Paper SAS2033-2018

Regression Model Building for Large, Complex Data with SAS® Viya® Procedures

Robert N. Rodriguez and Weijie Cai, SAS Institute Inc.

Abstract

Analysts who do statistical modeling, data mining, and machine learning often ask the following question: "I have hundreds of variables—even thousands. Which should I include in my regression model?" This paper describes SAS® Viya® procedures for building linear and logistic regression models, generalized linear models, quantile regression models, generalized additive models, and proportional hazards regression models. The paper explains how these procedures capitalize on the in-memory environment of SAS Viya, and it compares their syntax, features, and output with those of high-performance regression modeling procedures in SAS/STAT® software.

Introduction

High-dimensional data now provide the foundation for many business applications and fields of scientific research. Because these data are increasingly large and complex, they require greater computational power for building regression models that are interpretable or that accurately predict future responses. When interpretability is the goal, you need inferential results, such as standard errors and p-values, to decide which effects are important. When prediction is the goal, you need to evaluate the accuracy of prediction and assess whether it could be improved by a sparser, more parsimonious model.

This paper describes six statistical procedures available in SAS Viya that meet these needs. In addition to fitting models, these procedures provide modern approaches for building models by selecting variables and effects, such as classification effects and spline effects. These approaches rely on penalized least squares and penalized likelihood as theoretical frameworks for variable selection and feature extraction.

The paper is organized into six main sections, one for each procedure:

- "Building Least Squares Regression Models with the REGSELECT Procedure"
- "Building Logistic Regression Models with the LOGSELECT Procedure"
- "Building Generalized Linear Models with the GENSELECT Procedure"
- "Building Quantile Regression Models with the QTRSELECT Procedure"
- "Fitting Generalized Additive Models with the GAMMOD Procedure"
- "Building Proportional Hazards Regression Models with the PHSELECT Procedure"

Each section introduces a procedure by explaining its approach and illustrating its use with a basic example.

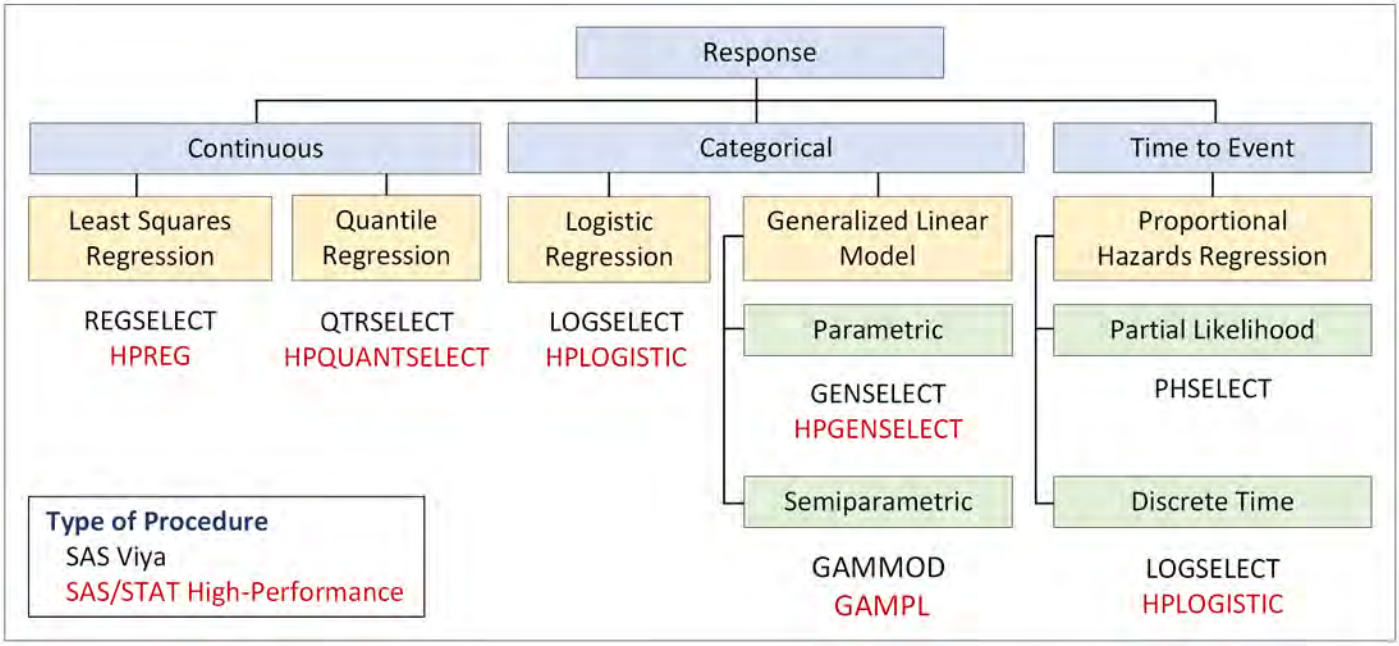
Figure 1 shows the different classes of regression models that are supported by the six SAS Viya procedures. With the exception of the PHSELECT procedure, these procedures are successors to high-performance regression procedures in SAS/STAT® software that have similar functionality, and which are described by Rodriguez (2016).

SAS Viya is the third generation of SAS® software for high-performance in-memory analytics, and the analytic engine in SAS Viya is SAS® Cloud Analytic Services (CAS). Because the SAS Viya statistical procedures were developed specifically for CAS, they enable you to do the following:

- run on a cluster of machines that distribute the data and the computations
- run in single-machine mode
- exploit all the available cores and concurrent threads

These procedures operate only on in-memory CAS tables, and you must license SAS® Visual Statistics to run them. If you also have SAS® 9.4M5 installed, you can run procedures in both SAS Viya and SAS 9.4M5 from the same SAS interface, such as the SAS windowing environment, SAS® Enterprise Guide®, and SAS® Studio. (Note that if you have only licensed SAS Visual Statistics, SAS/STAT procedures are included and you can access them through SAS Studio.)

Figure 1 SAS Viya Procedures and SAS/STAT High-Performance Procedures for Regression Modeling



Building Least Squares Regression Models with the REGSELECT Procedure

The REGSELECT procedure fits and builds general linear models of the form

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i, \quad i = 1, \ldots, n$$

where the response y_i is continuous and the predictors $x_{i1}, \ldots, x_{ip}$ represent main effects that consist of continuous or classification variables and can include interaction effects or constructed effects of these variables.

With too many predictors, the model can overfit the training data, leading to poor prediction with future data. To deal with this problem, the REGSELECT procedure supports the model selection methods summarized in Table 1.

Table 1 Effect Selection Methods in the REGSELECT Procedure

Method	Description
Forward selection	Starts with no effects in the model and adds effects
Forward swap	Before adding an effect, makes all pairwise swaps of in-model and out-of-model effects that improve the selection criterion
Backward elimination	Starts with all effects in the model and deletes effects
Stepwise selection	Starts with no effects in the model and adds or deletes effects
Least angle regression	Starts with no effects and adds effects; at each step, $\hat{\beta}$s shrink toward zero
Lasso	Constrains the sum of absolute $\hat{\beta}$s; some $\hat{\beta}$s are set to zero, others shrink toward zero

For each method, PROC REGSELECT supports modern model evaluation criteria for selection and stopping, which penalize large numbers of parameters in a principled manner. The procedure also supports stopping and selection based on external validation and leave-one-out cross validation. In practice, no single method consistently outperforms the rest, but—depending on your goal—an informed and judicious choice of these features can lead to models that have greater predictive accuracy or models that are more interpretable.

The first four methods in Table 1 estimate the regression coefficients for candidate models by solving the following least squares problem:

$$\min_{\beta_0,\ldots,\beta_p} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2$$

In contrast, the lasso method places an ℓ_1 penalty on the coefficients:

$$\min_{\beta_0,\ldots,\beta_p} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2$$

$$\text{subject to } \sum_{j=1}^{p} \left|\beta_j\right| \leq t$$

For large values of t, the lasso method produces ordinary least squares estimates. Decreasing t in discrete steps leads to a sequence of coefficient estimates, where some are exactly zero and the rest, which correspond to selected effects, are shrunk toward zero. This mitigates the influence of selected effects that do not belong in the model, and it distinguishes the lasso method from methods such as forward selection, as illustrated in the following example.

Example: Predicting the Mean Close Rate for Retail Stores

The close rate for a retail store is the percentage of shoppers who enter the store and make a purchase. Understanding what factors predict the mean close rate is critical to the profitability and growth of large retail companies, and a regression model is constructed to study this question.

