used for optimization problems that have large p. For the unconstrained or boundary-constrained case, the CONGRA technique requires only $O(p)$ bytes of working memory, whereas all other optimization techniques require order $O(p^2)$ bytes of working memory. During p successive iterations that are uninterrupted by restarts or changes in the working set, the CONGRA technique computes a cycle of p conjugate search directions. In each iteration, a line search is performed along the search direction to find an approximate optimum of the objective function. The line-search technique uses quadratic interpolation and cubic extrapolation to obtain a step size α that satisfies the Goldstein conditions. One of the Goldstein conditions can be violated if the feasible region defines an upper limit for the step size.

Nelder-Mead Simplex Optimization (NMSIMP)

The Nelder-Mead simplex technique does not use any derivatives and does not assume that the objective function has continuous derivatives. The objective function itself needs to be continuous. This technique is quite expensive in the number of function calls, and it might be unable to generate precise results for $p \gg 40$.

The original Nelder-Mead simplex technique is implemented and extended to boundary constraints. This technique does not compute the objective for infeasible points, but it changes the shape of the simplex and adapts to the nonlinearities of the objective function. This change contributes to an increased speed of convergence and uses a special termination criterion.

Displayed Output

The following sections describe the output that the GAMPL procedure produces by default. The output is organized into various tables, which are discussed in the order of their appearance.

Performance Information

The "Performance Information" table is produced by default. It displays information about the execution mode. For single-machine mode, the table displays the number of threads used. For distributed mode, the table displays the grid mode (symmetric or asymmetric), the number of compute nodes, and the number of threads per node.

If you specify the DETAILS option in the PERFORMANCE statement, PROC GAMPL also produces a "Timing" table, which displays elapsed times (absolute and relative) for the main tasks of the procedure.

Data Access Information

The "Data Access Information" table is produced by default. For the input and output data sets, it displays the libref and data set name, the engine that was used to access the data, the role (input or output) of the data set, and the path that the data followed to reach the computation.

Model Information

The "Model Information" table displays basic information about the model, such as the response variable, frequency variable, link function, and model category that the GAMPL procedure determines based on your input and options. The "Model Information" table also displays the distribution of the data that PROC GAMPL assumes. For information about the supported response distributions, see the section "DISTRIBUTION=*keyword*" on page 3296.

Number of Observations

The "Number of Observations" table displays the number of observations that are read from the input data set and the number of observations that are used in the analysis. If a FREQ statement is present, the sum of the frequencies read and used is displayed. If the events/trials syntax is used, the number of events and trials is also displayed.

Response Profile

The "Response Profile" table displays the ordered values from which the GAMPL procedure determines the probability that is modeled as an event in binary models. For each response category level, the frequency that is used in the analysis is reported. You can affect the ordering of the response values by specifying *response-options* in the MODEL statement. For binary models, the note that follows the "Response Profile" table indicates which outcome is modeled as the event in binary models and which value serves as the reference category.

The "Response Profile" table is not produced for binomial (events/trials) data. You can find information about the number of events and trials in the "Number of Observations" table.

Class Level Information

The "Class Level Information" table lists the levels of every variable that is specified in the CLASS statement. You should check this information to make sure that the data are correct. You can adjust the order of the CLASS variable levels by specifying the ORDER= option in the CLASS statement. You can suppress the "Class Level Information" table completely or partially by specifying the NOCLPRINT= option in the PROC GAMPL statement.

If the classification variables use reference parameterization, the "Class Level Information" table also displays the reference value for each variable.

Specifications for Spline(*spline-variables*)

The "Specifications for Spline(*spline-variables*)" table displays basic information (such as number of variables, specified degrees of freedom, search range for the smoothing parameter, and so on) about how to construct a spline term that the GAMPL procedure uses to construct basis expansions and search for the smoothing parameter. PROC GAMPL generates the "Specifications for Spline(*spline-variables*)" table only when you specify the DETAILS option for a spline term.

Optimization Iteration History

For each iteration of the optimization, the "Iteration History" table displays the number of function evaluations (including gradient and Hessian evaluations), the value of the objective function, the change in the objective function from the previous iteration, and the absolute value of the largest (projected) gradient element. The objective function that PROC GAMPL uses in the optimization is normalized by default to enable comparisons across data sets that have different sampling intensity.

If you specify the ITDETAILS option in the PROC GAMPL statement, information about the parameter estimates and gradients in the course of the optimization is added to the "Iteration History" table.

For a parametric generalized linear model or for a generalized additive model that has fixed smoothing parameters, the "Iteration History" table displays information about regression parameter estimates and gradients. For a generalized additive model that has unknown smoothing parameters, the "Iteration History" table displays information about smoothing parameter estimates and gradients. If the performance iteration method is used, a column of performance iteration steps is added to the table.

Convergence Status

The convergence status table is a small ODS table that follows the "Iteration History" table in the default output. In the listing, this table appears as a message that indicates whether the optimization succeeded and which convergence criterion was met. If the optimization fails, the message indicates the reason for the failure. If you save the convergence status table to an output data set, a numeric Status variable is added that enables you to programmatically assess convergence. The values of the Status variable encode the following:

0	Convergence was achieved, or an optimization was not performed because TECHNIQUE=NONE was specified.
1	The objective function could not be improved.
2	Convergence was not achieved because of a user interrupt or because a limit (such as the maximum number of iterations or the maximum number of function evaluations) was reached. To modify these limits, see the MAXITER=, MAXFUNC=, and MAXTIME= options in the PROC GAMPL statement.
3	Optimization failed to converge because function or derivative evaluations failed at the starting values or during the iterations or because a feasible point that satisfies the parameter constraints could not be found in the parameter space.

Fit Statistics

The "Fit Statistics" table displays a variety of likelihood-based measures of fit in addition to the model roughness measurement. All information criteria are presented in "smaller is better" form.

The calculation of the information criteria uses the following formulas, where df denotes the model degrees of freedom, f denotes the number of frequencies used, and ℓ is the log likelihood that is evaluated at the converged estimates:

$$\mathrm{AIC} = -2\ell + 2\mathrm{df}$$

$$\mathrm{AICC} = \begin{cases} -2\ell + 2\mathrm{df}\, f/(f - \mathrm{df} - 1) & \text{when } f > \mathrm{df} + 2 \\ -2\ell + 2\mathrm{df}(\mathrm{df} + 2) & \text{otherwise} \end{cases}$$

$$\mathrm{BIC} = -2\ell + \mathrm{df}\log(f)$$

If no FREQ statement is specified, then f equals n, the number of observations that are used.

Parameter Estimates

The “Parameter Estimates” table displays the regression parameter estimates, their estimated (asymptotic) standard errors, chi-square statistics, and p-values for the hypothesis that the parameter is 0.

Estimates for Smoothing Components

The “Estimates for Smoothing Components” table displays a summary of the fitted spline terms, including effective degrees of freedom, smoothing parameters, roughness penalty values, number of parameters, rank of penalty matrix, and number of knots.

Tests for Smoothing Components

The “Tests for Smoothing Components” table displays effective degrees of freedom, effective degrees of freedom for test, F/χ_2 statistics, and p-values for rejecting the hypothesis that the smoothing component has zero contribution.

ODS Table Names

Each table that the GAMPL procedure creates has a name that is associated with it, and you must use this name to refer to the table when you use ODS statements. These names are listed in Table 46.13.

Table 46.13 ODS Tables Produced by PROC GAMPL

Table Name	Description	Statement	Option
ClassLevels	Level information from the CLASS statement	CLASS	
ConvergenceStatus	Status of optimization at conclusion of optimization	Default output	
DataAccessInfo	Information about modes of data access	Default output	
FitStatistics	Fit statistics	Default output	
LikelihoodHist	Iteration history for maximum likelihood estimation or penalized likelihood estimation	PROC GAMPL	ITDETAILS
ModelInfo	Information about the modeling environment	Default output	
NObs	Number of observations read and used, and number of events and trials, if applicable	Default output	
ParameterEstimates	Solutions for the parameter estimates that are associated with effects in MODEL statements	Default output	
PerformanceInfo	Information about the high-performance computing environment	Default output	

Table 46.13 *continued*

Table Name	Description	Statement	Option
ResponseProfile	Response categories and the category that is modeled in models for binary and multinomial data	Default output	
SmoothingEstimates	Information for spline terms after model fitting	Default output	
SmoothingHist	Iteration history for smoothing parameter estimation	PROC GAMPL	ITDETAILS
SmoothingTests	Smoothing components test result	Default output	
SplineDetails	Information about spline construction and smoothing parameter search	MODEL	SPLINE(*variables* / DETAILS)
Timing	Absolute and relative times for tasks performed by the procedure	PERFORMANCE	DETAILS

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.

ODS Graphics

Statistical procedures use ODS Graphics to create graphs as part of their output. ODS Graphics is described in detail in Chapter 21, "Statistical Graphics Using ODS."

Before you create graphs, ODS Graphics must be enabled (for example, by specifying the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" on page 623 in Chapter 21, "Statistical Graphics Using ODS."

The overall appearance of graphs is controlled by ODS styles. Styles and other aspects of using ODS Graphics are discussed in the section "A Primer on ODS Statistical Graphics" on page 622 in Chapter 21, "Statistical Graphics Using ODS."

When ODS Graphics is enabled, the GAMPL procedure by default produces plots of the partial predictions for each spline term in the model. Use the PLOTS option in the PROC GAMPL statement to control aspects of these plots.

PROC GAMPL assigns a name to each graph that it creates by using ODS. You can use these names to refer to the graphs when using ODS. The names are listed in Table 46.14.

