Paper SAS4900-2016

Statistical Model Building for Large, Complex Data: Five New Directions in SAS/STAT® Software

Robert N. Rodriguez, SAS Institute Inc.

Abstract

The increasing size and complexity of data in research and business applications require a more versatile set of tools for building explanatory and predictive statistical models. In response to this need, SAS/STAT® software continues to add new methods.

This paper provides a high-level tour of five modern approaches to model building that are available in recent releases of SAS/STAT: building sparse regression models with the GLMSELECT procedure, building generalized linear models with the HPGENSELECT procedure, building quantile regression models with the QUANTSELECT procedure, fitting generalized additive models with the GAMPL procedure, and building classification and regression trees with the HPSPLIT procedure. The paper reviews the key concepts of each approach and illustrates the syntax and output of each procedure with a basic example.

Introduction

One of the most frequently asked questions in statistical practice is the following: "I have hundreds of variables—even thousands. Which should I include in my regression model?" This paper presents overviews of five modern approaches to selecting the effects in a regression model when you need a model that is interpretable or that accurately predicts future data. 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.

The paper is organized into five main sections, one for each approach:

- Building Sparse Regression Models with the GLMSELECT Procedure
- Building Generalized Linear Models with the HPGENSELECT Procedure
- Building Quantile Regression Models with the QUANTSELECT Procedure
- Fitting Generalized Additive Models with the GAMPL Procedure
- Building Classification and Regression Tree Models with the HPSPLIT Procedure

These approaches are implemented in new or enhanced procedures that are available in recent releases of SAS/STAT software. The paper introduces each procedure, explains key concepts, and illustrates syntax and output with a basic example.

SAS has accelerated the pace of SAS/STAT releases in order to meet customer requirements for versatile statistical methods that are driven by data needs and by advances in methodology. SAS/STAT 14.1, the current production release, is the fifth release of SAS/STAT software during the past four years. As indicated in Table 1, these releases have their own numbering scheme, because they occur more frequently than new versions of Base SAS®.

Table 1 Recent Releases of SAS/STAT Software

Release	Year	Overview Paper	Base SAS Version
SAS/STAT 12.1	2012	Stokes et al. (2012)	SAS 9.3
SAS/STAT 12.3	2013	Stokes (2013)	SAS 9.4
SAS/STAT 13.1	2013	Rodriguez (2014)	SAS 9.4M1
SAS/STAT 13.2	2014	Stokes and Statistical R&D Staff (2015)	SAS 9.4M2
SAS/STAT 14.1	2015	Stokes and Statistical R&D Staff (2015)	SAS 9.4M3

Building Sparse Regression Models with the GLMSELECT Procedure

The GLMSELECT procedure selects effects in 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 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 GLMSELECT procedure supports the model selection methods summarized in Table 2.

Table 2 Effect Selection Methods in the GLMSELECT Procedure

Method	Description
Forward selection	Starts with no effects and adds effects
Backward elimination	Starts with all effects and deletes effects
Stepwise selection	Starts with no effects; effects are added and can be deleted
Least angle regression	Starts with no effects and adds effects; at each step, estimated βs are shrunk toward 0
Lasso	Constrains sum of absolute βs; some βs set to 0
Elastic net	Constrains sums of absolute and squared βs; some βs set to 0
Adaptive lasso	Constrains sum of absolute weighted βs; some βs set to 0
Group lasso	Constrains sum of Euclidean norms of βs corresponding to effects; all βs for the same effect are set to 0 or are non-zero

Forward selection, backward elimination, and stepwise regression reduce the number of effects in the model. In contrast, the lasso, elastic net, adaptive lasso, and group lasso methods are based on regularization. These methods leave all the effects in the model, but they restrict their parameters by setting some to zero while shrinking others toward zero.

Whereas the classical regression estimator solves the least squares problem

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

the lasso estimator solves the least squares problem by placing an ℓ_1 penalty on the parameters:

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

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

Provided that the lasso parameter t is small enough, some of the regression coefficients will be exactly zero. Increasing t in discrete steps leads to a sequence of regression coefficients, where the nonzero coefficients at each step correspond to selected parameters. Thus the lasso method produces sparser and potentially more interpretable models than traditional methods such as forward selection. The following example illustrates this distinction.

Example: Predicting the 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 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 data set 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 illustrative purposes.

Results with the Forward Selection Method

The following statements use the GLMSELECT procedure to build a model with the forward selection method:

```
proc glmselect plots=coefficient data=Stores;
   model Close_Rate = X1-X20 L1-L6 P1-P6 / selection=forward(choose=aic);
run;
```

The SELECTION= option requests the forward method, and the CHOOSE= suboption specifies that the selected model minimize Akaike's information criterion (AIC). The settings for the selection process are listed in Figure 1.

Figure 1 Model Information

The GLMSELECT Procedure

Data Set	WORK.STORES
Dependent Variable	Close_Rate
Selection Method	Forward
Select Criterion	SBC
Stop Criterion	SBC
Choose Criterion	AIC
Effect Hierarchy Enforced	None

At each step of the forward selection process, AIC is evaluated, and the model that yields the minimal value of AIC is chosen. By default, the GLMSELECT procedure uses the Schwarz Bayesian information criterion (SBC) as the select criterion for determining the order in which effects enter at each step. The effect that is selected is the effect 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 with a larger value of SBC. Both AIC and SBC guard against overfitting by penalizing the model for having a large number of parameters.

As shown in Figure 2, the minimum value of AIC is reached at Step 9, when **P1** enters the model.

Figure 2 Selection Summary with Forward Selection

The GLMSELECT Procedure

Forward Selection Summary

Step	Effect Entered	Number Effects In	AIC	SBC
0	**Intercept**	1	545.6009	47.8155
1	**X2**	2	466.3833	-27.1875
2	**X4**	3	436.8566	-52.4996
3	**P3**	4	424.5035	-60.6381
4	**P4**	5	413.4923	-67.4347
5	**L1**	6	402.9892	-73.7232
6	**L3**	7	393.1296	-79.3681
7	**P5**	8	385.0985	-83.1847
8	**L2**	9	377.8229	-86.2457
9	**P1**	10	371.2472*	-88.6068*

*** Optimal Value of Criterion**

The coefficient progression plot in Figure 3, requested using the PLOTS= option, visualizes the selection process.