The close rates for 500 stores are saved in a CAS table named **Stores**. Each observation provides information about a store. The variables available for the model are the response **Close_Rate** and the following candidate predictors:

- **X1**, . . . , **X20**, which measure 20 general characteristics of stores, such as floor size and number of employees
- **P1**, . . . , **P6**, which measure six promotional activities, such as advertising and sales
- **L1**, . . . , **L6**, which measure special layouts of items in six departments

In practice, close rate data can involve hundreds of candidate predictors. A small set is used here for illustration.

Results with the Forward Selection Method

The following statements use the forward selection method in the REGSELECT procedure to build a model:

```
ods graphics on;
proc regselect data=mycas.Stores;
   model Close_Rate = X1-X20 L1-L6 P1-P6;
   selection method=forward plots=all;
run;
```

The DATA= option specifies a CAS table named **mycas.Stores**. The first level of the name is the CAS engine libref, and the second level is the table name. The SELECTION statement requests model selection. The settings for the selection process are listed in Figure 2.

Figure 2 Selection Information

The REGSELECT Procedure

Selection Information	
Selection Method	Forward
Select Criterion	SBC
Stop Criterion	SBC
Effect Hierarchy Enforced	None
Stop Horizon	3

By default, the REGSELECT procedure uses the Schwarz Bayesian criterion (SBC) as the selection criterion for determining the order in which effects enter at each step. The effect that is selected is the one whose addition maximizes the decrease in SBC. By default, the procedure also uses SBC as the stop criterion. Selection stops at the step where the next step yields a model that has a larger SBC value. The stop horizon, which is 3 by default, specifies the number of consecutive steps at which the stop criterion must decrease in order to detect a minimum.

As shown in Figure 3, the minimum value of SBC is reached at Step 9, when **P1** enters the model.

Figure 3 Selection Summary with Forward Selection

The REGSELECT Procedure

Selection Details

Selection Summary

Step	Effect Entered	Number Effects In	SBC
0	**Intercept**	1	47.8155
1	**X2**	2	-27.1875
2	**X4**	3	-52.4996
3	**P3**	4	-60.6381
4	**P4**	5	-67.4347
5	**L1**	6	-73.7232
6	**L3**	7	-79.3681
7	**P5**	8	-83.1847
8	**L2**	9	-86.2457
9	**P1**	10	-88.6068*
10	**L5**	11	-87.8923
11	**P2**	12	-84.7692

*** Optimal Value Of Criterion**

The coefficient progression plot in Figure 4, requested by the PLOTS= option, visualizes the selection process.

Figure 4 Coefficient Progression with Forward Selection

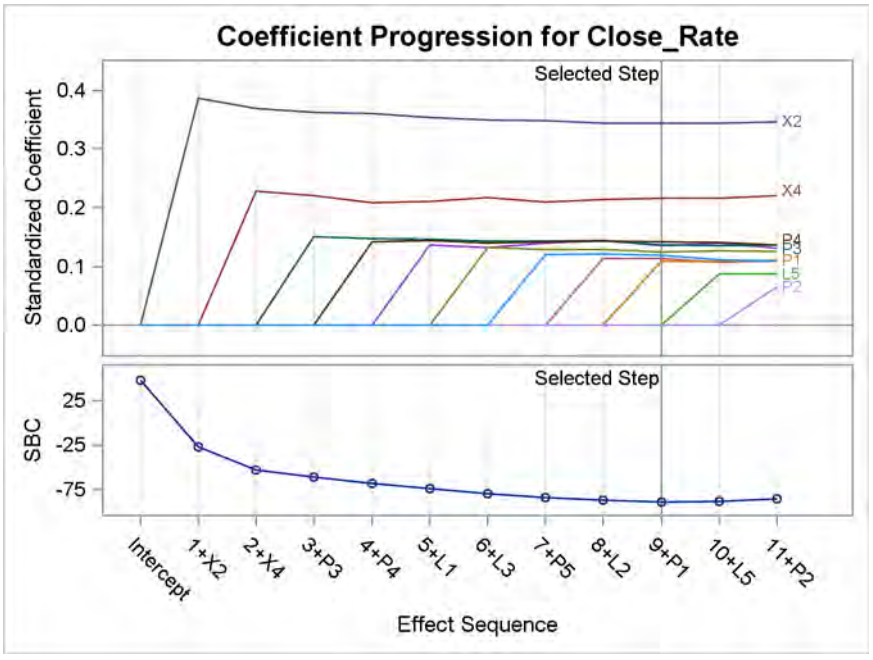


Figure 5 shows the parameter estimates for the final model. The estimates for **X2** and **X4** are larger than the estimates for the seven other predictors, and all the standard errors are comparable. The p-values should be interpreted with care because they are computed conditionally on the final selected model and do not take into account the process by which the model was selected.

Figure 5 Parameter Estimates for Model Selected with Forward Selection

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	60.412202	0.119136	507.09	<.0001
X2	1	1.225952	0.133595	9.18	<.0001
X4	1	0.798252	0.138799	5.75	<.0001
L1	1	0.496037	0.137290	3.61	0.0003
L2	1	0.379632	0.125270	3.03	0.0026
L3	1	0.438092	0.131785	3.32	0.0010
P1	1	0.400154	0.137440	2.91	0.0038
P3	1	0.479429	0.131241	3.65	0.0003
P4	1	0.520183	0.136973	3.80	0.0002
P5	1	0.420284	0.132103	3.18	0.0016

Results with the Lasso Method

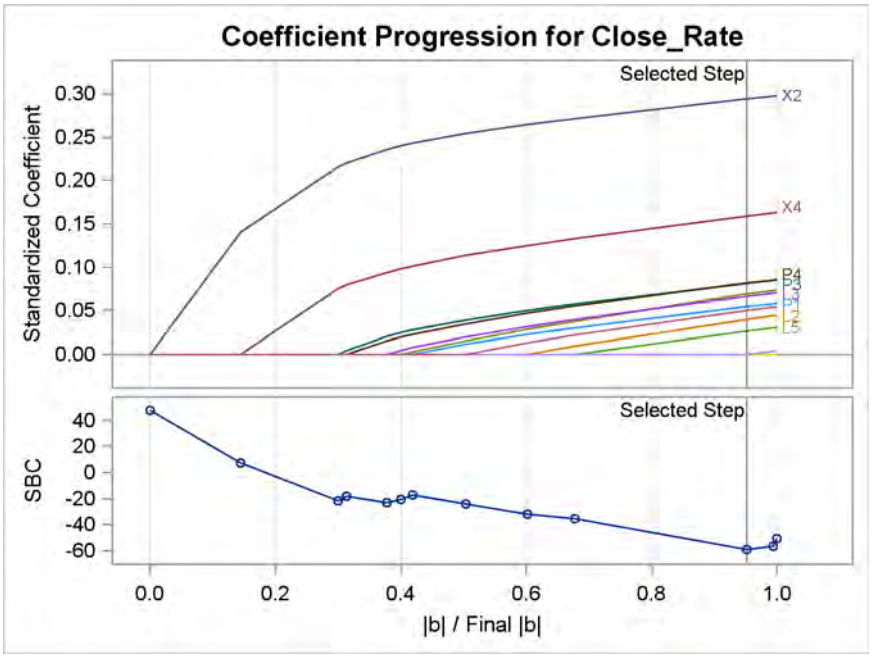
The following statements use the lasso method to build a model:

```
proc regselect data=mycas.Stores;
   model Close_Rate = X1-X20 L1-L6 P1-P6;
   selection method=lasso plots(stepaxis=normb)=all;
run;
```

For the lasso method, the REGSELECT procedure uses the least angle regression algorithm, introduced by Efron et al. (2004), to produce a sequence of regression models in which one parameter is added at each step. By default, the selection criterion is SBC, the stop criterion is SBC, and the stop horizon is 3.

The lasso method selects a model that has 10 variables when the minimum value of SBC is reached at Step 10, as shown in Figure 6. The stop horizon enables the small local minima of SBC at Steps 3 and 5 to be ignored.

Figure 6 Coefficient Progression with Lasso Method



The scale for the horizontal axis, requested by the STEPAXIS= suboption, is more appropriate for the lasso method than the default step scale in Figure 4 because it expresses the size of the ith step as the ℓ_1 norm of the parameters relative to the ℓ_1 norm of the parameters at the final step.