Table 46.14 Graphs Produced by PROC GAMPL

ODS Graph Name	Plot Description	PLOTS= Option
SmoothingComponentPanel	Panel of multiple prediction curves	Default COMPONENTS

SmoothingComponentPlot	Unpacked prediction curves	PLOTS(UNPACK)
		PLOTS=COMPONENTS(UNPACK)

By default, prediction plots for each spline component are displayed in panels that contain at most six plots. If you specify more than six smoothing components, multiple panels are used. Use the PLOTS(UNPACK) option in the PROC GAMPL statement to display these plots individually.

Examples: GAMPL Procedure

Example 46.1: Scatter Plot Smoothing

This example shows how you can use PROC GAMPL to perform scatter plot smoothing.

The example uses the LIDAR data set (Ruppert, Wand, and Carroll 2003). This data set is used in many books and journals to illustrate different smoothing techniques. Scientists use a technique known as LIDAR (light detection and ranging), which uses laser reflections to detect chemical compounds in the atmosphere. The following DATA step creates the data set Lidar:

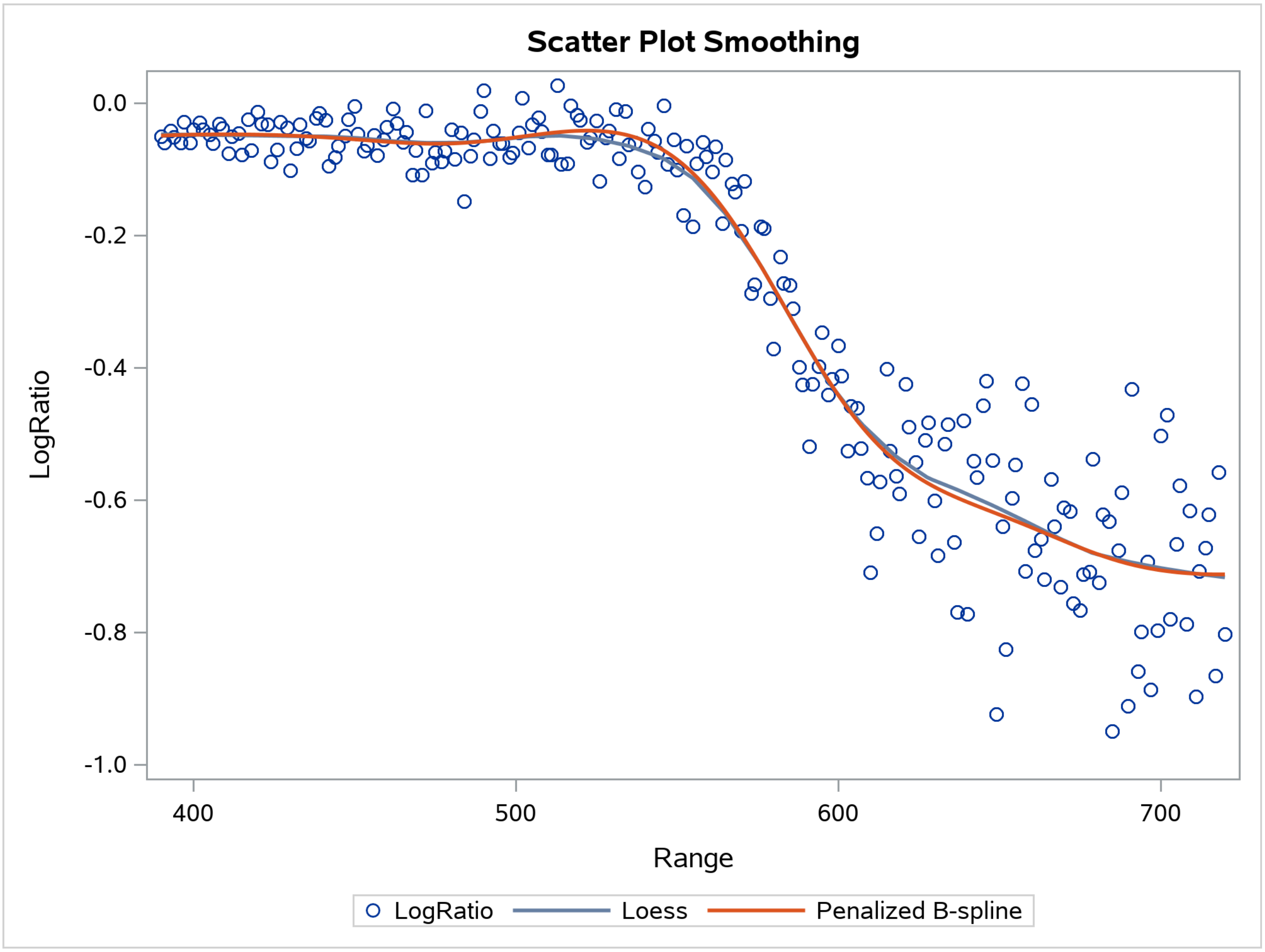
```
title 'Scatter Plot Smoothing';
data Lidar;
   input Range LogRatio @@;
   datalines;
390  -0.05035573    391  -0.06009706    393  -0.04190091    394   -0.0509847
396  -0.05991345    397  -0.02842392    399  -0.05958421    400  -0.03988881
402  -0.02939582    403  -0.03949445    405  -0.04764749    406     -0.06038
408  -0.03123034    409  -0.03816584    411  -0.07562269    412  -0.05001751
414   -0.0457295    415  -0.07766966    417  -0.02460641    418  -0.07133184

   ... more lines ...

702   -0.4716702    703   -0.7801088    705   -0.6668431    706   -0.5783479
708   -0.7874522    709   -0.6156956    711   -0.8967602    712   -0.7077379
714    -0.672567    715   -0.6218413    717   -0.8657611    718    -0.557754
720   -0.8026684
;
```

In this data set, Range records the distance that light travels before it is reflected back to the source. LogRatio is the logarithm of the ratio of light that is received from two laser sources. The objective is to use scatter plot smoothing to discover the nonlinear pattern in the data set. SAS provides different methods (for example local regression) for scatter plot smoothing. You can perform scatter plot smoothing by using the SGPLOT procedure, as shown in the following statements:

```
proc sgplot data=Lidar;
   scatter  x=Range y=LogRatio;
   loess    x=Range y=LogRatio / nomarkers;
   pbspline x=Range y=LogRatio / nomarkers;
run;
```

Output 46.1.1 shows the scatter plot of Range and LogRatio and the smoothing curves that are fitted by local regression and penalized B-splines smoothing techniques.

Output 46.1.1 Scatter Plot Smoothing



Both scatter plot smoothing techniques show a significant nonlinear structure between Range and LogRatio that cannot be easily modeled by ordinary polynomials. You can also use the GAMPL procedure to perform scatter plot smoothing on this data set, as in the following statements:

```
proc gampl data=Lidar seed=12345;
   model LogRatio = spline(Range/details);
   output out=LidarOut pred=p;
run;
```

The "Specifications for Spline(Range)" table in Output 46.1.2 displays the specifications for constructing the spline term for Range. The maximum degrees of freedom is 10, which sets the upper limit of effective degrees of freedom for the spline term to be 9 after one degree of freedom is absorbed in the intercept. The order of the derivative in the penalty is 2, which means the unpenalized portion of the spline term involves polynomials with degrees up to 2.

Output 46.1.2 Spline Specification

Scatter Plot Smoothing

The GAMPL Procedure

Specifications for Spline(Range)	
Number of Variables	1
Rank of Penalty Approximation	10
Order of Derivative in the Penalty	2
Maximum Number of Knots	2000

The "Fit Statistics" table in Output 46.1.3 shows the summary statistics for the fitted model.

Output 46.1.3 Fit Statistics

Fit Statistics	
Penalized Log Likelihood	251.44124
Roughness Penalty	0.00000426
Effective Degrees of Freedom	9.99988
Effective Degrees of Freedom for Error	211.00000
AIC (smaller is better)	-482.88271
AICC (smaller is better)	-481.83512
BIC (smaller is better)	-448.90148
GCV (smaller is better)	0.00654

The "Estimates for Smoothing Components" table in Output 46.1.4 shows that the effective degrees of freedom for the spline term of Range is approximately 8 after the GCV criterion is optimized with respect to the smoothing parameter. The roughness penalty is small, suggesting that there is an important contribution from the penalized part of thin-plate regression splines beyond nonpenalized polynomials.

Output 46.1.4 Estimates for Smoothing Components

Estimates for Smoothing Components

Component	Effective DF	Smoothing Parameter	Roughness Penalty	Number of Parameters	Rank of Penalty Matrix	Number of Knots
Spline(Range)	7.99988	1.0000	4.263E-6	9	10	221

Because the optimal model is obtained by searching in a functional space that is constrained by the maximum degrees of freedom for a spline term, you might wonder whether PROC GAMPL produces a much different model if you increase the value. The following statements fit another model in which the maximum degrees of freedom is increased to 20:

```
proc gampl data=Lidar seed=12345;
   model LogRatio = spline(Range/maxdf=20);
   output out=LidarOut2 pred=p2;
run;
```

Output 46.1.5 displays fit summary statistics for the second model. The model fit statistics from the second model are very close to the ones from the first model, indicating that the second model is not much different from the first model.

Output 46.1.5 Fit Statistics

Scatter Plot Smoothing

The GAMPL Procedure

Fit Statistics	
Penalized Log Likelihood	250.95136
Roughness Penalty	0.06254
Effective Degrees of Freedom	10.04384
Effective Degrees of Freedom for Error	209.03211
AIC (smaller is better)	-481.87757
AICC (smaller is better)	-480.82094
BIC (smaller is better)	-447.74696
GCV (smaller is better)	0.00657

Output 46.1.6 shows that the effective degrees of freedom for the spline term of Range is slightly larger than 8, which is understandable because increasing the maximum degrees of freedom expands the functional space for model searching. Functions in the expanded space can provide a better fit to the data, but they are also penalized more because the roughness penalty value for the second model is much larger than the one for the first model. This suggests that functions in the expanded space do not help much, given the nonlinear relationship between Range and LogRatio.