Figure 3 Coefficient Progression with Forward Selection

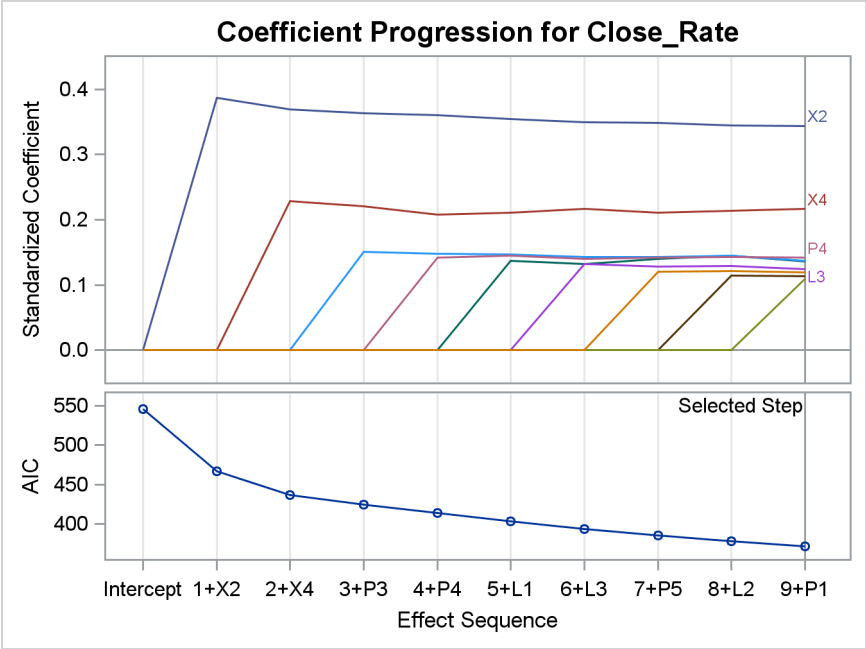


Figure 4 shows the parameter estimates for the final model. The estimates for **X2** and **X4** are larger than the estimates for the seven other predictors, and all the standard errors are comparable in size.

Figure 4 Parameter Estimates with Forward Selection

Parameter Estimates

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

Results with the Lasso Method

The following statements build a model with the lasso method:

```
proc glmselect plots=coefficient data=Stores;
   model Close_Rate = X1-X20 L1-L6 P1-P6 / selection=lasso(choose=aic);
run;
```

The settings for the selection process are listed in Figure 5. As with the settings for the forward method in Figure 1, the choose criterion is AIC and the stop criterion is SBC. However, for the lasso method the GLMSELECT 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.

Figure 5 Model Information

The GLMSELECT Procedure

Data Set	WORK.STORES
Dependent Variable	Close_Rate
Selection Method	LASSO
Stop Criterion	SBC
Choose Criterion	AIC
Effect Hierarchy Enforced	None

In contrast to the forward method, which selects a model with nine variables, the lasso method selects a sparse model with two variables, **X2** and **X4**, as shown in Figure 6 and Figure 7.

Figure 6 Selection Summary with Lasso

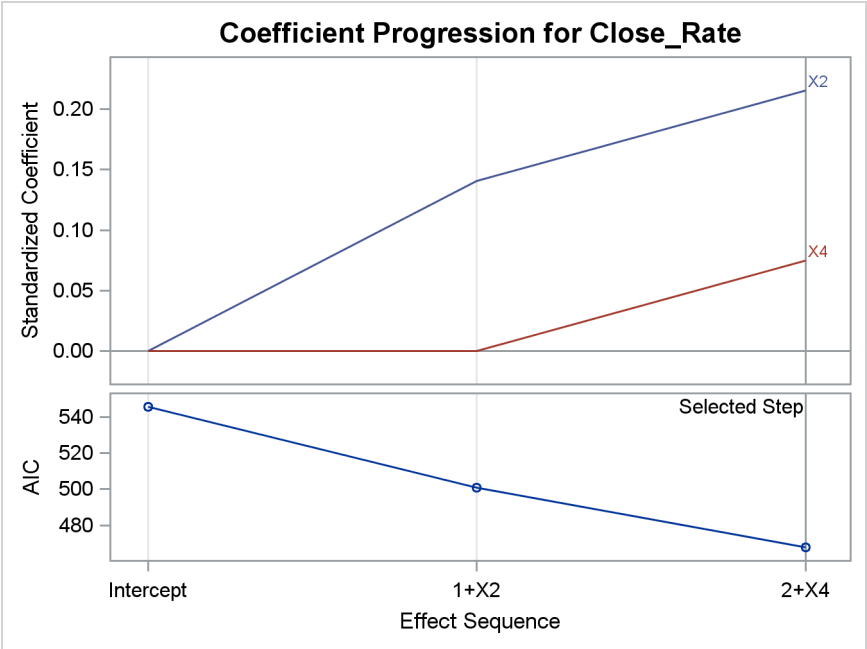
The GLMSELECT Procedure

LASSO Selection Summary

Step	Effect Entered	Effect Removed	Number Effects In	AIC	SBC
0	**Intercept**		1	545.6009	47.8155
1	**X2**		2	500.9692	7.3984
2	**X4**		3	467.7680*	-21.5882*

*** Optimal Value of Criterion**

Figure 7 Coefficient Progression with Lasso



The parameter estimates for the sparse model are shown in Figure 8. Note that these estimates are closer to zero than the corresponding estimates in Figure 4.