The parameter estimates for the final model are shown in Figure 7. These estimates are closer to zero than the corresponding estimates in Figure 5.

Figure 7 Parameter Estimates for Model Selected with Lasso Method

Parameter Estimates		
Parameter	**DF**	**Estimate**
Intercept	1	60.720592
X2	1	1.046158
X4	1	0.584167
L1	1	0.254223
L2	1	0.133271
L3	1	0.235543
L5	1	0.095443
P1	1	0.199461
P3	1	0.285626
P4	1	0.298742
P5	1	0.176449

Comparison with the HPREG and GLMSELECT Procedures

The functionality of the REGSELECT procedure closely resembles that of the HPREG procedure, which is a SAS/STAT high-performance procedure. Both procedures perform model selection for ordinary least squares regression models, which you can specify as general linear models. You request model selection by using the SELECTION statement.

The functionality of the REGSELECT procedure also resembles that of the GLMSELECT procedure in SAS/STAT, which is multithreaded. Both procedures offer multiple methods of effect selection, the ability to use external validation data and cross validation as selection criteria, and extensive options to customize the selection process. Both procedures provide the ability to specify constructed effects in the EFFECT statement. The preceding example produces the same results when run with the GLMSELECT procedure (Rodriguez 2016), provided that you specify a stop horizon of 1. The default stop horizon in the REGSELECT procedure is 3 because it provides better protection against small local minima in the stop criterion.

Building Logistic Regression Models with the LOGSELECT Procedure

The LOGSELECT procedure fits and builds binary response models of the form

$$g(\pi_i) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i, \quad i = 1, \ldots, n$$

where π_i is the predicted probability of an event and the link function g is the logit, probit, log-log, or complementary log-log function. As in models supported by the REGSELECT procedure, the predictors represent main effects that consist of continuous or classification variables and can include interaction effects or constructed effects of these variables. When the response has more than two values, the LOGSELECT procedure fits and builds ordinal response models and generalized logit models.

With too many predictors, the model can overfit the training data, leading to poor prediction with future data. To deal with this problem, the LOGSELECT procedure supports the selection methods summarized in Table 2.

Table 2 Effect Selection Methods in the LOGSELECT Procedure

Method	Description
Forward selection	Starts with no effects in the model and adds effects
Backward elimination	Starts with all effects in the model and deletes effects
Backward (fast)	Starts with all effects in the model and deletes effects without refitting the model
Stepwise selection	Starts with no effects in the model and adds or deletes effects
Lasso	Constrains the sum of absolute $\hat{\beta}$s; some $\hat{\beta}$s are set to zero, others shrink toward zero

Example: Predicting High Customer Satisfaction for Retail Stores

The CAS table **Stores** in the previous example contains a binary response variable named **HighSatisfaction**, which is equal to 1 if a store achieved the highest level of satisfaction in a customer survey and is equal to 0 otherwise. The following statements use the LOGSELECT procedure to build a logistic regression model for predicting high satisfaction:

```
proc logselect data=mycas.Stores;
   model HighSatisfaction(event='1') = X1-X20 L1-L6 P1-P6;
   selection method=forward(choose=validate) plots=all;
   partition fraction(validate=0.25 seed=14591);
   code file='HighPredict.sas';
run;
```

The PARTITION statement partitions the observations into disjoint subsets for model training (75%) and model validation (25%). The SELECTION statement requests model selection based on the forward method. At each step, the training data are used to fit the candidate model. The CHOOSE=VALIDATE suboption requests that the average square error (ASE) be computed on the validation data for the model at each step of the selection process. The selected model is the smallest model at any step that yields the lowest ASE. The CODE statement writes SAS DATA step code for predicting high satisfaction to a file named *HighPredict.sas*.

Figure 8 shows the number of observations at each level of the response in each of the partitions.

Figure 8 Partition Counts

The LOGSELECT Procedure

Response Profile

Ordered Value	HighSatisfaction	Total Frequency	Training	Validation
1	0	379	293	86
2	1	121	90	31

Probability modeled is HighSatisfaction = 1.

As shown in Figure 9, the minimum validation ASE is reached at Step 5, when **X4** enters the model.

Figure 9 Coefficient Progression with Forward Selection

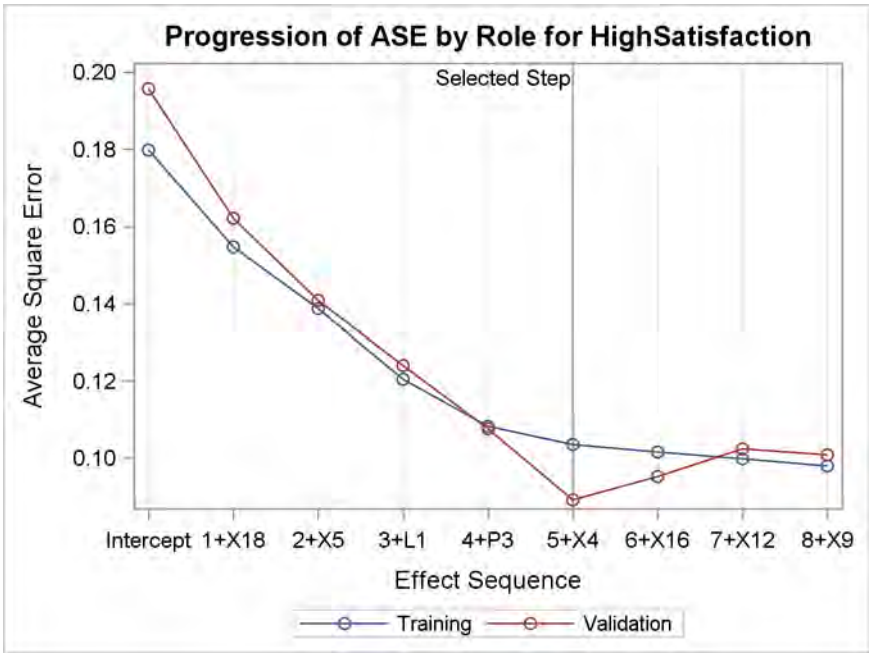


Figure 10 displays the fit statistics for the selected model, which are computed for both the training and validation data. The ASE, misclassification rate, and difference of means are comparable for the two groups of data, indicating a good predictive fit.

Figure 10 Fit Statistics for Training and Validation Partitions

Fit Statistics		
Description	**Training**	**Validation**
-2 Log Likelihood	247.84673	65.04619
AIC (smaller is better)	259.84673	77.04619
AICC (smaller is better)	260.07014	77.80983
SBC (smaller is better)	283.53494	93.61923
Average Square Error	0.10347	0.08912
-2 Log L (Intercept-only)	417.64791	135.29376
R-Square	0.35811	0.45141
Max-rescaled R-Square	0.53938	0.65864
McFadden's R-Square	0.40657	0.51922
Misclassification Rate	0.14621	0.14530
Difference of Means	0.42865	0.51168

Figure 11 shows the parameter estimates for the selected model.

Figure 11 Parameter Estimates for Selected Model

Parameter Estimates					
Parameter	**DF**	**Estimate**	**Standard Error**	**Chi-Square**	**Pr > ChiSq**
Intercept	1	-4.063480	0.494970	67.3967	<.0001
X4	1	1.955431	0.588277	11.0489	0.0009
X5	1	-4.346695	0.689653	39.7243	<.0001
X18	1	-5.018940	0.705420	50.6207	<.0001
L1	1	3.916054	0.677884	33.3723	<.0001
P3	1	-3.249121	0.634598	26.2140	<.0001

Comparison with the HPLOGISTIC Procedure

The functionality of the LOGSELECT procedure closely resembles that of the HPLOGISTIC procedure, which is a SAS/STAT high-performance procedure. In addition to the selection methods available in the HPLOGISTIC procedure, the LOGSELECT procedure provides the lasso method. The LOGSELECT procedure also produces selection plots and constructs complex effects, such as univariate spline effects and polynomial effects—features not available in the HPLOGISTIC procedure.