Output 46.1.6 Estimates for Smoothing Components

Estimates for Smoothing Components

Component	Effective DF	Smoothing Parameter	Roughness Penalty	Number of Parameters	Rank of Penalty Matrix	Number of Knots
Spline(Range)	8.04384	23134.6	0.0625	19	20	221

The two fitted models are all based on thin-plate regression splines, in which polynomials that have degrees higher than 2 are penalized. You might wonder whether allowing higher-order polynomials yields a much different model. The following statements fit a third spline model by penalizing polynomials that have degrees higher than 3:

```
proc gampl data=Lidar seed=12345;
   model LogRatio = spline(Range/m=3);
   output out=LidarOut3 pred=p3;
run;
```

The fit summary statistics shown in Output 46.1.7 are close to the ones from the previous two models, albeit slightly smaller.

Output 46.1.7 Fit Statistics

Scatter Plot Smoothing

The GAMPL Procedure

Fit Statistics	
Penalized Log Likelihood	249.79779
Roughness Penalty	9.440383E-9
Effective Degrees of Freedom	10.00000
Effective Degrees of Freedom for Error	211.00000
AIC (smaller is better)	-479.59559
AICC (smaller is better)	-478.54797
BIC (smaller is better)	-445.61397
GCV (smaller is better)	0.00664

As shown in Output 46.1.8, the effective degrees of freedom for the spline term where polynomials with degrees less than 4 are allowed without penalization is 8. The roughness penalty is quite small compared to the previous two fits. This also suggests that there are important contributions from the penalized part of the thin-plate regression splines even after the nonpenalized polynomials are raised to order 3.

Output 46.1.8 Estimates for Smoothing Components

Estimates for Smoothing Components

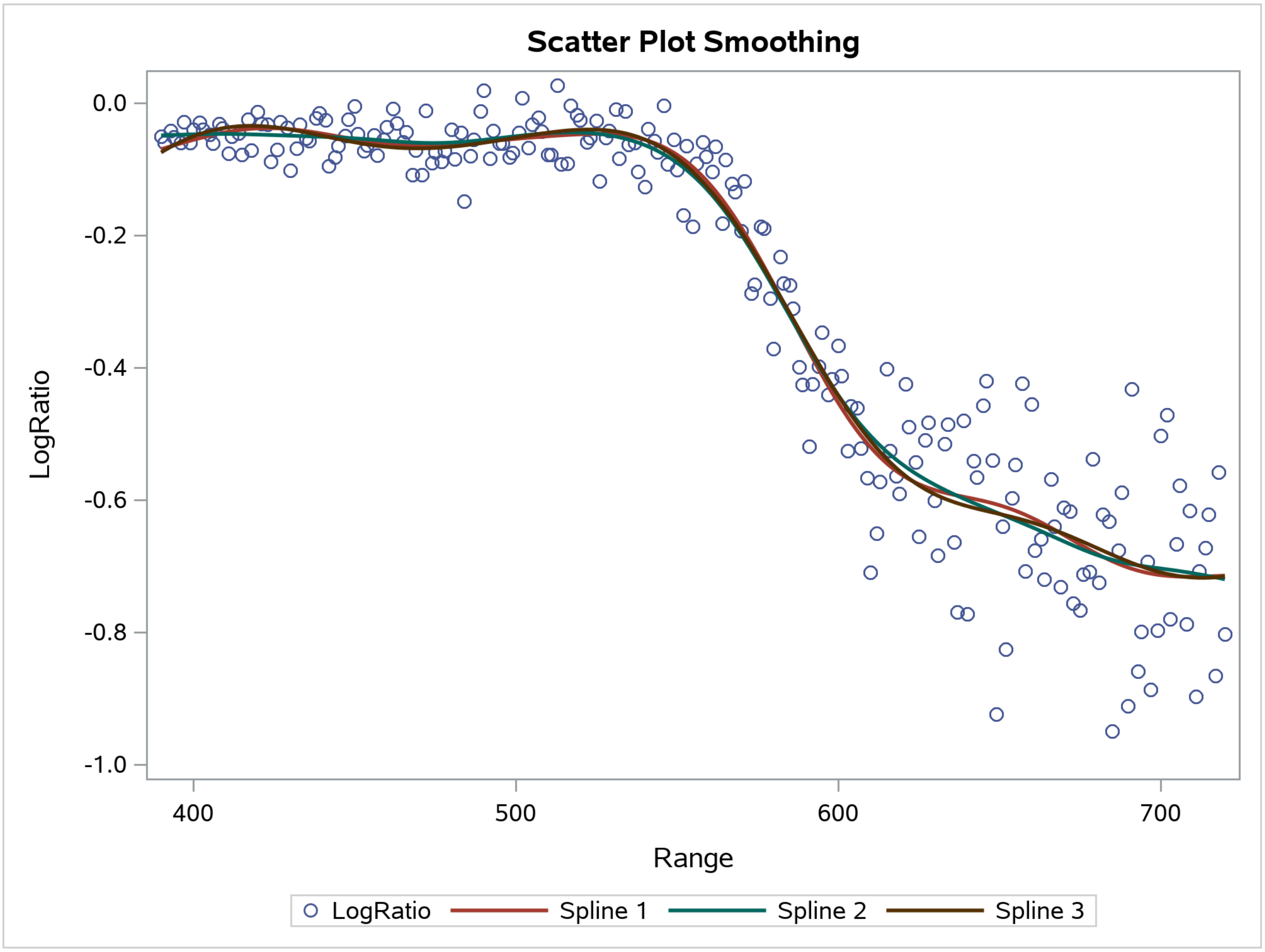
Component	Effective DF	Smoothing Parameter	Roughness Penalty	Number of Parameters	Rank of Penalty Matrix	Number of Knots
Spline(Range)	8.00000	1.0000	9.44E-9	9	10	221

The following statements use the DATA step to merge the predictions from the three scatter plot smoothing fits by PROC GAMPL and use the SGPLOT procedure to visualize them:

```
data LidarPred;
   merge Lidar LidarOut LidarOut2 LidarOut3;
run;

proc sgplot data=LidarPred;
   scatter x=Range y=LogRatio / markerattrs=GraphData1(size=7);
   series  x=Range y=p        / lineattrs  =GraphData2(thickness=2)
                                legendlabel="Spline 1";
   series  x=Range y=p2       / lineattrs  =GraphData3(thickness=2)
                                legendlabel="Spline 2";
   series  x=Range y=p3       / lineattrs  =GraphData4(thickness=2)
                                legendlabel="Spline 3";
run;
```

Output 46.1.9 displays the scatter plot smoothing fits by PROC GAMPL under three different spline specifications.

Output 46.1.9 Scatter Plot Smoothing



Example 46.2: Nonparametric Logistic Regression

This example shows how you can use PROC GAMPL to build a nonparametric logistic regression model for a diabetes data set that contains a binary response and then use that model to classify observations.

Diabetes is a major American disease. The American Diabetes Association estimates that over 8% of Americans have diabetes, and diabetes costs Americans over $175 billion a year. The National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK) has been studying diabetes and obesity in the Akimel O'otham (formerly known as Pima Indians) in Arizona for over 30 years. Smith et al. (1988) prepared some of the NIDDK data for forecasting the onset of diabetes and then donated the data for community use. In this particular example, data are for female patients who are at least 21 years old. The objective of this study is to investigate the relationship between a diabetes diagnosis and variables that represent physiological measurements and medical attributes. Some missing or invalid observations are removed from the analysis. The reduced data set contains 532 records. The following DATA step creates the data set DiabetesStudy:

```
title 'Diabetes Study';
data DiabetesStudy;
   input NPreg Glucose Pressure Triceps BMI Pedigree Age Diabetes Test@@;
   datalines;
 6  148  72  35  33.6  0.627  50  1  1    1   85  66  29  26.6  0.351  31  0  1
 1   89  66  23  28.1  0.167  21  0  0    3   78  50  32    31  0.248  26  1  0
 2  197  70  45  30.5  0.158  53  1  0    5  166  72  19  25.8  0.587  51  1  1
 0  118  84  47  45.8  0.551  31  1  1    1  103  30  38  43.3  0.183  33  0  1
 3  126  88  41  39.3  0.704  27  0  0    9  119  80  35    29  0.263  29  1  0

   ... more lines ...

 1  128  48  45  40.5  0.613  24  1  0    2  112  68  22  34.1  0.315  26  0  0
 1  140  74  26  24.1  0.828  23  0  1    2  141  58  34  25.4  0.699  24  0  1
 7  129  68  49  38.5  0.439  43  1  1    0  106  70  37  39.4  0.605  22  0  0
 1  118  58  36  33.3  0.261  23  0  1    8  155  62  26    34  0.543  46  1  0
;
```

The data set contains nine variables, including the binary response variable Diabetes. Table 46.15 describes the variables.

Table 46.15 Variables in the DiabetesStudy Data Set

Variable	Description
NPreg	Number of pregnancies
Glucose	Two-hour plasma glucose concentration in an oral glucose tolerance test
Pressure	Diastolic blood pressure (mm Hg)
Triceps	Triceps skin fold thickness (mm)
BMI	Body mass index (weight in kg/(height in m)^2)
Pedigree	Diabetes pedigree function
Age	Age (years)
Diabetes	0 if test negative for diabetes, 1 if test positive
Test	0 for training role, 1 for test

The Test variable splits the data set into training and test subsets. The training observations (whose Test value is 0) hold approximately 59.4% of the data. To build a model that is based on the training data and evaluate its performance by predicting the test data, you use the following statements to create a new variable, Result, whose value is the same as that of the Diabetes variable for a training observation and is missing for a test observation:

```
data DiabetesStudy;
   set DiabetesStudy;
   Result = Diabetes;
   if Test=1 then Result=.;
run;
```

As a starting point of your analysis, you can build a parametric logistic regression model on the training data and predict the test data. The following statements use PROC HPLOGISTIC to perform the analysis:

```
proc hplogistic data=DiabetesStudy;
   model Diabetes(event='1') = NPreg Glucose Pressure Triceps
                               BMI Pedigree Age;
   partition role=Test(test='1' train='0');
run;
```

Output 46.2.1 shows the summary statistics from the parametric logistic regression model.