Figure 8 Parameter Estimates with Lasso

Parameter Estimates		
Parameter	**DF**	**Estimate**
Intercept	1	61.089916
X2	1	0.767684
X4	1	0.276289

The Elastic Net Method

The elastic net method is a generalization of the lasso method that estimates regression coefficients by solving the doubly penalized least squares problem:

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

$$\text{subject to } \sum_{j=1}^{p} |\beta_j| \le t_1 \text{ and } \sum_{j=1}^{p} \beta_j^2 \le t_2$$

In other words, the elastic net method balances between the ℓ_1 lasso penalty and the ℓ_2 penalty for ridge regression. If t_1 is a large value, the elastic net method reduces to ridge regression. If t_2 is a large value, the elastic net method reduces to the lasso method.

The elastic net method offers advantages over the lasso method in three situations (Zou and Hastie 2005; Hastie, Tibshirani, and Wainwright 2015):

- The elastic net method can select more than n variables when the number of parameters p exceeds n. The lasso method can select at most n variables.
- The elastic net method can achieve better prediction when the predictors are highly correlated and $n > p$.
- The elastic net method can handle groups of highly correlated variables more effectively. For an illustration, see Hastie, Tibshirani, and Wainwright (2015, chap. 4).

The following statements use the elastic net method to build a model for **Close_Rate**:

```
proc glmselect plots=coefficient data=Stores;
   model Close_Rate = X1-X20 L1-L6 P1-P6 / selection=elasticnet(choose=aic);
run;
```

In this example, the predictors are not highly correlated, and the selected model (not shown) is identical to the model that is selected with the lasso method.

Other Recent Enhancements

To address the computational demands of model selection when you have a very large number of effects, the GLMSELECT procedure has added screening approaches that you can combine with variable selection methods to reduce the number of regressors to a smaller subset on which the selection is performed.

The procedure provides the SASVI safe screening method proposed by Liu et al. (2014), for which the resulting solution is the same as the solution when no screening is performed. The procedure also provides sure independence screening, proposed by Fan and Lv (2008), a heuristic method that is faster but is not guaranteed to reproduce the true lasso or elastic net solution.

The GLMSELECT procedure has also added the group lasso selection method (Yuan and Lin 2006), which requires groups of parameters to enter the model together. This method is especially useful when the model includes classification effects or spline effects.

For more information, see the chapter on the GLMSELECT procedure in the *SAS/STAT 14.1 User's Guide*.

Building Generalized Linear Models with the HPGENSELECT Procedure

The HPGENSELECT procedure provides model fitting and model building for generalized linear models. It fits models with standard response distributions in the exponential family, such as the normal, Poisson, and Tweedie distributions. In addition, PROC HPGENSELECT fits multinomial models for ordinal and unordered multinomial responses, and it fits zero-inflated Poisson and negative binomial models for count data. For all these models, the HPGENSELECT procedure provides forward, backward, stepwise, and lasso variable selection. The procedure estimates the parameters of a generalized linear model by using maximum likelihood techniques.

Generalized linear models offer versatility for analyzing 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).

Table 3 summarizes these three components.

Table 3 Components of Generalized Linear Models

Component	Description
Linear predictor	Effects involving continuous or classification variables
Link function	Log, logit, inverse, and so on
Distribution	Normal, binomial, Poisson, gamma, Tweedie, and so on

What Is the Difference between the HPGENSELECT and GENMOD Procedures?

Both PROC HPGENSELECT and PROC GENMOD fit generalized linear models. However, there are important design differences in the statistical capabilities of these procedures, as summarized in Table 4.

Table 4 Comparison of PROC HPGENSELECT and PROC GENMOD

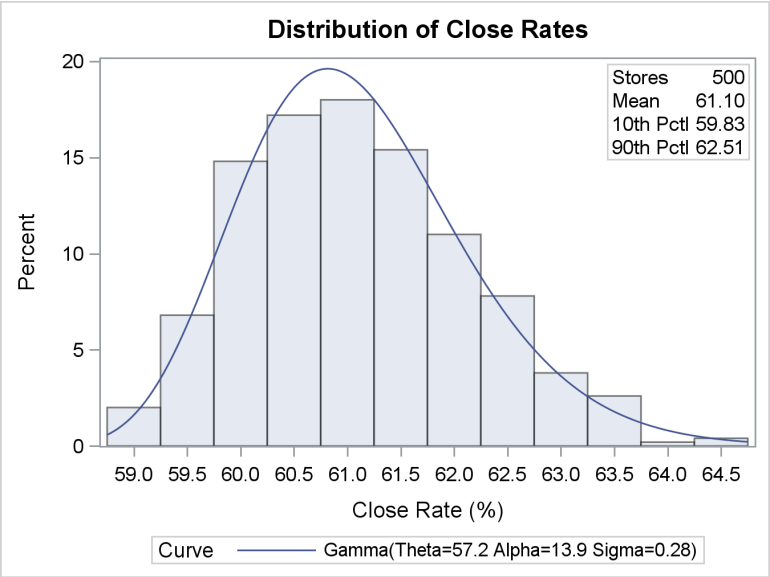
HPGENSELECT Procedure	GENMOD Procedure
Fits and builds generalized linear models	Fits generalized linear models
Analyzes large to massive data	Analyzes moderate to large data
Designed for predictive modeling	Designed for inferential analysis
Runs in single-machine or distributed mode	Runs in single-machine mode

PROC HPGENSELECT is referred to as a high-performance procedure, because it runs in either single-machine mode or distributed mode. For a discussion of these modes, see Cohen and Rodriguez (2013) and Johnston and Rodriguez (2015).

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

Figure 9 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 is worth exploring.