Building Generalized Linear Models with the GENSELECT Procedure

The GENSELECT procedure fits and builds generalized linear models, which can analyze many types of responses. A generalized linear model consists of three components:

- A linear predictor, which is defined in the same way as for general linear models:

$$\eta_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}, \quad i = 1, \ldots, n$$

- A specified link function g, which describes how μ_i, the expected value of y_i, is related to η_i:

$$g(\mu_i) = \eta_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$$

- An assumed distribution for the responses y_i. For distributions in the exponential family, the variance of the response depends on the mean μ through a variance function V,

$$\mathrm{Var}(y_i) = \frac{\phi V(\mu_i)}{w_i}$$

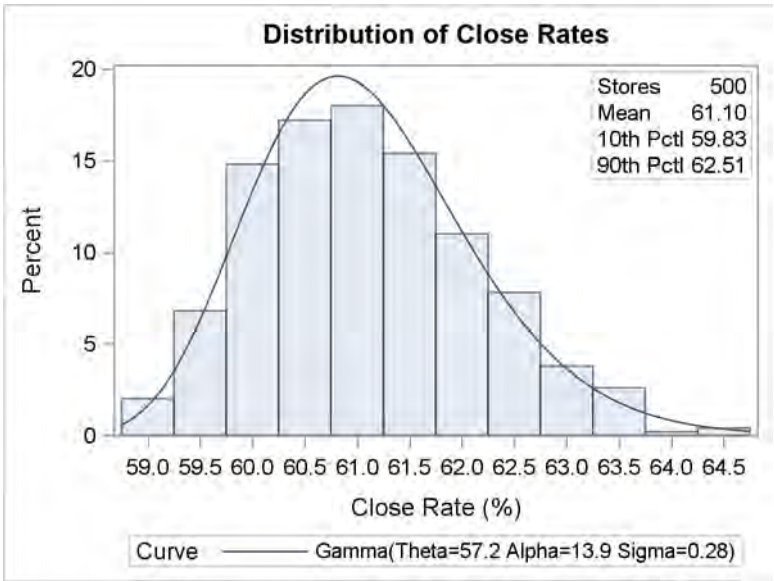
where ϕ is a constant and w_i is a known weight for each observation. The dispersion parameter ϕ is either estimated or known (for example, $\phi = 1$ for the binomial distribution).

The GENSELECT procedure supports standard response distributions in the exponential family, such as the normal, Poisson, and Tweedie distributions. In addition, the procedure supports ordinal and unordered multinomial response distributions. For all these distributions, the GENSELECT procedure estimates model parameters by using maximum likelihood techniques. To deal with the problem of overfitting, the procedure provides the forward, backward, fast backward, stepwise, and lasso methods of effect selection.

Example: Predicting the Mean Close Rate for Retail Stores (continued)

Figure 12 shows the marginal distribution of the close rates in **Stores**. A gamma distribution provides a good fit, suggesting that a gamma regression model for the conditional mean of close rate might improve on the model that was obtained with the REGSELECT procedure in the example on page 3.

Figure 12 Distribution of Close Rates for 500 Stores



The following statements use the GENSELECT procedure to build a gamma regression model. A preliminary shift transformation is applied to **Close_Rate** because the gamma distribution has a threshold at 0.

```
data mycas.Stores; set mycas.Stores;
   Close_Rate_0 = Close_Rate - 58;
run;

proc genselect data=mycas.Stores;
   model Close_Rate_0 = X1-X20 L1-L6 P1-P6 / distribution=gamma link=log;
   selection method=forward;
run;
```

The METHOD= option requests the forward selection method. The default criterion for choosing the model at each step is SBC, which is also the default stop criterion. The default stop horizon is 3.

Figure 13 shows that the minimum SBC value is reached at Step 9, when **P1** enters the model. The selected variables happen to be the same as those selected by the REGSELECT procedure when it uses the forward method, as shown in Figure 3. However, an additional dispersion parameter is estimated for the gamma regression model.

Figure 13 Selection Summary with Forward Method

The GENSELECT Procedure

Selection Details

Selection Summary

Step	Effect Entered	Number Effects In	SBC
0	Intercept	1	1456.6448
1	X2	2	1391.4830
2	X4	3	1359.0476
3	P3	4	1348.9659
4	P4	5	1341.2893
5	L3	6	1334.9530
6	L1	7	1330.0412
7	P5	8	1325.6188
8	L2	9	1322.8629
9	P1	10	1320.4778*
10	L5	11	1322.7962
11	X3	12	1325.2128
12	P2	13	1327.5805

* Optimal Value Of Criterion

Figure 14 shows the parameter estimates for the selected model. As in Figure 5, the estimates for **X2** and **X4** are larger in magnitude than the estimates for the other predictors.

Figure 14 Parameter Estimates for Gamma Regression Model Selected with Forward Method

Parameter Estimates

Parameter	DF	Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept	1	0.882152	0.039605	496.1157	<.0001
X2	1	0.412054	0.044286	86.5731	<.0001
X4	1	0.273341	0.046033	35.2594	<.0001
L1	1	0.161623	0.045409	12.6684	0.0004
L2	1	0.124663	0.041233	9.1406	0.0025
L3	1	0.153185	0.043612	12.3371	0.0004
P1	1	0.132601	0.045129	8.6334	0.0033
P3	1	0.168153	0.043051	15.2564	<.0001
P4	1	0.183079	0.045080	16.4931	<.0001
P5	1	0.142619	0.043513	10.7428	0.0010
Dispersion	1	12.194522	0.760933		

Comparison with the HPGENSELECT and GENMOD Procedures

The functionality of the GENSELECT procedure resembles that of the SAS/STAT high-performance HPGENSELECT procedure. The GENSELECT procedure is additionally capable of constructing complex effects, such as univariate spline and polynomial expansions. The HPGENSELECT procedure (but not the GENSELECT procedure) provides models for zero-inflated data. The GENSELECT procedure uses the log link function as the default for both the gamma and the inverse Gaussian distributions. The HPGENSELECT procedure uses the reciprocal link function as the default for the gamma distribution, and it uses the reciprocal squared link function as the default for the inverse Gaussian distribution.

The GENSELECT procedure, the HPGENSELECT procedure, and the GENMOD procedure in SAS/STAT fit generalized linear models. However, there are important design differences in their capabilities, as summarized in Table 3.

Table 3 Comparison of the GENSELECT, HPGENSELECT, and GENMOD Procedures

GENSELECT and HPGENSELECT Procedures	GENMOD Procedure
Fit and build generalized linear models	Fits generalized linear models
Analyze large to massive data	Analyzes moderate to large data
Designed for predictive modeling	Designed for inferential analysis
Run in single-machine or distributed mode	Runs in single-machine mode
Use all cores and concurrent threads	Is single threaded

Building Quantile Regression Models with the QTRSELECT Procedure

The QTRSELECT procedure fits and builds quantile regression models, which predict the quantiles (or equivalently, the percentiles) of a continuous response variable. The quantile regression model for the τth quantile (100τth percentile) is of the form

$$Q_\tau(y_i) = \beta_0(\tau) + \beta_1(\tau)x_{i1} + \cdots + \beta_p(\tau)x_{ip}, \quad i = 1, \ldots, n, \quad 0 < \tau < 1$$

where the predictors $x_{i1}, \ldots, x_{ip}$ represent main effects that consist of continuous or classification variables and can include interaction effects or constructed effects of these variables. The regression coefficients $\beta_j(\tau)$ are estimated by solving the minimization problem

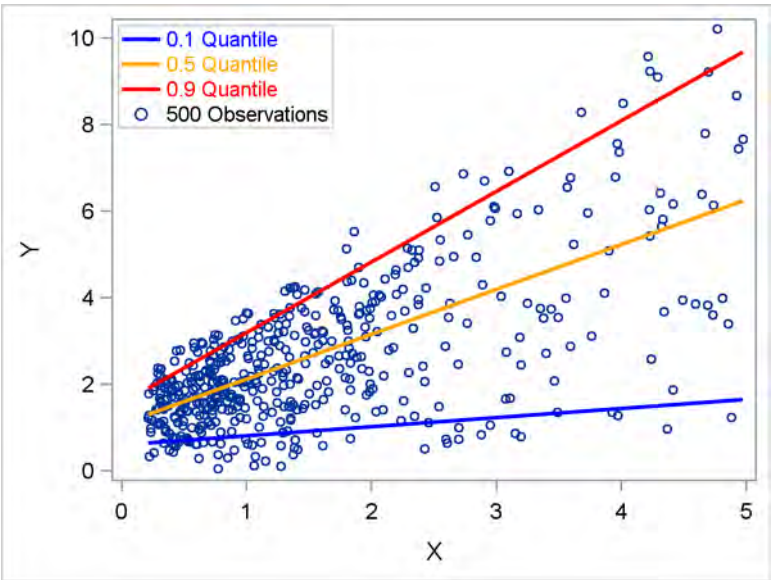
$$\min_{\beta_0,\ldots,\beta_p} \sum_{i=1}^{n} \rho_\tau \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)$$

where $\rho_\tau(r) = \tau \max(r, 0) + (1-\tau)\max(-r, 0)$. The function $\rho_\tau(r)$ is referred to as the check loss function. To avoid overfitting, the QTRSELECT procedure provides the forward, backward, and stepwise methods of effect selection.