Output 46.2.1 Fit Statistics

Diabetes Study

The HPLOGISTIC Procedure

Fit Statistics		
Description	**Training**	**Testing**
-2 Log Likelihood	295.79	176.90
AIC (smaller is better)	311.79	192.90
AICC (smaller is better)	312.24	193.65
BIC (smaller is better)	342.19	219.37

Output 46.2.2 shows fit statistics for both training and test subsets of the data, including the misclassification error for the test data.

Output 46.2.2 Partition Fit Statistics

Partition Fit Statistics		
Statistic	**Training**	**Testing**
Area under the ROCC	0.8607	0.8547
Average Square Error	0.1452	0.1401
Hosmer-Lemeshow Test	0.9726	0.1487
Misclassification Error	0.2061	0.2178
R-Square	0.3438	0.2557
Max-rescaled R-Square	0.4695	0.3706
McFadden's R-Square	0.3197	0.2522
Mean Difference	0.3780	0.3605
Somers' D	0.7213	0.7093
True Negative Fraction	0.8702	0.8435
True Positive Fraction	0.6639	0.6182

The parametric logistic regression model is restricted in the sense that all variables affect the response in strictly linear fashion. If you are uncertain that a variable is an important factor and its contribution is linear in predicting the response, you might want to choose a nonparametric logistic regression model to fit the data. You can use PROC GAMPL to form a nonparametric model by including the spline transformation of each explanatory variable, as shown in the following statements:

```
proc gampl data=DiabetesStudy seed=12345;
   model Result(event='1') = spline(NPreg)    spline(Glucose)
                             spline(Pressure) spline(Triceps)
                             spline(BMI)      spline(Pedigree)
                             spline(Age) / dist=binary;
run;
```

Because the response variable Result is binary, the DIST=BINARY option in the MODEL statement specifies a binary distribution for the response variable. By default, PROC GAMPL models the probability of the first ordered response category, which is a negative diabetes testing result in this case. The EVENT= option specifically requests that PROC GAMPL model the probability of positive diabetes testing results. The "Response Profile" table in Output 46.2.3 shows the frequencies of the response in both categories.

Output 46.2.3 Response Profile

Diabetes Study

The GAMPL Procedure

Response Profile

Ordered Value	Result	Total Frequency
1	0	208
2	1	122

You are modeling the probability that Result='1'.

Output 46.2.4 lists the summary statistics from the nonparametric logistic regression model, which include spline transformations of all variables.

Output 46.2.4 Fit Statistics

Fit Statistics	
Penalized Log Likelihood	-130.55095
Roughness Penalty	3.20011
Effective Degrees of Freedom	25.85896
Effective Degrees of Freedom for Error	302.97752
AIC (smaller is better)	309.61970
AICC (smaller is better)	314.20203
BIC (smaller is better)	407.86031
UBRE (smaller is better)	0.14225

The "Tests for Smoothing Components" table in Output 46.2.5 shows approximate tests results. Although some spline terms are significant, others are not. The null testing hypothesis is whether the total contribution from a variable is 0. So you can form a reduced model by removing those nonsignificant spline terms from the model. In this case, spline transformations for NPreg, Pressure, BMI, and Triceps are dropped from the model because their *p*-values are larger than the 0.1 nominal level.

Output 46.2.5 Tests for Smoothing Components

Tests for Smoothing Components

Component	Effective DF	Effective DF for Test	Chi-Square	Pr > ChiSq
Spline(NPreg)	1.00001	1	0.0758	0.7831
Spline(Glucose)	1.00000	1	38.4662	<.0001
Spline(Pressure)	7.80438	8	10.2605	0.2472
Spline(Triceps)	1.00000	1	0.6343	0.4258
Spline(BMI)	8	8	11.8047	0.1601
Spline(Pedigree)	1.00002	1	8.5323	0.0035
Spline(Age)	5.05456	7	15.7671	0.0273

The following statements use PROC GAMPL to fit a reduced nonparametric logistic regression model. The OUTPUT statement requests predicted probabilities for both training and test data sets. The ID statement requests that the Diabetes and Test variables also be included in the output data set so that you can use them to identify test observations and compute misclassification errors.

```
ods graphics on;
proc gampl data=DiabetesStudy plots seed=12345;
   model Result(event='1') = spline(Glucose)
                             spline(Pedigree) spline(Age) / dist=binary;
   output out=DiabetesStudyOut;
   id Diabetes Test;
run;
```

Output 46.2.6 shows the summary statistics from the reduced nonparametric logistic regression model. The values of the information criteria are better than of the parametric logistic regression model.

Output 46.2.6 Fit Statistics

Diabetes Study

The GAMPL Procedure

Fit Statistics	
Penalized Log Likelihood	-149.85765
Roughness Penalty	2.85613
Effective Degrees of Freedom	8.05242
Effective Degrees of Freedom for Error	320.61181
AIC (smaller is better)	312.96402
AICC (smaller is better)	313.41826
BIC (smaller is better)	343.55593
UBRE (smaller is better)	-0.00230

In the “Estimates for Smoothing Components” table in Output 46.2.7, PROC GAMPL reports that the effective degrees of freedom value for spline transformations of Glucose is quite close to 1. This suggests strictly linear form for Glucose. For Pedigree, the degrees of freedom value demonstrates a moderate amount of nonlinearity. For Age, the degrees of freedom value is much larger than 1. The measure suggests a nonlinear pattern in the dependency of the response on Age.

Output 46.2.7 Estimates for Smoothing Components

Estimates for Smoothing Components

Component	Effective DF	Smoothing Parameter	Roughness Penalty	Number of Parameters	Rank of Penalty Matrix	Number of Knots
Spline(Glucose)	1.00000	9.032E10	2.711E-7	9	10	110
Spline(Pedigree)	1.51071	0.4383	0.5086	9	10	283
Spline(Age)	4.54171	69.8810	2.3475	9	10	42

The “Tests for Smoothing Components” table in Output 46.2.8 shows that all spline transformations are significant in predicting diabetes testing results.

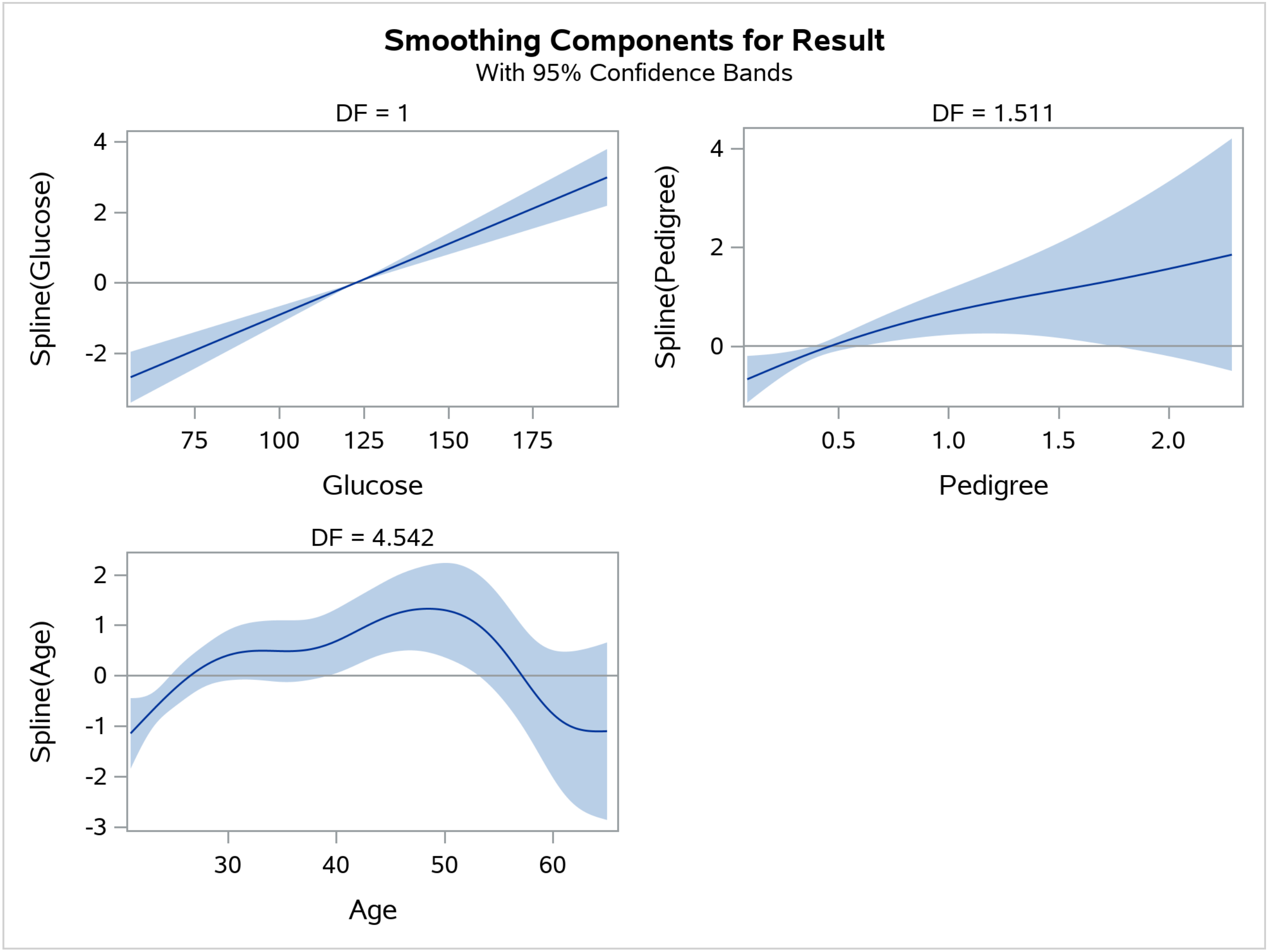
Output 46.2.8 Tests for Smoothing Components

Tests for Smoothing Components

Component	Effective DF	Effective DF for Test	Chi-Square	Pr > ChiSq
Spline(Glucose)	1.00000	1	53.0363	<.0001
Spline(Pedigree)	1.51071	2	9.9354	0.0070
Spline(Age)	4.54171	6	23.0661	0.0008

The smoothing component panel (which is produced by the PLOTS option and is shown in Output 46.2.9) visualizes the spline transformations for the four variables in addition to 95% Bayesian curvewise confidence bands. For Glucose, the spline transformation is almost a straight line. For Pedigree, the spline transformation shows a slightly nonlinear trend. For Age, the dependency is obviously nonlinear.