Figure 9 Distribution of Close Rates for 500 Stores



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

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

proc hpgenselect data=Stores;
   model Close_Rate_0 = X1-X20 L1-L6 P1-P6 / distribution = gamma;
   selection method=forward(choose=aic);
run;
```

The METHOD= option requests the forward selection method, and the CHOOSE= suboption specifies that the selected model minimize Akaike's information criterion.

Results with the Forward Selection Method

The settings for the selection process are listed in Figure 10.

Figure 10 Selection Information with Forward Method

The HPGENSELECT Procedure

Selection Information	
Selection Method	Forward
Select Criterion	Significance Level
Stop Criterion	Significance Level
Choose Criterion	AIC
Effect Hierarchy Enforced	None
Entry Significance Level (SLE)	0.05
Stop Horizon	1

Figure 11 shows that the minimum value of AIC is reached at Step 10, when **L5** enters the model. Note that the selected variables are the same as those selected by the GLMSELECT procedure with the forward method (see Figure 2), with the addition of **L5**.

Figure 11 Selection Summary with Forward Method

The HPGENSELECT Procedure

Selection Summary

Step	Effect Entered	Number Effects In	AIC	p Value
0	Intercept	1	1448.2155	.
1	X2	2	1372.9559	<.0001
2	X4	3	1345.6873	<.0001
3	P3	4	1333.3930	0.0002
4	L3	5	1322.5714	0.0004
5	P4	6	1312.2416	0.0005
6	L1	7	1304.9794	0.0025
7	P5	8	1297.9234	0.0027
8	L2	9	1291.8963	0.0048
9	P1	10	1286.2800	0.0061
10	L5	11	1282.0650*	0.0129

*** Optimal Value of Criterion**

Figure 12 shows the fit statistics for the selected model.

Figure 12 Fit Statistics for Gamma Regression Model Selected with Forward Method

Fit Statistics	
-2 Log Likelihood	1258.06
AIC (smaller is better)	1282.06
AICC (smaller is better)	1282.71
BIC (smaller is better)	1332.64
Pearson Chi-Square	41.4567
Pearson Chi-Square/DF	0.08478

Figure 13 shows the parameter estimates for the selected model. As in Figure 4, the estimates for **X2** and **X4** are larger in magnitude than the estimates for the other predictors.

Figure 13 Parameter Estimates for Gamma Regression Model Selected with Forward Method

Parameter Estimates

Parameter	DF	Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept	1	0.421938	0.015141	776.6306	<.0001
X2	1	-0.129234	0.014444	80.0555	<.0001
X4	1	-0.083540	0.014834	31.7168	<.0001
L1	1	-0.048919	0.014309	11.6878	0.0006
L2	1	-0.035614	0.013278	7.1939	0.0073
L3	1	-0.049864	0.013921	12.8299	0.0003
L5	1	-0.034887	0.013950	6.2544	0.0124
P1	1	-0.040273	0.014554	7.6575	0.0057
P3	1	-0.049916	0.013947	12.8092	0.0003
P4	1	-0.051448	0.014473	12.6367	0.0004
P5	1	-0.039721	0.013947	8.1112	0.0044
Dispersion	1	12.053493	0.752016	.	.

Results with the Lasso Method

The following statements build a gamma regression model with the lasso method:

```
proc hpgenselect data=Stores;
   model Close_Rate_0 = X1-X20 L1-L6 P1-P6 / distribution = gamma;
   selection method=lasso(choose=aic);
run;
```

The lasso again selects a sparse model with two variables, **X2** and **X4**. The regularization parameter that minimizes AIC is shown in Figure 14.

Figure 14 Lasso Regularization Parameter

The HPGENSELECT Procedure

Maximum Regularization Parameter	0.118143
Chosen Regularization Parameter	0.060489

The lasso estimates for **X2** and **X4** in Figure 15 are shrunk toward zero, compared with the estimates in Figure 13.

Figure 15 Parameter Estimates for Gamma Regression Model Selected with Lasso Method

Parameter Estimates

Parameter	DF	Estimate
Intercept	1	0.324793