Quantile regression was introduced 40 years ago by Koenker and Bassett (1978), but only recently—because of computational advances—has it become practical for large data. With sufficient data, quantile regression can potentially describe the entire conditional distribution of the response. General linear models and generalized linear models are computationally less expensive, but the only aspect of the conditional distribution that they describe is the mean.

You should consider using quantile regression when the conditional distribution of the response varies with the predictors. Figure 15 illustrates data in which the conditional variance of the response (Var$[Y|X]$) increases with X.

Figure 15 Quantile Regression Models for Heteroscedastic Data



The three lines in Figure 15 represent quantile regression models for Y that correspond to the quantile levels 0.1, 0.5, and 0.9, or equivalently the 10th, 50th, and 90th percentiles. Fitting such models for a more extensive grid of quantile levels yields a description of the entire conditional distribution.

Table 4 summarizes the differences between least squares regression and quantile regression.

Table 4 Quantile Regression Compared with Least Squares Regression

Least Squares Regression	Quantile Regression
Predicts the conditional mean	Predicts conditional quantiles
Often assumes normality	Assumes no parametric distribution
Is sensitive to outliers	Is robust to outliers
Applies even with a small number of observations	Needs sufficient data
Is computationally inexpensive	Is computationally intensive

In many fields, such as financial risk management and fraud detection, important questions can be answered by modeling extreme percentiles of critical factors. Quantile regression can yield valuable insights that would not be readily obtained with standard regression methods. This is illustrated in the next example.

Example: Predicting Low and High Percentiles of Close Rates for Retail Stores

The examples on page 3 and page 9 show how you can use the REGSELECT and GENSELECT procedures to predict the average close rate for a store. This example shows how you can use the QTRSELECT procedure to predict low and high percentiles of close rates. Here the close rate for a store is considered to be low if it is less than the 10th percentile for stores that have the same combination of predictor values. Likewise, a close rate is considered to be high if it is greater than the 90th percentile for stores that have the same combination of predictor values.

The following statements use the QTRSELECT procedure to build regression models that predict the 10th and 90th percentiles, which correspond to the quantile levels 0.1 and 0.9:

```
proc qtrselect data=mycas.Stores;
   model Close_Rate = X1-X20 L1-L6 P1-P6 / quantile=0.1 0.9;
   selection method=forward(choose=aic) stophorizon=1 plots=all;
run;
```

Figure 16 visualizes the forward selection process for quantile level 0.1.

Figure 16 Coefficient Progression for Quantile Level 0.1

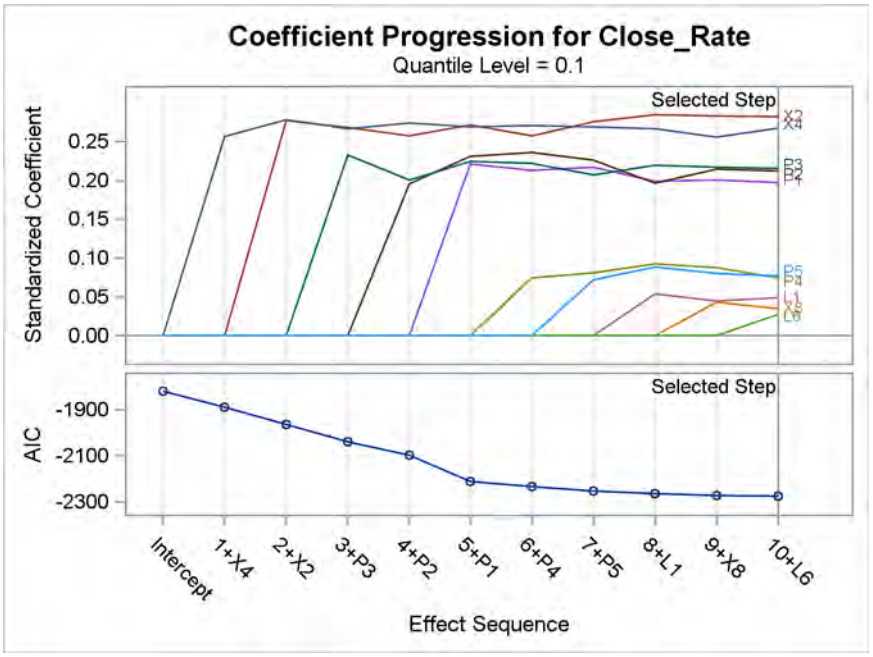


Figure 17 shows the fit statistics for the final model for quantile level 0.1.

Figure 17 Fit Statistics for Model Selected for Quantile Level 0.1

The QTRSELECT Procedure

Quantile Level = 0.1
Selected Model

Objective Function	50.27587
R1	0.37774
Adj R1	0.36501
AIC	-2275.08286
AICC	-2274.54187
SBC	-2228.72217
ACL	0.10055

Figure 18 shows the parameter estimates for the final model for quantile level 0.1.

Figure 18 Parameter Estimates for Model Selected for Quantile Level 0.1

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	59.98266	0.01989	3015.08	<.0001
X2	1	1.00585	0.02599	38.71	<.0001
X4	1	0.98467	0.02765	35.62	<.0001
X8	1	0.12872	0.02640	4.88	<.0001
L1	1	0.17769	0.02703	6.57	<.0001
L6	1	0.09756	0.02415	4.04	<.0001
P1	1	0.72146	0.02530	28.52	<.0001
P2	1	0.76697	0.02626	29.21	<.0001
P3	1	0.75435	0.02601	29.00	<.0001
P4	1	0.27254	0.02725	10.00	<.0001
P5	1	0.27256	0.02350	11.60	<.0001

The QTRSELECT procedure produces a distinct set of results for quantile level 0.9 because the corresponding check loss function is different from the check loss function for quantile level 0.1. Figure 19 shows the parameter estimates for the final model for quantile level 0.9.

Figure 19 Parameter Estimates for Model Selected for Quantile Level 0.9

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	60.26032	0.10430	577.77	<.0001
X2	1	1.30182	0.11886	10.95	<.0001
X4	1	0.81080	0.11156	7.27	<.0001
X14	1	-0.33776	0.11109	-3.04	0.0025
L1	1	0.61383	0.11502	5.34	<.0001
L2	1	0.91291	0.10822	8.44	<.0001
L3	1	0.93599	0.11590	8.08	<.0001
L4	1	0.63198	0.10882	5.81	<.0001
L5	1	0.61580	0.11252	5.47	<.0001
L6	1	0.51865	0.10089	5.14	<.0001
P4	1	0.70377	0.12686	5.55	<.0001

Two of the store characteristic variables (**X2** and **X4**) are selected in both the model for quantile level 0.1 and the model for quantile level 0.9. Five of the promotion variables (**P1–P5**) are selected in the model for level 0.1, but only one (**P4**) is selected in the model for level 0.9. All the layout variables (**L1–L6**) are selected in the model for level 0.9, but only two (**L1** and **L6**) are selected in the model for level 0.1. These results give you information about low- and high-performing stores that you would not obtain directly from least squares regression.

Comparison with the HPQUANTSELECT and QUANTSELECT Procedures

The functionality of the QTRSELECT procedure closely resembles that of the HPQUANTSELECT procedure, which is a SAS/STAT high-performance procedure. The QTRSELECT procedure is additionally capable of constructing complex effects (such as univariate spline and polynomial expansions) and producing plots that visualize the effect selection process.

Both the QTRSELECT procedure and the QUANTSELECT procedure in SAS/STAT fit and perform model selection for quantile regression models. The QTRSELECT procedure (but not the QUANTSELECT procedure) provides confidence limits and Wald tests for parameters and prediction limits for quantiles. The QUANTSELECT procedure (but not the QTRSELECT procedure) provides the lasso and adaptive lasso effect-selection methods and effect selection for quantile process regression. See Rodriguez and Yao (2017) for an analysis of the close rate data that uses the QUANTSELECT procedure.