Output 46.2.9 Smoothing Components Panel



The following statements compute the misclassification error on the test data set from the reduced nonparametric logistic regression model that PROC GAMPL produces:

```
data test;
   set DiabetesStudyOut(where=(Test=1));
   if ((Pred>0.5 & Diabetes=1) | (Pred<0.5 & Diabetes=0))
   then Error=0;
   else Error=1;
run;

proc freq data=test;
   tables Diabetes*Error/nocol norow;
run;
```

Output 46.2.10 shows the misclassification errors for observations in the test set and observations of each response category. The error is smaller than the error from the parametric logistic regression model.

Output 46.2.10 Crosstabulation Table for Test Set Prediction

Diabetes Study

The FREQ Procedure

Frequency Percent	Table of Diabetes by Error		
	Error		
Diabetes	**0**	**1**	**Total**
0	130 64.36	17 8.42	147 72.77
1	35 17.33	20 9.90	55 27.23
Total	165 81.68	37 18.32	202 100.00

Example 46.3: Semiparametric Negative Binomial Model for Mackerel Egg Density

This example demonstrates how you can use PROC GAMPL to fit a semiparametric negative binomial model for a count data set that has overdispersions.

The example concerns a study of mackerel egg density. The data are a subset of the 1992 mackerel egg survey that was conducted over the Porcupine Bank west of Ireland. The survey took place in the peak spawning area. Scientists took samples by hauling a net up from deep sea to the sea surface. Then they counted the number of spawned mackerel eggs and used other geographic information to estimate the sizes and distributions of spawning stocks. The data set is used as an example in Bowman and Azzalini (1997).

The following DATA step creates the data set Mackerel. This data set contains 634 observations and five variables. The response variable Egg_Count is the number of mackerel eggs that are collected from each sampling net. Longitude and Latitude are the location values in degrees east and north, respectively, of each sample station. Net_Area is the area of the sampling net in square meters. Depth records the sea bed depth in meters at the sampling location, and Distance is the distance in geographic degrees from the sample location to the continental shelf edge.

```
title 'Mackerel Egg Density Study';
data Mackerel;
   input Egg_Count Longitude Latitude  Net_Area Depth Distance;
   datalines;
   0     -4.65     44.57     0.242    4342    0.8395141177
   0     -4.48     44.57     0.242    4334    0.8591926336
   0      -4.3     44.57     0.242    4286    0.8930152895
   1     -2.87     44.02     0.242    1438    0.3956408691
   4     -2.07     44.02     0.242     166    0.0400088237
   3     -2.13     44.02     0.242     460    0.0974234463
   0     -2.27     44.02     0.242     810    0.2362566569

   ... more lines ...

  22     -4.22     46.25      0.19     205    0.1181120828
  21     -4.28     46.25      0.19     237     0.129990854
   0     -4.73     46.25      0.19    2500    0.3346500536
   5     -4.25     47.23      0.19     114     0.718192582
   3     -3.72     47.25      0.19     100    0.9944669778
   0     -3.25     47.25      0.19      64    1.2639918431
;
```

The response values are counts, so the Poisson distribution might be a reasonable model. The statistic of interest is the mackerel egg density, which can be formed as

$$\text{Density} = E(\text{Count})/\text{Net\_Area}$$

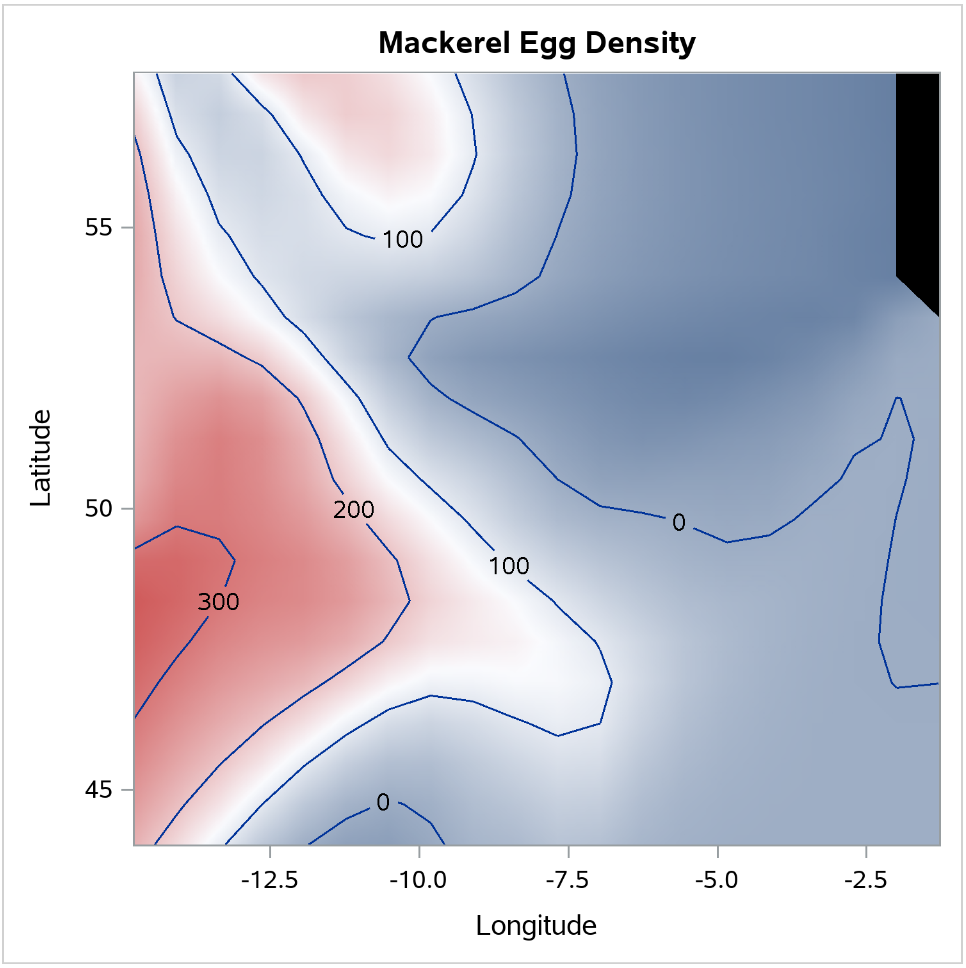
This is equivalent to a Poisson regression that uses the response variable Egg_Count, an offset variable $\log(\text{Net\_Area})$, and other covariates.

The following statements produce the plot of the mackerel egg density with respect to the sampling station location:

```
data Temp;
   set Mackerel;
   Density = Egg_Count/Net_Area;
run;

%let off0 = offsetmin=0 offsetmax=0
            linearopts=(thresholdmin=0 thresholdmax=0);
%let off0 = xaxisopts=(&off0) yaxisopts=(&off0);
proc template;
   define statgraph surface;
      dynamic _title _z;
      begingraph / designwidth=defaultDesignHeight;
         entrytitle _title;
         layout overlay / &off0;
            contourplotparm z=_z y=Latitude x=Longitude / gridded=FALSE;
         endlayout;
      endgraph;
   end;
run;

proc sgrender data=Temp template=surface;
   dynamic _title='Mackerel Egg Density' _z='Density';
run;
```

Output 46.3.1 displays the mackerel egg density in the sampling area. The black hole in the upper right corner is caused by missing values in that area.

Output 46.3.1 Mackerel Egg Density



In this example, the dependent variable is the mackerel egg count, the independent variables are the geographical information about each of the sampling stations, and the offset variable is the logarithm of the sampling area. The following statements use PROC GAMPL to fit the semiparametric Poisson regression model. In the program, one parametric term for Distance models the linear dependency of egg density, and another univariate spline term for Depth models the nonlinear dependency of egg density. The maximum degrees of freedom for the univariate spline term is set to 5 to enforce a smooth fit. One additional bivariate spline term for both Longitude and Latitude models the nonlinear spatial effect. To allow more flexibility of the bivariate spline term, its maximum degrees of freedom is increased to 40.

```
data Mackerel;
   set Mackerel;
   Depth = Depth/1000;
   Log_Net_Area = log(Net_Area);
run;

proc gampl data=Mackerel plots seed=321;
   model Egg_Count = param(Distance) spline(Depth/maxdf=5)
                     spline(Longitude Latitude/maxdf=40)
                      / offset=Log_Net_Area dist=poisson;
   output out=MackerelOut1 pred=Pred1 pearson=Pea1;
run;
```

Output 46.3.2 displays the summary statistics from the Poisson model.