X2	1	-0.069242
X4	1	-0.014639
Dispersion	0	1.000000

Building Quantile Regression Models with the QUANTSELECT Procedure

The QUANTSELECT procedure performs effect selection in the framework of quantile regression, which models the quantiles (percentiles) of a response variable conditional on covariates. Quantile regression models, introduced by Koenker and Bassett (1978), can potentially describe the entire conditional distribution of the response. By comparison, general linear models and generalized linear models describe only the conditional mean of the response but are computationally less expensive.

Quantile regression does not assume a particular distribution for the response, nor does it assume a constant variance for the response, unlike ordinary least squares regression. Figure 16 illustrates data in which the variance of the response **Y** increases with the covariate **X**. Simple linear regression models the conditional mean $E[Y|X]$, but it does not capture the conditional variance $\mathrm{Var}[Y|X]$.

Figure 16 Variance in Y Increases with X

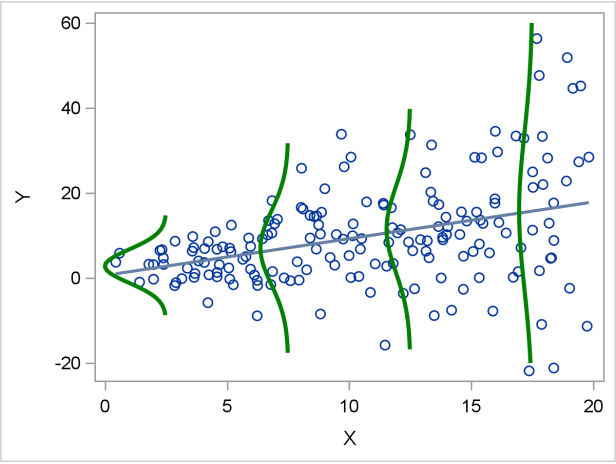
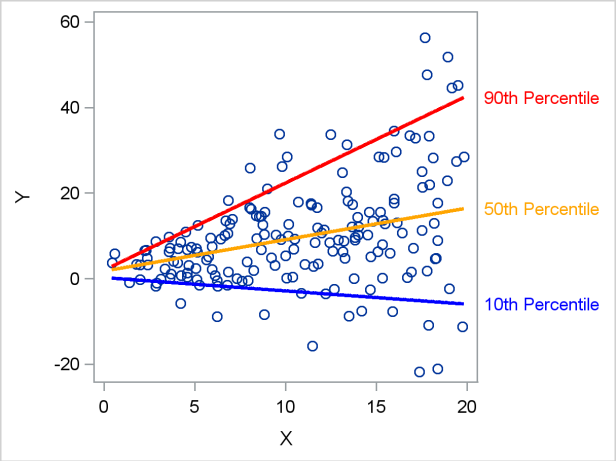


Figure 17 shows quantile regression lines for the 10th, 50th, and 90th conditional percentiles of **Y**. These are formally referred to as the quantile regression lines that correspond to the quantile levels 0.10, 0.50, and 0.90.

Figure 17 Regression Models for Three Percentiles



Fitting a Quantile Regression Model

The regression model for quantile level τ is

$$Q_\tau(Y|X) = X\beta(\tau), \quad 0 < \tau < 1$$

where $\beta(\tau)$ is estimated by solving the minimization problem

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

and $\rho_\tau(r) = \tau \max(r, 0) + (1 - \tau) \max(-r, 0)$. The function $\rho_\tau(r)$ is referred to as the check loss, because its shape resembles a check mark.

For each quantile level τ, the solution to the minimization problem yields a distinct set of regression coefficients. Note that $\tau = 0.5$ corresponds to median regression, and $2\rho_{0.5}(r)$ is the absolute value function.

Using the QUANTSELECT Procedure

The QUANTSELECT procedure fits and builds quantile regression models. It is designed primarily as an effect selection procedure and does not include regression diagnostics and hypothesis testing, which are provided by the QUANTREG procedure.

The QUANTSELECT procedure supports the model selection methods summarized in Table 5.

Table 5 Effect Selection Methods in the QUANTSELECT Procedure

Method	Description
Forward selection	Starts with no effects and adds effects
Backward elimination	Starts with all effects and deletes effects
Stepwise selection	Starts with no effects; effects are added and can be deleted
Lasso	Adds and deletes effects based on a constrained version of estimated check risk where the ℓ_1 norm of the βs is penalized
Adaptive lasso	Constrains sum of absolute weighted βs; some βs set to 0

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

The examples in the preceding sections show how you can build a standard regression model and a gamma regression model for the close rate data. These models answer the following questions:

How can I predict the close rate for a new store?

Which variables explain the average close rate of a store?

By building a quantile regression model, you can answer a different question:

Are there variables that differentiate low and high close rates?

The following statements use the QUANTSELECT procedure to build quantile regression models for levels 0.1, 0.5, and 0.9:

```
proc quantselect data=Stores plots=Coefficients seed=15531;
   model Close_Rate = X1-X20 L1-L6 P1-P6 / quantile = 0.1 0.5 0.9
                                           selection=lasso(sh=3);
   partition fraction(validate=0.3);
run;
```

The SELECTION= option specifies the lasso method with a stop horizon of 3. The PARTITION statement reserves 30% of the data for validation, leaving the remaining 70% for training.

Figure 18 summarizes the effect selection process for quantile level 0.1. The lasso method generates a sequence of candidate models, and the process chooses the model that minimizes the average validation check loss (ACL). The process stops at Step 14.

Figure 18 Selection Summary for Quantile Level 0.1

The QUANTSELECT Procedure
Quantile Level = 0.1

Selection Summary

Step	Effect Entered	Effect Removed	Number Effects In	Validation ACL
0	Intercept		1	0.1578
1	X2		2	0.1667
2	X4		3	0.1566
3	P3		4	0.1380
4	P1		5	0.1326
5	P2		6	0.1119
6	P4		7	0.1104
7	X20		8	0.1113
8	X3		9	0.1111
9	P5		10	0.1096
10		P5	9	0.1111
11	P5		10	0.1096
12		X3	9	0.1083*
13	L1		10	0.1105
14	X3		11	0.1117

The coefficient progression plot in Figure 19 visualizes the selection process, and it is similar to the coefficient progression plot that is constructed by the GLMSELECT procedure in Figure 3. In both plots, **X2** and **X4** are the first two variables that enter the model.

Figure 19 Coefficient Progression for Quantile Level 0.1

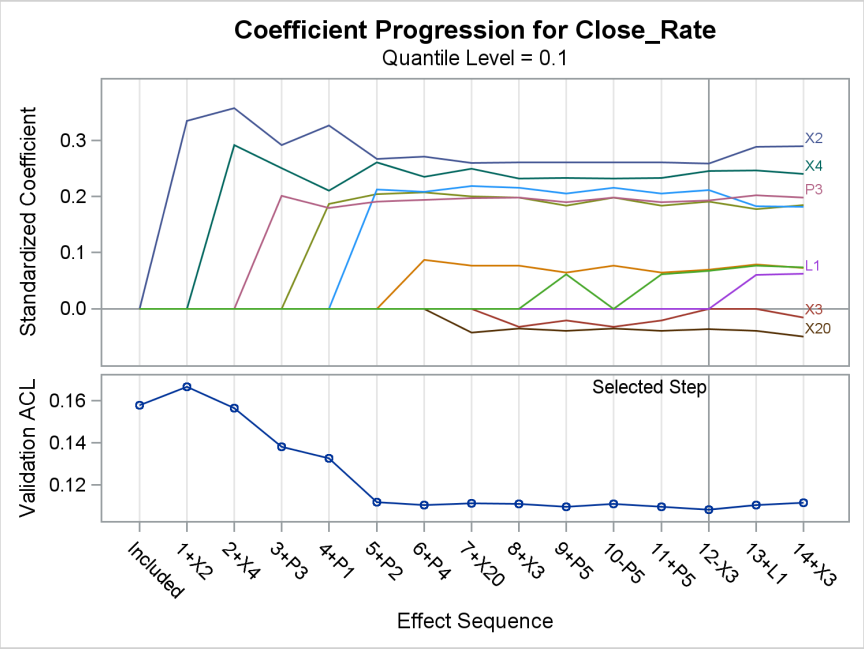


Figure 20 shows the fit statistics for the final model for quantile level 0.1.

Figure 20 Fit Statistics for Model Selected for Quantile Level 0.1

The QUANTSELECT Procedure
Quantile Level = 0.1

Fit Statistics	
Objective Function	36.17929
R1	0.38327
Adj R1	0.36909
AIC	-1616.52369
AICC	-1616.00496
SBC	-1581.62407
ACL (Train)	0.10134
ACL (Validate)	0.10826

Figure 21 shows the parameter estimates for the final model for quantile level 0.1.

Figure 21 Parameter Estimates for Model Selected for Quantile Level 0.1

Parameter Estimates			
Parameter	**DF**	**Estimate**	**Standardized Estimate**
Intercept	1	60.097618	0
X2	1	0.953402	0.258498
X4	1	0.933705	0.245902
X20	1	-0.140895	-0.035981
P1	1	0.724145	0.190798
P2	1	0.783880	0.211752
P3	1	0.696274	0.193163
P4	1	0.260641	0.069442
P5	1	0.242147	0.067135

The QUANTSELECT procedure produces a parallel but distinct set of results for quantile levels 0.5 and 0.9. The parameter estimates for the final models are shown in Figure 22 and Figure 23.

Figure 22 Parameter Estimates for Model Selected for Quantile Level 0.5

Parameter Estimates			
Parameter	DF	Estimate	Standardized Estimate
Intercept	1	60.950579	0
X2	1	1.508595	0.409029
X4	1	0.710687	0.187168
P3	1	0.361047	0.100163
P4	1	0.669943	0.178491
P5	1	0.544278	0.150902

Figure 23 Parameter Estimates for Model Selected for Quantile Level 0.9

Parameter Estimates			
Parameter	DF	Estimate	Standardized Estimate
Intercept	1	61.079231	0
X2	1	0.982776	0.266463
X4	1	1.118507	0.294572
L2	1	1.027725	0.297930
L3	1	0.859988	0.240257
L5	1	0.672210	0.186588
P5	1	0.192967	0.053500

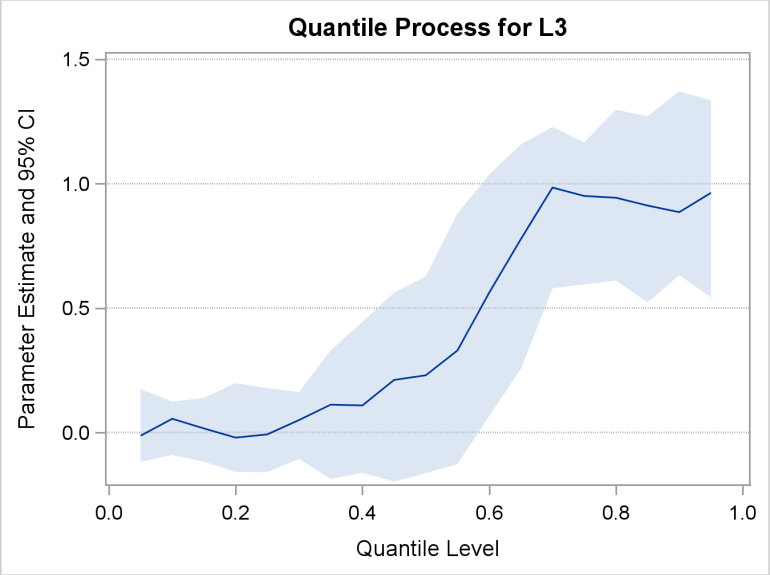
A sparse model with only six variables (**X2**, **X4**, **L2**, **L3**, **L5**, and **P5**) is selected as the best conditional model for predicting the 90th percentile. The layout variables **L2**, **L3**, and **L5** are in this model, but not in the models for the 10th and 50th percentiles. The variables **X2** and **X4** are common to the models for all three percentiles. These results give you insights about store performance that you would not obtain directly from standard regression methods.

You can create quantile process plots that show how the estimated regression coefficients for a covariate change as a function of the quantile level τ in the interval (0,1). The following program creates a process plot for **L3**. First the QUANTSELECT procedure is used to build a quantile process regression model. Then the QUANTREG procedure is used to compute 95% confidence limits for the coefficients.