Fitting Generalized Additive Models with the GAMMOD Procedure

The GAMMOD procedure fits generalized additive models that are based on low-rank regression splines (Wood 2006). Generalized additive models are extensions of generalized linear models. In addition to allowing linear predictors, they allow spline terms in order to capture nonlinear dependency that is either unknown or too complex to be characterized with a parametric effect such as a linear or quadratic term. Table 5 summarizes the components of a generalized additive model.

Table 5 Components of Generalized Additive Models

Component	Description
Linear predictor	Effects that involve continuous or classification variables
Nonparametric predictor	Spline terms that involve one or more continuous variables
Link function	Log, logit, log-log, complementary log-log, probit, reciprocal, reciprocal square
Distribution	Binary, binomial, gamma, inverse Gaussian, negative binomial, normal, Poisson, Tweedie

The GAMMOD procedure constructs spline terms by using 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 roughness of the spline curve.

Unlike the other procedures discussed in this paper, the GAMMOD procedure does not select variables or effects. Instead, it finds optimal models by automatically selecting smoothing parameters based on global model-evaluation criteria such as generalized cross validation (GCV) and unbiased risk estimation (UBRE).

Generalized additive models are useful for problems that involve unknown—possibly nonlinear—relationships between the response and the predictors, and relationships that can be assumed to be linear. Frigo and Osterloo (2016) describe a problem of this type in the context of insurance pricing.

In some situations, the spline fits that you obtain using PROC GAMMOD suggest parametric effects in a model that you can then fit with the GENSELECT procedure, as illustrated in the following example.

Example: Predicting Claim Rates for Loans

This example is drawn from the mortgage insurance industry, where analysts create models to predict conditional claim rates for specific types of loans. Understanding how claim rates depend on predictors is critical, because the model is used to assess risk and allocate funds for potential claims.

Claim rates for 10,000 mortgages are saved in a CAS table named **Claims**. The response variable **Rate** is the number of claims per 10,000 contracts in a policy year, and it is assumed to follow a Poisson distribution whose mean depends on the predictors listed in Table 6.

Table 6 Predictors for Claim Rate

Predictor	Description	Contribution
Age	Age of loan	Unknown, possibly quadratic
Price	Price of house	Unknown, nonlinear
RefInd	Indicator of a refinanced loan	Linear
PayIncmRatio	Payment-to-income ratio	Linear
RefInctvRatio	Refinance incentive ratio	Linear
UnempRate	Unemployment rate	Linear

In practice, models of this type involve many more predictors. A subset is used here for illustration.

The following statements use the GAMMOD procedure to fit an additive Poisson regression model for **Rate**:

```
proc gammod data=mycas.Claims plots=components;
   class RefInd;
   model Rate = param(RefInd PayIncmRatio RefInctvRatio UnempRate)
                spline(Age) spline(Price) / dist=poisson;
run;
```

The PARAM option requests parametric linear terms for **RefInd**, **PayIncmRatio**, **RefInctvRatio**, and **UnempRate**. The SPLINE options request spline effects for **Age** and **Price**.

Figure 20 displays information about the model fitting process. The Poisson mean of **Rate** is modeled by a log link function. The performance iteration algorithm (Gu and Wahba 1991) is used to obtain optimal smoothing parameters for the spline effects. The unbiased risk estimator (UBRE) criterion is used for model evaluation during the process of selecting smoothing parameters for the spline effects.

Figure 20 Model Information

The GAMMOD Procedure

Model Information	
Data Source	CLAIMS
Response Variable	Rate
Distribution	Poisson
Link Function	Log
Fitting Method	Performance Iteration
Fitting Criterion	UBRE
Optimization Technique for Smoothing	Newton-Raphson
Random Number Seed	1494796320

Figure 21 shows the fit statistics. You can use effective degrees of freedom to compare generalized additive models with generalized linear models, which do not involve spline terms. You can also use the information criteria, AIC, AICC, and BIC, for model comparisons, and you can request either the GCV criterion or the UBRE criterion for comparisons with other generalized additive models or penalized models.

Figure 21 Fit Statistics from the GAMMOD Procedure

Fit Statistics	
Penalized Log Likelihood	-26776
Roughness Penalty	7.83929
Effective Degrees of Freedom	16.54638
Effective Degrees of Freedom for Error	9982.63859
AIC (smaller is better)	53577
AICC (smaller is better)	53577
BIC (smaller is better)	53697
UBRE (smaller is better)	-0.00359

Figure 22 shows estimates for the parametric effects in the model.

Figure 22 Estimates for Parametric Terms

Parameter Estimates

Parameter	DF	Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept	1	2.484732	0.020876	14166.0978	<.0001
RefInd 0	1	-0.008894	0.005571	2.5488	0.1104
RefInd 1	0	0	.	.	.
PayIncmRatio	1	0.035731	0.009740	13.4582	0.0002
RefInctvRatio	1	-0.031308	0.009627	10.5765	0.0011
UnempRate	1	0.008047	0.002764	8.4763	0.0036

Figure 23 shows the effective degrees of freedom (DF) for the smoothing components of the model. The component for **Age** has a lower DF, indicating a more linear contribution than the contribution of the component for **Price**.

Figure 23 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(Age)	3.54638	35807.3	7.8393	9	10	24
Spline(Price)	8.00000	1.0000	1.3E-6	9	10	2000

Figure 24 shows tests for the existence of a contribution for each smoothing component. The results should be interpreted with caution because the tests do not take into account the process of selecting the smoothing parameter.

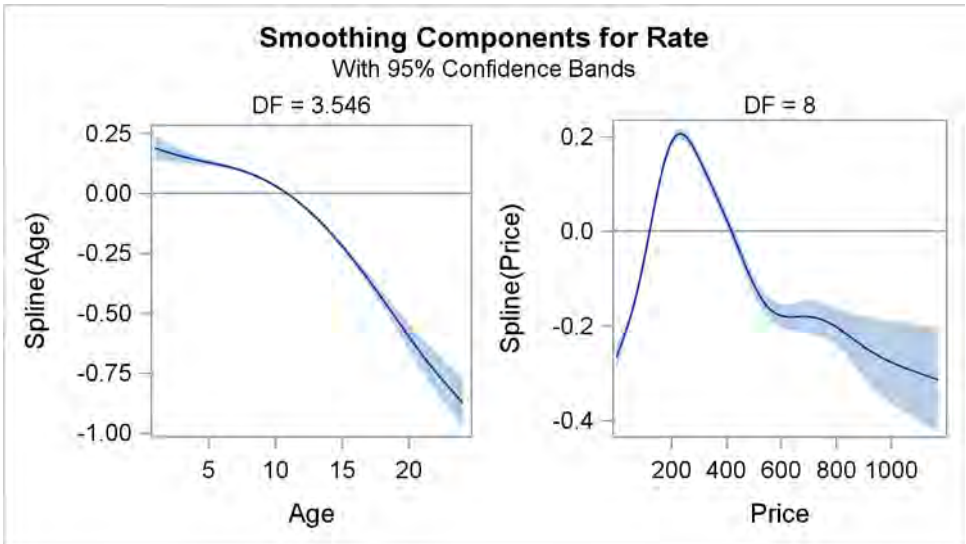
Figure 24 Tests for Smoothing Components

Tests for Smoothing Components

Component	Effective DF	Effective DF for Test	Chi-Square	Pr > ChiSq
Spline(Age)	3.54638	5	1685.6996	<.0001
Spline(Price)	8.00000	8	2844.2140	<.0001

Figure 25 displays plots of the components for **Age** and **Price**.

Figure 25 Spline Components for Age and Price



The plots suggest that quadratic effects might characterize the nonlinearity in **Age** and **Price**. The following statements incorporate these effects in a generalized linear model that is fitted with the GENSELECT procedure (you could also use the GENMOD procedure):

```
proc genselect data=mycas.Claims;
   class RefInd;
   model Rate = RefInd PayIncmRatio RefInctvRatio UnempRate
                Age Age*Age Price Price*Price / dist=poisson link=log;
run;
```

Fit statistics for the model that is fitted with PROC GENSELECT are shown in Figure 26.

Figure 26 Fit Statistics from the GENSELECT Procedure

The GENSELECT Procedure

Fit Statistics	
-2 Log Likelihood	54754
AIC (smaller is better)	54772
AICC (smaller is better)	54772
SBC (smaller is better)	54837

The SBC statistic is also referred to as the BIC statistic. The AIC, AICC, and BIC statistics in Figure 21 are smaller than the corresponding statistics in Figure 26, indicating that the generalized additive model produces a better fit.