Output 46.3.2 Fit Statistics

Mackerel Egg Density Study

The GAMPL Procedure

Fit Statistics	
Penalized Log Likelihood	-2792.30377
Roughness Penalty	8.67814
Effective Degrees of Freedom	43.40595
Effective Degrees of Freedom for Error	589.83438
AIC (smaller is better)	5662.74129
AICC (smaller is better)	5669.27963
BIC (smaller is better)	5855.98669
UBRE (smaller is better)	7.10457

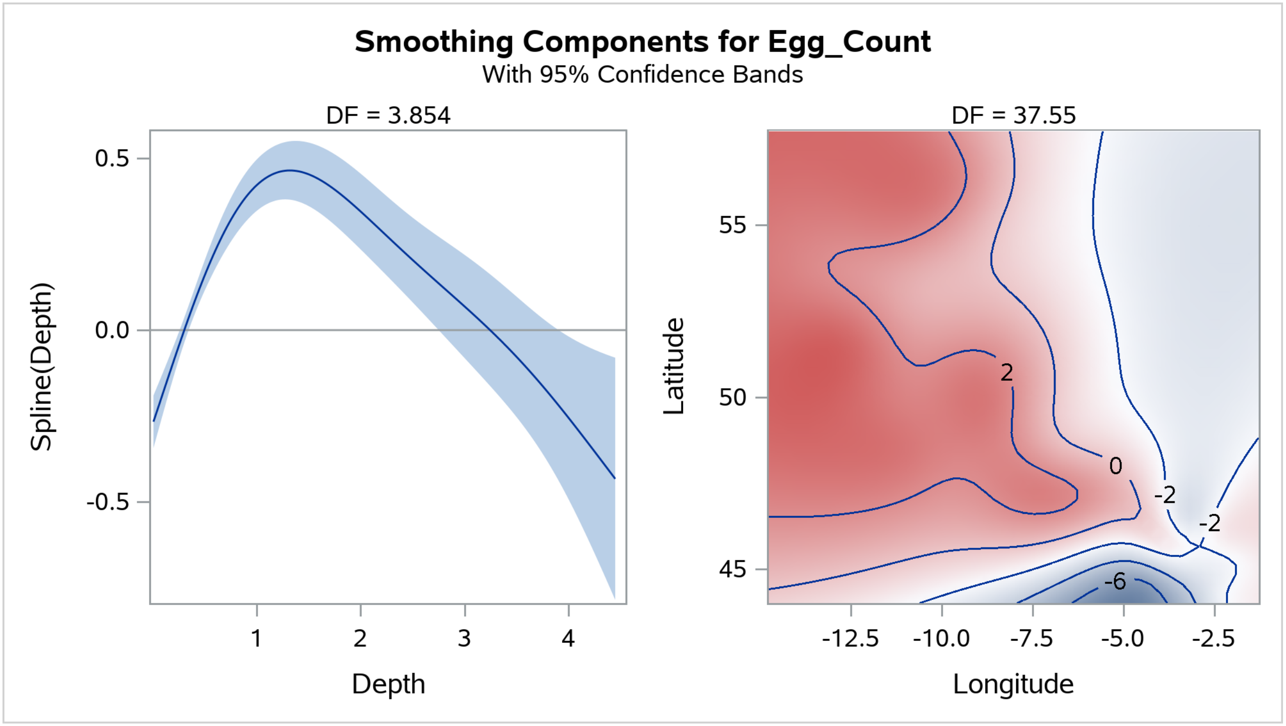
The "Tests for Smoothing Components" table in Output 46.3.3 shows that both univariate and bivariate spline terms are significant.

Output 46.3.3 Tests for Smoothing Components

Component	Effective DF	Effective DF for Test	Chi-Square	Pr > ChiSq
Spline(Depth)	3.85366	4	166.2720	<.0001
Spline(Longitude Latitude)	37.55229	39	4968.8032	<.0001

The smoothing component panel in Output 46.3.4 contains a contour for the spatial effect by Longitude and Latitude and one fitted univariate spline for Depth, both demonstrating significantly nonlinear structures.

Output 46.3.4 Smoothing Components Panel



Overdispersion sometimes occurs for count data. One way to measure whether the data are overdispersed is to determine whether Pearson's chi-square divided by the error degrees of freedom is much larger than 1. You can compute that statistic by taking the sum of squares of the Pearson residuals in the MackerelOut1 data set and then dividing that sum by the error degrees of freedom that is reported in Output 46.3.2. The computed value is approximately 8, which is much larger than 1, indicating the existence of overdispersion in this data set.

There are many ways to solve the overdispersion problem. One approach is to fit a negative binomial distribution in which the variance of the mean contains a dispersion parameter. The following statements use PROC GAMPL to fit the semiparametric negative binomial regression model:

```
proc gampl data=Mackerel plots seed=321;
   model Egg_Count = param(Distance) spline(Depth/maxdf=5)
                     spline(Longitude Latitude/maxdf=40)
                      / offset=Log_Net_Area dist=negbin;
   output out=MackerelOut2 pred=Pred2 pearson=Pea2;
run;
```

Output 46.3.5 displays the summary statistics from the negative binomial model. The model's effective degrees of freedom is less than that of the Poisson model, even though the dispersion parameter costs one extra degree of freedom to fit. The values of the information criteria are also much less, indicating a better fit.

Output 46.3.5 Fit Statistics

Mackerel Egg Density Study

The GAMPL Procedure

Fit Statistics	
Penalized Log Likelihood	-1561.91513
Roughness Penalty	17.57290
Effective Degrees of Freedom	38.88268
Effective Degrees of Freedom for Error	590.88741
AIC (smaller is better)	3184.02271
AICC (smaller is better)	3189.24304
BIC (smaller is better)	3357.13028
UBRE (smaller is better)	0.53927

The "Tests for Smoothing Components" table in Output 46.3.6 shows that both univariate and bivariate spline terms are significant.

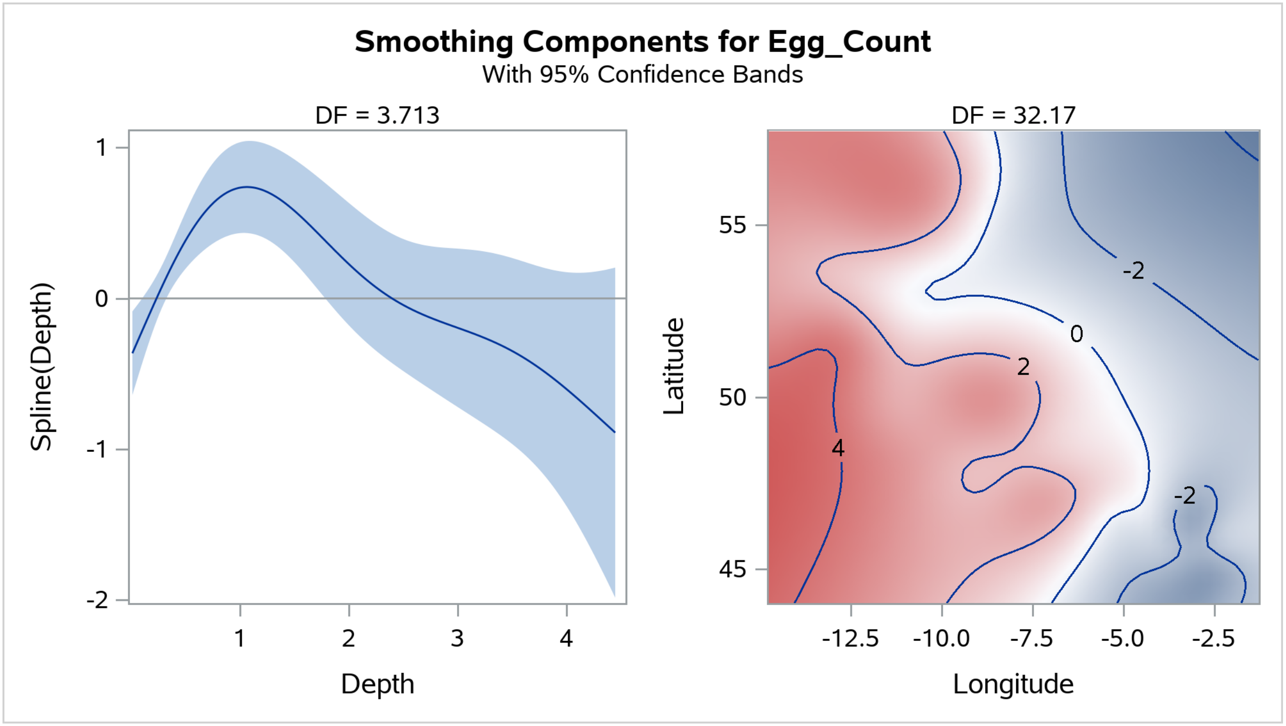
Output 46.3.6 Tests for Smoothing Components

Tests for Smoothing Components

Component	Effective DF	Effective DF for Test	F Value	Pr > F
Spline(Depth)	3.71286	4	32.88	<.0001
Spline(Longitude Latitude)	32.16982	37	1260.67	<.0001

The smoothing component panel in Output 46.3.7 contains a fitted surface and a curve for the two spline terms. Compared to the Poisson model, the fitted surface and curves are smoother. It is possible that the nonlinear dependency structure becomes smoother after its partial variations are accounted for by the dispersion parameter.

Output 46.3.7 Smoothing Components Panel



Pearson's chi-square divided by the error degrees of freedom for the negative binomial model is approximately 1.5, which is close to 1. This suggests that the negative binomial model explains the overdispersion in the data.

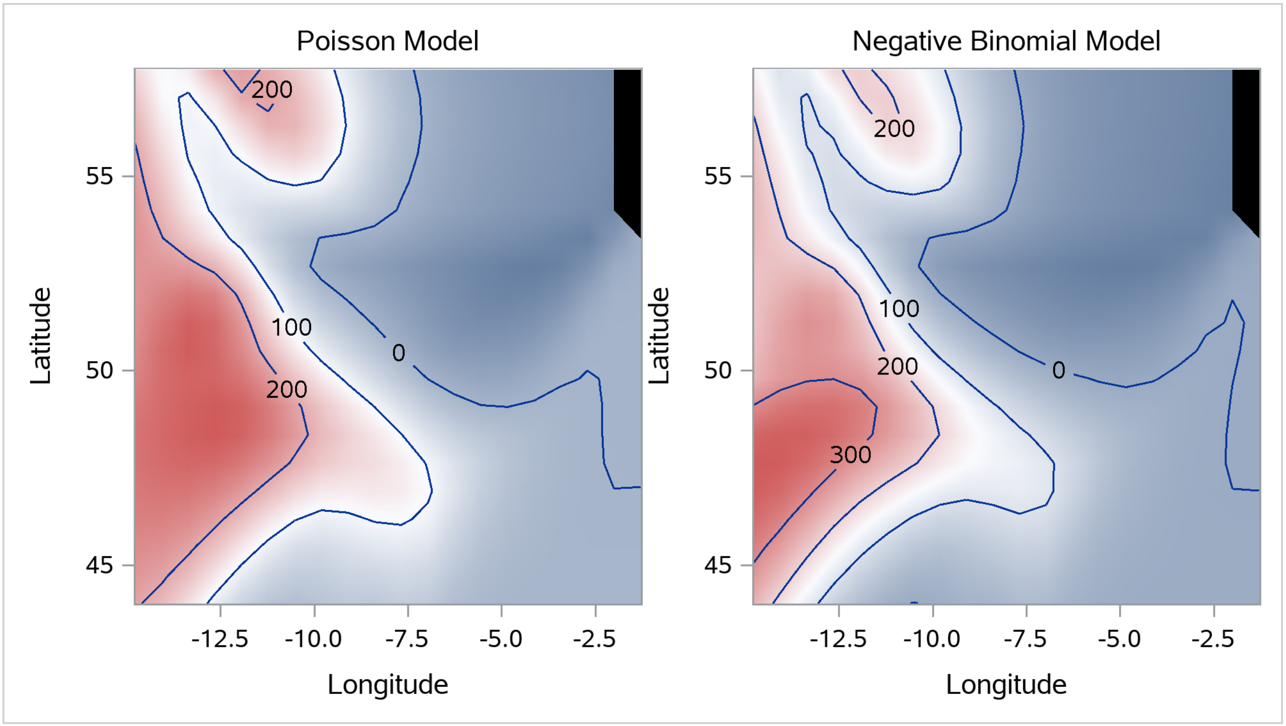
The following DATA step and statements visualize the contours of mackerel egg density for both Poisson and negative binomial models:

```
data SurfaceFit;
   set Mackerel(keep=Longitude Latitude Net_Area Egg_Count);
   set MackerelOut1(keep=Pred1);
   set MackerelOut2(keep=Pred2);
   Pred1 = Pred1 / Net_Area;
   Pred2 = Pred2 / Net_Area;
run;

%let eopt = location=outside valign=top textattrs=graphlabeltext;
proc template;
   define statgraph surfaces;
      begingraph / designheight=360px;
         layout lattice / columns=2;
            layout overlay / &off0;
               entry "Poisson Model" / &eopt;
               contourplotparm z=Pred1 y=Latitude x=Longitude /
                               gridded=FALSE;
            endlayout;
            layout overlay / &off0;
               entry "Negative Binomial Model" / &eopt;
               contourplotparm z=Pred2 y=Latitude x=Longitude /
```

```
                                    gridded=FALSE;
                 endlayout;
              endlayout;
           endgraph;
        end;
     run;

     proc sgrender data=SurfaceFit template=surfaces;
     run;
```

Output 46.3.8 displays two fitted surfaces from the Poisson model and the negative binomial model, respectively.

Output 46.3.8 Fitted Surfaces from Poisson Model and Negative Binomial Model



Example 46.4: Nonparametric Tweedie Regression

This example illustrates how you can use the GAMPL procedure to fit a nonparametric regression model with spline effects for a response variable that has a Tweedie distribution.

You can view this model as a generalized linear model because the Tweedie distribution belongs to the exponential family. The Tweedie distribution is particularly useful for modeling response variables that are continuous for positive values and take the value 0 with a positive probability. For example, in the insurance industry the Tweedie distribution is often assumed for claims, which are positive for customers who have filed claims and are 0 for other customers. You can use the Tweedie distribution to model such response variables without special transformations. For more information about this distribution, see the section "Tweedie Distribution For Generalized Linear Models" on page 3512.

The GENMOD procedure fits Tweedie regression models that incorporate linear effects for the covariates as illustrated in “Example 48.12: Tweedie Regression” on page 3601. However, as demonstrated in this example, the assumption of linearity can be too restrictive, and you might want to use the GAMPL procedure to explore the nonlinear dependency structures in the data.

The following DATA step simulates a response variable Y by using a Tweedie distribution and four continuous predictors. Each continuous predictor is sampled independently from the uniform distribution $U(0, 1)$. The linear predictor η for the true model is formed by applying additive nonlinear transformations to x_1, x_2, and x_3:

$$\eta = \frac{2\sin(\pi x_1) + 0.8\exp(2x_2) + 0.2x_3^{11}(10(1-x_3))^6 + 10(10x_3)^3(1-x_3)^{10}}{20}$$

The predictor x_4 is a nuisance parameter that does not enter the model. The parameters for the Tweedie distribution are $\phi = 0.4$ and $p = 1.5$, and the link function is $\log\mu = \eta$, where the expected value is $\mathrm{E}(Y) = \mu$ and the variance is $\mathrm{Var}(Y) = \phi\mu^p$. The power parameter p controls the variance of the distribution. When p is between 1 and 2, as in this example, a Tweedie random variable can be generated from a compound Poisson distribution (Smyth 1996).

```
title 'Nonparametric Tweedie Model';
%let phi=0.4;
%let power=1.5;

data one;
   do i=1 to 1000;

      /* Sample the predictors */
      x1=ranuni(1);
      x2=ranuni(1);
      x3=ranuni(1);
      x4=ranuni(1);

      /* Apply nonlinear transformations to predictors */
      f1=2*sin(3.14159265*x1);
      f2=exp(2*x2)*0.8;
      f3=0.2*x3**11*(10*(1-x3))**6+10*(10*x3)**3*(1-x3)**10;
      xb=f1+f2+f3;
      xb=xb/20;
      mu=exp(xb);

      /* Compute parameters of compound Poisson distribution */
      lambda=mu**(2-&power)/(&phi*(2-&power));
      alpha=(2-&power)/(&power-1);
      gamma=&phi*(&power-1)*(mu**(&power-1));

      /* Simulate the response */
      rpoi=ranpoi(1,lambda);
      if rpoi=0 then y=0;
      else do;
         y=0;
         do j=1 to rpoi;
            y=y+rangam(1,alpha);
         end;
         y=y*gamma;
      end;
```

```
         output;
      end;
   run;
```

You can use PROC GENMOD to fit a parametric model for *Y* by using the following statements:

```
   proc genmod data=one;
      model y=x1 x2 x3 x4/dist=tweedie;
   run;
```

Equivalently, you can also use PROC GAMPL to fit a parametric linear model by using the PARAM option in the MODEL statement, as shown in the following statements.

```
   proc gampl data=one seed=1234;
      model y=param(x1 x2 x3 x4)/dist=tweedie;
   run;
```

The "Fit Statistics" table in Output 46.4.1 shows the summary statistics for the fitted parametric Tweedie model.

Output 46.4.1 Fit Statistics

Nonparametric Tweedie Model

The GAMPL Procedure

Fit Statistics	
Penalized Log Likelihood	-1178.69507
Roughness Penalty	0
Effective Degrees of Freedom	7.00000
Effective Degrees of Freedom for Error	993.00000
AIC (smaller is better)	2371.39014
AICC (smaller is better)	2371.50305
BIC (smaller is better)	2405.74443

The "Parameter Estimates" table in Output 46.4.2 shows the estimates for regression parameters in addition to dispersion and power parameters (ϕ and p).

Output 46.4.2 Parameter Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept	1	0.441872	0.067335	43.0637	<.0001
x1	1	-0.043723	0.064966	0.4530	0.5009
x2	1	0.202190	0.064216	9.9135	0.0016
x3	1	-0.253928	0.066626	14.5257	0.0001
x4	1	-0.061118	0.066040	0.8565	0.3547
Dispersion	1	0.416005	0.016987		
Power	1	1.524591	0.048676		

If you are uncertain about the relationship between the linked mean for the fitted Tweedie model and the covariates, you can add spline terms to the model to explore possible nonlinear dependence. The following statements fit a nonparametric Tweedie model and produce a plot of fitted smoothing components.

```
proc gampl data=one seed=1234 plots;
   model y=spline(x1) spline(x2) spline(x3) spline(x4)/dist=tweedie;
run;
```

The “Fit Statistics” table in Output 46.4.3 shows the summary statistics for the fitted nonparametric Tweedie model. Compared to the parametric model, the nonparametric model has more effective degrees of freedom, but smaller AIC and AICC values. The penalized likelihood value is larger for the nonparametric model. These statistics indicate that the nonparametric model provides a better fit.

Output 46.4.3 Fit Statistics

Nonparametric Tweedie Model

The GAMPL Procedure

Fit Statistics	
Penalized Log Likelihood	-1157.17502
Roughness Penalty	2.62230
Effective Degrees of Freedom	14.11920
Effective Degrees of Freedom for Error	983.62098
AIC (smaller is better)	2339.96615
AICC (smaller is better)	2340.39965
BIC (smaller is better)	2409.25974
GCV (smaller is better)	0.39845

The “Regression Parameter Estimates” table in Output 46.4.4 lists the estimate for the intercept, in addition to the estimates for the dispersion and power parameters for the Tweedie model. Both variance parameters are close to the true values.