Comparison with the GAMPL and GAM Procedures

The GAMMOD procedure resembles the GAMPL and GAM procedures in SAS/STAT; all three procedures fit generalized additive models. The results of the GAMMOD and GAMPL procedures should be very similar, but in general you should not expect similar results between these two procedures and the GAM procedure. Table 7 summarizes important design differences in these procedures.

Table 7 Comparison of the GAMMOD, GAMPL, and GAM Procedures

GAMMOD and GAMPL Procedures	GAM Procedure
Use low-rank regression splines for smoothers	Uses smoothing splines and loess for smoothers
Use performance iteration or outer iterations	Uses backfitting to fit models
Search for smoothing parameters by optimizing global criteria	Searches for smoothing parameters by fitting splines that have fixed degrees of freedom
Analyze large to massive data	Analyzes moderate to large data
Run in single-machine or distributed mode	Runs in single-machine mode
Use all cores and concurrent threads	Is single threaded

The GAMMOD procedure and the GENSELECT procedure use the log link function as the default for the gamma and the inverse Gaussian distributions. The GAMPL procedure uses the reciprocal link function as the default for the gamma distribution, and it uses the reciprocal squared link function as the default for the inverse Gaussian distribution.

Building Proportional Hazards Regression Models with the PHSELECT Procedure

Time-to-event models predict the probability that the lifetime of a subject exceeds t—denoted as the survivor function $S(t)$—from lifetime data that are incomplete because of censoring. These models are broadly applicable to data that range from patient survival times in medical research to customer lifetimes in business applications where turnover is a concern.

The PHSELECT procedure fits and builds Cox proportional hazards models. These models are semiparametric; they assume a linear parametric form for the effects of the predictors, but they do not require a parametric form for the

underlying survivor function. The survival time of each member of a population is assumed to follow its own hazard function, $\lambda_i(t)$, which is the instantaneous risk that an event will occur at time t and is expressed as

$$\lambda_i(t) = \lambda_0(t) \exp(\beta_1 x_{i1} + \cdots + \beta_p x_{ip}) \quad i = 1, \ldots, n$$

The function $\lambda_0(t)$ is called the baseline hazard function. The predictors $x_{i1}, \ldots, x_{ip}$ represent main effects that consist of continuous or classification variables and can include interaction effects or constructed effects of these variables.

The PHSELECT procedure uses the partial likelihood approach of Cox (1972, 1975) to estimate the coefficients $\beta_1, \ldots, \beta_p$. These estimates can then be used to predict $S(t)$ at specified times t for a new subject with specified covariates. The prediction is based on Breslow's estimator of the baseline cumulative hazard rate (Breslow 1974).

Like the other procedures discussed in this paper that build regression models, the PHSELECT procedure offers extensive capabilities for effect selection, including the backward, fast backward, forward, stepwise, and lasso methods, together with a wide variety of selection and stopping criteria for customizing the selection. The PHSELECT procedure also provides Cox regression diagnostics that are conditional on the selected model.

Example: Predicting the Retention of Insurance Customers

A health insurance company carries out a study of younger customers who are experiencing life changes. The goals are to identify factors that explain the risk of switching to a different insurance plan, and to predict the probability of retaining a customer from one to five years in the future.

The customer lifetimes are saved in a CAS table named **Customers**. Each observation provides information about a customer. The variables available for building a Cox regression model are **Time** (the customer lifetime, measured in months), **Status** (the censoring indicator variable), and the candidate predictors shown in Table 8.

Table 8 Candidate Predictors for Customer Lifetime Model

Predictor	Type
Age	Continuous
Area	Classification with levels `'Urban'`, `'Rural'`
CurrentPlan	Classification with levels `'A'`, `'B'`
Education	Continuous
Income	Continuous
LifeChange	Classification with levels `'Married'`, `'New Job'`, `'Child'`, `'None'`
Satisfaction	Classification with levels `'Excellent'`, `'Good'`, `'Poor'`
Smoking	Classification with levels `'Yes'`, `'No'`, `'Quit'`

The following statements build a Cox model by using the forward selection method:

```
proc phselect data=mycas.Customers;
   class Area(ref='Urban') CurrentPlan(ref='B') LifeChange(ref='None')
         Satisfaction(ref='Poor') Smoking (ref='No') / param=ref;
   model Time*Status(0) = Age Area CurrentPlan Education Income LifeChange
                          Satisfaction Smoking;
   selection method=forward(select=sbc stop=sbc) plots=all ;
   code file='ScoreCode.sas' timepoint=12 24 36 48 60;
run;
```

The CLASS statement specifies that **Area**, **CurrentPlan**, **LifeChange**, **Satisfaction**, and **Smoking** are classification variables, and the PARAM= option specifies reference cell coding for these variables. The REF= option specifies the reference level of each variable. Various parameterizations are available, including effect coding and less-than-full-rank reference (GLM) coding, which is the default.

The SELECTION statement requests the forward method, and the SELECT= option requests that the selected model minimize the Schwarz Bayesian criterion (SBC). By default, all the design columns that correspond to a classification variable enter the model together. If you specify the SPLIT option in the CLASS statement, the design columns will enter independently.

The CODE statement writes SAS DATA step code for computing predicted survival probabilities to a file named *ScoreCode.sas*. The TIMEPOINT= option specifies time points (in months) at which survival probabilities are to be predicted from the selected model.

Effects that provide the best improvement in SBC are added until no more effects can improve the criterion. As shown in Figure 27, the minimum value of SBC is reached when **Area** enters the model.

Figure 27 Model Selection Summary

The PHSELECT Procedure

Selection Details

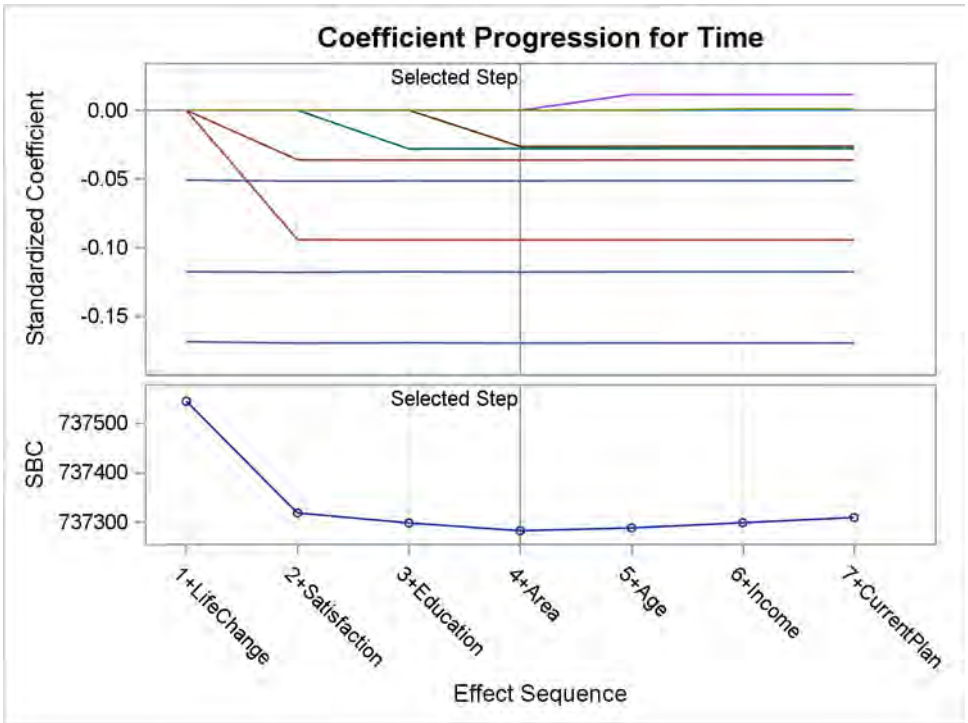
Selection Summary

Step	Effect Entered	Number Effects In	SBC
1	LifeChange	1	737543.656
2	Satisfaction	2	737318.101
3	Education	3	737297.864
4	Area	4	737282.214*
5	Age	5	737287.812
6	Income	6	737298.311
7	CurrentPlan	7	737308.843

*** Optimal Value Of Criterion**

The plot in Figure 28 visualizes the selection process. The forward method selects a model with seven parameters that involve one continuous variable and three classification variables.