Output 46.4.4 Regression Parameter Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept	1	0.356825	0.018329	379.0079	<.0001
Dispersion	1	0.400459	0.518084		
Power	1	1.501771	1.514451		

The “Estimates for Smoothing Components” table in Output 46.4.5 shows the fitted information for the spline terms in the model. The degrees of freedom value and the roughness penalty value suggest some moderate nonlinear relationship between the linked mean and the three covariates x_1, x_2, and x_3. The spline term for the nuisance parameter x_4 has a linear form.

Output 46.4.5 Estimates for Smoothing Components

Estimates for Smoothing Components

Component	Effective DF	Smoothing Parameter	Roughness Penalty	Number of Parameters	Rank of Penalty Matrix	Number of Knots
Spline(x1)	2.10969	0.9560	0.5659	9	10	1000
Spline(x2)	2.38340	0.5604	0.3814	9	10	1000
Spline(x3)	5.62608	0.0145	1.6750	9	10	1000
Spline(x4)	1.00002	133078	3.662E-6	9	10	1000

The "Tests for Smoothing Components" table in Output 46.4.6 shows the approximate Wald test for the four spline terms. The spline term for the nuisance predictor x_4 is not significant.

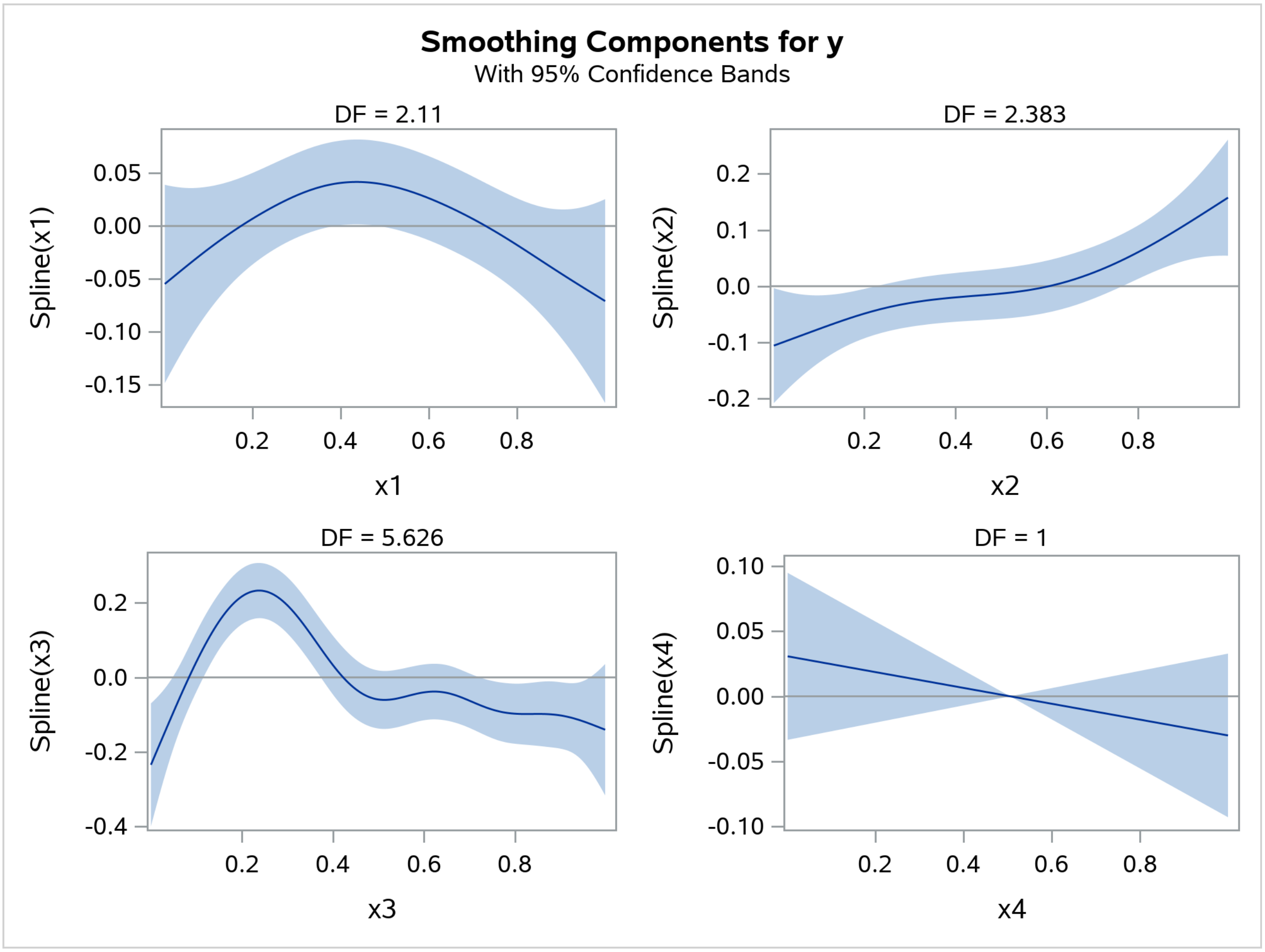
Output 46.4.6 Tests for Smoothing Components

Tests for Smoothing Components

Component	Effective DF	Effective DF for Test	F Value	Pr > F
Spline(x1)	2.10969	3	4.76	0.0027
Spline(x2)	2.38340	3	13.64	<.0001
Spline(x3)	5.62608	7	49.02	<.0001
Spline(x4)	1.00002	1	0.88	0.3484

Output 46.4.7 displays the "Smoothing Component Panel" for all the spline terms used in the model. For x_1, x_2, and x_3, the fitted curves are reasonably smooth and close to the true functions. For x_4, the spline plot shows a strict a linear fit, with 95% Bayesian confidence bands covering the horizontal line at 0. This reinforces the conclusion that x_4 does not contribute to the model and can be removed from further analysis.

Output 46.4.7 Smoothing Component Panel



References

Bowman, A. W., and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis*. New York: Oxford University Press.

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

Dennis, J. E., Gay, D. M., and Welsch, R. E. (1981). "An Adaptive Nonlinear Least-Squares Algorithm." *ACM Transactions on Mathematical Software* 7:348–368.

Dennis, J. E., and Mei, H. H. W. (1979). "Two New Unconstrained Optimization Algorithms Which Use Function and Gradient Values." *Journal of Optimization Theory and Applications* 28:453–482.

Duchon, J. (1976). "Fonctions-spline et espérances conditionnelles de champs gaussiens." *Annales scientifiques de l'Université de Clermont-Ferrand 2, Série Mathématique* 14:19–27.

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

Eskow, E., and Schnabel, R. B. (1991). "Algorithm 695: Software for a New Modified Cholesky Factorization." *ACM Transactions on Mathematical Software* 17:306–312.

Fletcher, R. (1987). *Practical Methods of Optimization*. 2nd ed. Chichester, UK: John Wiley & Sons.

Gay, D. M. (1983). "Subroutines for Unconstrained Minimization." *ACM Transactions on Mathematical Software* 9:503–524.

Gu, C., and Wahba, G. (1991). "Minimizing GCV/GML Scores with Multiple Smoothing Parameters via the Newton Method." *SIAM Journal on Scientific Computing* 12:383–398.

Moré, J. J., and Sorensen, D. C. (1983). "Computing a Trust-Region Step." *SIAM Journal on Scientific and Statistical Computing* 4:553–572.

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

Nychka, D. (1988). "Bayesian Confidence Intervals for Smoothing Splines." *Journal of the American Statistical Association* 83:1134–1143.

Pace, R. K., and Barry, R. (1997). "Quick Computation of Spatial Autoregressive Estimators." *Geographical Analysis* 29:232–247.

Ruppert, D., Wand, M. P., and Carroll, R. J. (2003). *Semiparametric Regression*. Cambridge: Cambridge University Press.

Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C., and Johannes, R. S. (1988). "Using the ADAP Learning Algorithm to Forecast the Onset of Diabetes Mellitus." In *Proceedings of the Symposium on Computer Applications and Medical Care*, 261–265. Los Alamitos, CA: IEEE Computer Society Press.

Smyth, G. K. (1996). "Regression Analysis of Quantity Data with Exact Zeros." In *Proceedings of the Second Australia-Japan Workshop on Stochastic Models in Engineering, Technology, and Management*, edited by R. J. Wilson, S. Osaki, and D. N. P. Murthy, 572–580. Queensland, Australia: Technology Management Centre, University of Queensland.

Vlachos, P. (1998). "StatLib—Datasets Archive." `http://lib.stat.cmu.edu/datasets/`.

Wahba, G. (1983). "Bayesian 'Confidence Intervals' for the Cross Validated Smoothing Spline." *Journal of the Royal Statistical Society, Series B* 45:133–150.

Wood, S. (2003). "Thin Plate Regression Splines." *Journal of the Royal Statistical Society, Series B* 65:95–114.

Wood, S. (2004). "Stable and Efficient Multiple Smoothing Parameter Estimation for Generalized Additive Models." *Journal of the American Statistical Association* 99:673–686.

Wood, S. (2006). *Generalized Additive Models*. Boca Raton, FL: Chapman & Hall/CRC.

Wood, S. (2008). "Fast Stable Direct Fitting and Smoothness Selection for Generalized Additive Models." *Journal of the Royal Statistical Society, Series B* 70:495–518.

Wood, S. (2011). "Fast Stable Restricted Maximum Likelihood and Marginal Likelihood Estimation of Semiparametric Generalized Linear Models." *Journal of the Royal Statistical Society, Series B* 73:3–36.

Wood, S. (2012). "On p-Values for Smooth Components of an Extended Generalized Additive Model." *Biometrika* 1–8. `http://biomet.oxfordjournals.org/content/early/2012/10/18/biomet.ass048.abstract`.

Xiang, D., and Wahba, G. (1996). "A Generalized Approximate Cross Validation for Smoothing Splines with Non-Gaussian Data." *Statistica Sinica* 6:675–692.

Subject Index

Syntax Index