Figure 28 Coefficient Progression with Forward Selection



The parameter estimates are shown in Figure 29. Effects that have negative coefficients decrease the predicted hazard function at time t and therefore increase the predicted survival time $\hat{S}(t \mid x_1, \ldots, x_p)$ because

$$\hat{S}(t \mid x_1, \ldots, x_p) = \exp(-\hat{\Lambda}_0(t) \exp(\hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p))$$

where $\hat{\Lambda}_0(t)$ is the Breslow estimator of the cumulative baseline hazard rate $\Lambda_0(t) = \int_0^t \lambda_0(u)du$.

Figure 29 Parameter Estimates for Selected Model

Parameter Estimates

Parameter	DF	Estimate	Standard Error	Chi-Square	Pr > ChiSq
Area Rural	1	-0.052843	0.010327	26.1831	<.0001
Education	1	-0.008096	0.001496	29.2890	<.0001
LifeChange Child	1	-0.391075	0.014684	709.2694	<.0001
LifeChange Married	1	-0.273049	0.014655	347.1629	<.0001
LifeChange New Job	1	-0.119519	0.014620	66.8349	<.0001
Satisfaction Excellent	1	-0.200931	0.012690	250.7106	<.0001
Satisfaction Good	1	-0.077127	0.012630	37.2929	<.0001

Predicting Survival Probabilities

The following statements use the generated code to compute retention probabilities for five new customers whose covariates are saved in a data set named **NewCustomers**:

```
data Predictions;
   set NewCustomers;
   %include 'ScoreCode.sas';
run;
```

Figure 30 lists the predicted retention probabilities in **NewCustomers**.

Figure 30 Listing of **Scores**

Area	Years of Education	Life Change	Satisfaction	Retention Probability at 1 Year	Retention Probability at 2 Years	Retention Probability at 3 Years	Retention Probability at 4 Years	Retention Probability at 5 Years
Rural	13	New Job	Poor	0.671	0.455	0.315	0.221	0.155
Urban	14	Married	Good	0.718	0.520	0.383	0.285	0.212
Rural	8	New Job	Excellent	0.711	0.512	0.373	0.276	0.204
Urban	11	New Job	Poor	0.652	0.431	0.290	0.198	0.136
Rural	17	Child	Excellent	0.786	0.622	0.498	0.402	0.324

Comparison with the PHREG Procedure

Compared with the PHREG procedure in SAS/STAT, the PHSELECT procedure provides many more features for model selection, including the lasso method and options for selection and stopping that are based on information criteria and validation. The PHSELECT procedure also enables you to partition the data into logical subsets for training, validation, and testing. On the other hand, the PHREG procedure provides more flexibility for fitting the Cox model, such as the ability to specify time-dependent covariates, and more inferential methods, such as the following:

- extensive postfitting analyses of regression parameters
- hazard ratios for any variable in the model at customized settings, and confidence limits for hazard ratios
- Schemper-Henderson and concordance statistics for model assessment
- time-dependent ROC curves for model assessment

The PHREG procedure also enables you to analyze competing risks and frailty models, and to perform Bayesian analysis of Cox models, piecewise exponential models, and frailty models.

Using the LOGSELECT Procedure with Discrete Time

In some applications of proportional hazards regression, events occur at regular, discrete points in time, or ties occur because continuous event times are grouped into intervals. Many ties present a problem for the Breslow partial likelihood approach that is implemented by the PHSELECT procedure, but this problem can be circumvented by using logistic regression and maximum likelihood methods that are implemented in the LOGSELECT procedure.

As explained by Allison (2010, chap. 7), the maximum likelihood approach treats the survival history of each individual as a series of distinct observations, one at each time unit. The observations are pooled, and a logistic regression model is used to predict the probability P_{it} that individual i has an event at time t. This model is of the form

$$\log\left(\frac{P_{it}}{1-P_{it}}\right) = \alpha_t + \beta_1 x_{it1} + \cdots + \beta_p x_{itp}, \quad i = 1,\ldots,n, \quad t = 1,2,\ldots$$

The maximum likelihood approach provides estimates of the effect of time on the hazard function, and it handles time-dependent covariates in a natural fashion. The total number of observations obtained by expanding individual survival histories can be large, especially if the time units are small, but keep in mind that statistical procedures in SAS Viya are designed to accommodate large data.

Summary of Benefits

Table 9 lists key benefits of methods that the SAS Viya procedures implement for regression modeling.

Table 9 Benefits of Methods for Regression Modeling

Method	Benefit	SAS Viya Procedure
Stepwise methods that use modern information and validation criteria for selection and stopping	Improved predictive accuracy and interpretability	GENSELECT, LOGSELECT, PHSELECT, QTRSELECT, REGSELECT
Lasso methods	Improved predictive accuracy and interpretability	GENSELECT, LOGSELECT, PHSELECT, REGSELECT
Data partitioning into training, validation, and testing roles	Improved predictive accuracy	GENSELECT, LOGSELECT, PHSELECT, QTRSELECT, REGSELECT
Effect selection for generalized linear models	Models for responses with a variety of discrete and continuous distributions	GENSELECT
Effect selection for time-to-event models	Prediction of survival probabilities by using censored lifetime data	PHSELECT
Effect selection for quantile regression	Models for conditional quantiles of a continuous response distribution	QTRSELECT
Generalized additive models with penalization	Flexibility for modeling complex, unknown dependency relationships	GAMMOD

No one method consistently outperforms the others. Furthermore, all the methods involve choices of tuning parameters and optimization techniques for which there are no universally best defaults. In order to decide which methods are appropriate for your work, you should understand their assumptions and characteristics; these are explained in the "Shared Concepts" and procedure chapters in *SAS Visual Statistics 8.2: Procedures*. You should also experiment with different combinations of options to learn about their behavior.

The ability to score future data is an essential aspect of predictive modeling. All the procedures discussed in this paper compute predicted values for observations in which only the response variable is missing (or only the censoring variable is missing in the case of the PHSELECT procedure). The values are saved in a CAS table that is created when you specify the OUTPUT statement. Except for PROC GAMMOD, all the procedures provide a CODE statement, which writes SAS DATA step code for computing predicted values to a file, to a catalog entry, or to a CAS table.

The regression modeling procedures in SAS Viya offer two general improvements in functionality over the SAS/STAT high-performance procedures that they succeed:

- capability for producing graphs with the PLOTS option
- capability for constructing special collections of columns for design matrices with the EFFECT statement (for example, you can use the EFFECT statement to specify polynomial and spline effects).

REFERENCES

Allison, P. D. (2010). *Survival Analysis Using the SAS System: A Practical Guide*. 2nd ed. Cary, NC: SAS Institute Inc.

Breslow, N. E. (1974). "Covariance Analysis of Censored Survival Data." *Biometrics* 30:89–99.

Cox, D. R. (1972). "Regression Models and Life-Tables." *Journal of the Royal Statistical Society, Series B* 34:187–220. With discussion.

Cox, D. R. (1975). "Partial Likelihood." *Biometrika* 62:269–276.

Efron, B., Hastie, T. J., Johnstone, I. M., and Tibshirani, R. (2004). "Least Angle Regression." *Annals of Statistics* 32:407–499. With discussion.

Frigo, C., and Osterloo, K. (2016). "exSPLINE That: Explaining Geographic Variation in Insurance Pricing." In *Proceedings of the SAS Global Forum 2016 Conference*. Cary, NC: SAS Institute Inc. http://support.sas.com/resources/papers/proceedings16/8441-2016.pdf.

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.

Koenker, R., and Bassett, G. W. (1978). "Regression Quantiles." *Econometrica* 46:33–50.

Rodriguez, R. N. (2016). "Statistical Model Building for Large, Complex Data: Five New Directions in SAS/STAT Software." In *Proceedings of the SAS Global Forum 2016 Conference*. Cary, NC: SAS Institute Inc. http://support.sas.com/resources/papers/proceedings16/SAS4900-2016.pdf.

Rodriguez, R. N., and Yao, Y. (2017). "Five Things You Should Know about Quantile Regression." In *Proceedings of the SAS Global Forum 2017 Conference*. Cary, NC: SAS Institute Inc. http://support.sas.com/resources/papers/proceedings17/SAS0525-2017.pdf.

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

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

Acknowledgments

The following statistical software developers at SAS contributed to this paper: Robert Cohen, Bob Derr, Gordon Johnston, Ying So, and Yonggang Yao. The authors also thank Anne Baxter for editorial assistance.

Contact Information

Your comments and questions are valued and encouraged. You can contact the authors at the following address:

Weijie Cai
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
Weijie.Cai@sas.com