```
proc quantselect data=Stores plots=Coefficients seed=15531;
   model Close_Rate = X1-X20 L1-L6 P1-P6 / quantile=process(ntau=10)
                                           selection=forward(sh=3);
run;

proc quantreg data=Stores;
   ods output ParameterEstimates=ParmEst;
   model Close_Rate = &_QRSIND / quantile=0.05 to 0.95 by 0.05;
run;

data ParmEstPlot; set ParmEst; if Parameter EQ "L3"; run;

title "Quantile Process for L3";
proc sgplot data=ParmEstPlot noautolegend;
   band upper=UpperCL lower=LowerCL x=Quantile / transparency=0.5;
   series y=Estimate x=Quantile;
   yaxis label='Parameter Estimate and 95% CI'
         grid gridattrs=(thickness=1px color=gray pattern=dot);
   xaxis label='Quantile Level';
run;
```

The process plot, shown in Figure 24, reveals that **L3** affects the upper half of the close rate distribution. Again, this is an insight that you would not obtain with standard regression methods.

Figure 24 Quantile Process Plot for L3



Fitting Generalized Additive Models with the 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).

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.

Each spline term is constructed 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.

Table 6 summarizes the components of a generalized additive model.

Table 6 Components of Generalized Additive Models

Component	Description
Linear predictor	Effects involving continuous or classification variables
Nonparametric predictor	Spline terms involving one or more continuous variables
Link function	Log, logit, inverse, and so on
Distribution	Normal, binomial, Poisson, gamma, and so on

Because a generalized additive model allows both linear and nonparametric predictors, it is useful for problems involving unknown—possibly nonlinear—relationships between the response and the predictors, as well as relationships that can be assumed to be linear. Frigo and Osterloo (2016) describe a problem of this type in the context of insurance pricing and propose solutions that use the GAMPL procedure and the HPGENSELECT procedure.

Strictly speaking, the GAMPL procedure does model fitting rather than model building. Unlike the GLMSELECT, HPGENSELECT, and QUANTSELECT procedures, the GAMPL procedure does not select variables. However, in some situations the results of spline fits that you obtain using PROC GAMPL suggest parametric effects in a model that you can then fit with the HPGENSELECT 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 data set 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 7.

Table 7 Predictors for Claim Rate

Predictor	Description	Contribution
Age	Age of loan	Unknown, possibly quadratic
Price	Price of house	Unknown, nonlinear
RefInd	Indicator if loan is refinanced	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 illustrative purposes.

The following statements use the GAMPL procedure to fit a generalized additive model for **Rate**:

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

The PARAM( ) option specifies parametric linear terms for **RefInd**, **PayIncmRatio**, **RefInctvRatio**, and **UnempRate**. The SPLINE options specify spline effects for **Age** and **Price**.

Figure 25 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 25 Model Information

The GAMPL Procedure

Model Information	
Data Source	WORK.CLAIMS
Response Variable	Rate
Class Parameterization	GLM
Distribution	Poisson
Link Function	Log
Fitting Method	Performance Iteration
Fitting Criterion	UBRE
Optimization Technique for Smoothing	Newton-Raphson
Random Number Seed	1990293722

Figure 26 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 use the GCV criterion for comparisons with other generalized additive models or penalized models.

Figure 26 Fit Statistics with GAMPL Procedure

Fit Statistics	
Penalized Log Likelihood	-26776
Roughness Penalty	7.83354
Effective Degrees of Freedom	16.54759
Effective Degrees of Freedom for Error	9982.63719
AIC (smaller is better)	53578
AICC (smaller is better)	53578
BIC (smaller is better)	53697
UBRE (smaller is better)	-0.00355

Figure 27 and Figure 28 show estimates for the components of the model.

Figure 27 Estimates for Parametric Terms

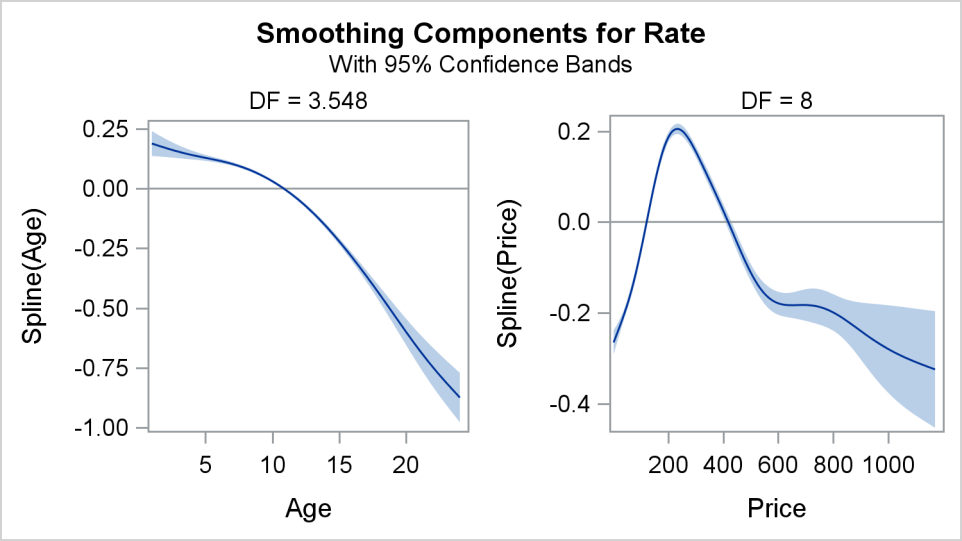
Parameter Estimates					
Parameter	**DF**	**Estimate**	**Standard Error**	**Chi-Square**	**Pr > ChiSq**
Intercept	1	2.484711	0.020877	14164.8501	<.0001
RefInd 0	1	-0.008901	0.005571	2.5532	0.1101
RefInd 1	0	0	.	.	.
PayIncmRatio	1	0.035740	0.009740	13.4642	0.0002
RefInctvRatio	1	-0.031276	0.009627	10.5555	0.0012
UnempRate	1	0.008048	0.002764	8.4778	0.0036

Figure 28 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)	3.54759	35754.3	7.8335	9	10	24
Spline(Price)	8.00000	1.0000	1.045E-6	9	10	2000

Figure 29 displays plots of the fitted splines for **Age** and **Price**.

Figure 29 Spline Components for Age and Price



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

```
proc hpgenselect data=Claims;
   class RefInd;
   model Rate = RefInd PayIncmRatio RefInctvRatio UnempRate
                Age Age*Age Price Price*Price / dist=poisson;
run;
```

Fit statistics for the model that is fitted with PROC HPGENSELECT are given in Figure 30.

Figure 30 Fit Statistics with HPGENSELECT Procedure

The HPGENSELECT Procedure

Fit Statistics	
-2 Log Likelihood	54754
AIC (smaller is better)	54772
AICC (smaller is better)	54772
BIC (smaller is better)	54837
Pearson Chi-Square	11284
Pearson Chi-Square/DF	1.1294

The AIC, AICC, and BIC statistics in Figure 26 are smaller even though the generalized additive model involves more parameters for the splines.

Building Classification and Regression Tree Models with the HPSPLIT Procedure

The HPSPLIT procedure is a high-performance procedure that builds tree-based statistical models for classification and regression. The procedure produces classification trees, which model a categorical response, and regression trees, which model a continuous response. Both types of trees are referred to as decision trees, because the model is expressed as a series of if-then statements.

The predictor variables for tree models can be categorical or continuous. The model is based on a partition of the predictor space into nonoverlapping segments, which correspond to the leaves (terminal nodes) of the tree. Partitioning is done recursively, starting with the root node, which contains all the data. At each step, the parent node is split into child nodes through selection of a predictor variable and a split value that minimize the variability in the response across the child nodes.

Tree models are built from training data for which the response values are known, and these models are subsequently used to score (classify or predict) response values for new data. For classification trees, the most frequent response level of the training observations in a leaf is used to classify observations in that leaf. For regression trees, the average response of the training observations in a leaf is used to predict the response for observations in that leaf. The splitting rules that define the leaves provide the information that is needed to score new data.

The process of building a decision tree begins with growing a large, full tree. Various measures, such as the Gini index, entropy, and residual sum of squares, are used to assess candidate splits for each node. To prevent overfitting, the full tree is pruned back to a smaller tree that balances the goals of fitting training data and predicting new data. Two approaches for finding the best tree are cost-complexity pruning (Breiman et al. 1984) and C4.5 pruning (Quinlan 1993).

Example: Predicting Claim Rates for Loans (continued)

The following statements use the HPSPLIT procedure to build a regression tree for **Rate**:

```
proc hpsplit data=Claims seed=15531
     plots=(wholetree zoomedtree(nodes=('0' '3') depth=2));
   class RefInd;
   model Rate = RefInd PayIncmRatio RefInctvRatio UnempRate Age Price;
```

```
    grow variance;
    prune costcomplexity;
    partition fraction(validate=0.3);
run;
```

With 10,000 observations, it is reasonable to use a PARTITION statement to reserve 30% of the data for validation, leaving the remaining 70% for training. The GROW statement specifies the variance (residual sum of squares) criterion for determining variable splits. The PRUNE statement requests the cost-complexity method of pruning. The procedure uses the validation set to determine the size of the optimal tree. If a validation set is not specified, the procedure uses k-fold cross validation for this purpose.

Figure 31 provides information about the methods that are used to grow and prune the tree.

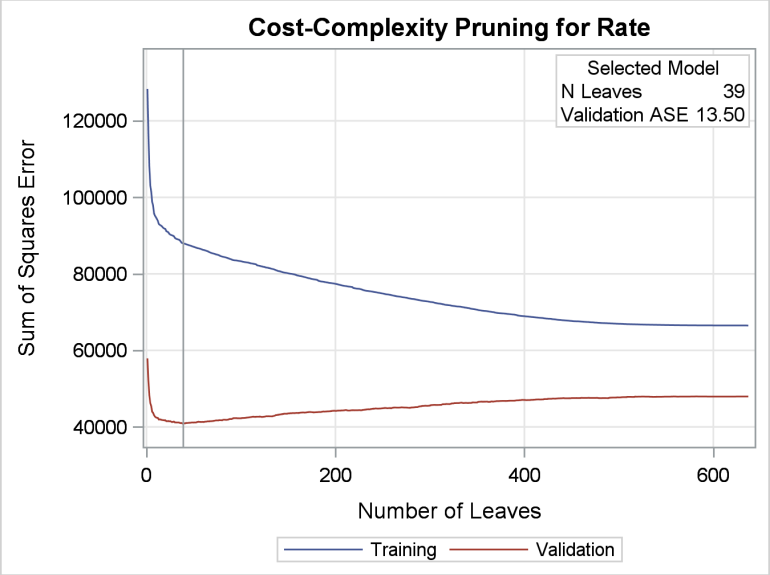
Figure 31 Model Information

The HPSPLIT Procedure

Model Information	
Split Criterion Used	Variance
Pruning Method	Cost-Complexity
Subtree Evaluation Criterion	Cost-Complexity
Number of Branches	2
Maximum Tree Depth Requested	10
Maximum Tree Depth Achieved	10
Tree Depth	10
Number of Leaves Before Pruning	637
Number of Leaves After Pruning	39

The cost-complexity pruning plot in Figure 32 displays the error sum of squares for the training and validation data as a function of the number of leaves. A tree size of 39 leaves minimizes this quantity.

Figure 32 Pruning Plot



The diagram in Figure 33, which is requested using the WHOLETREE option, provides an overview of the final tree, which has 39 leaves. The leaf color represents the predicted value of **Rate**, which is the average observed value of **Rate** for the training observations in that leaf.

Figure 33 Whole Tree Plot

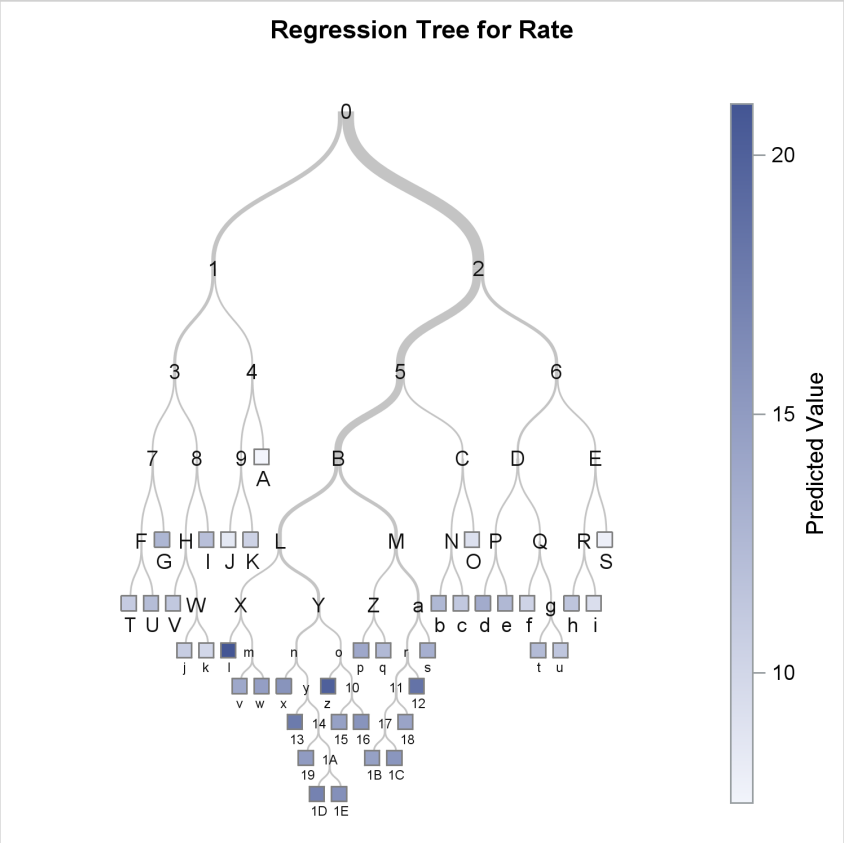
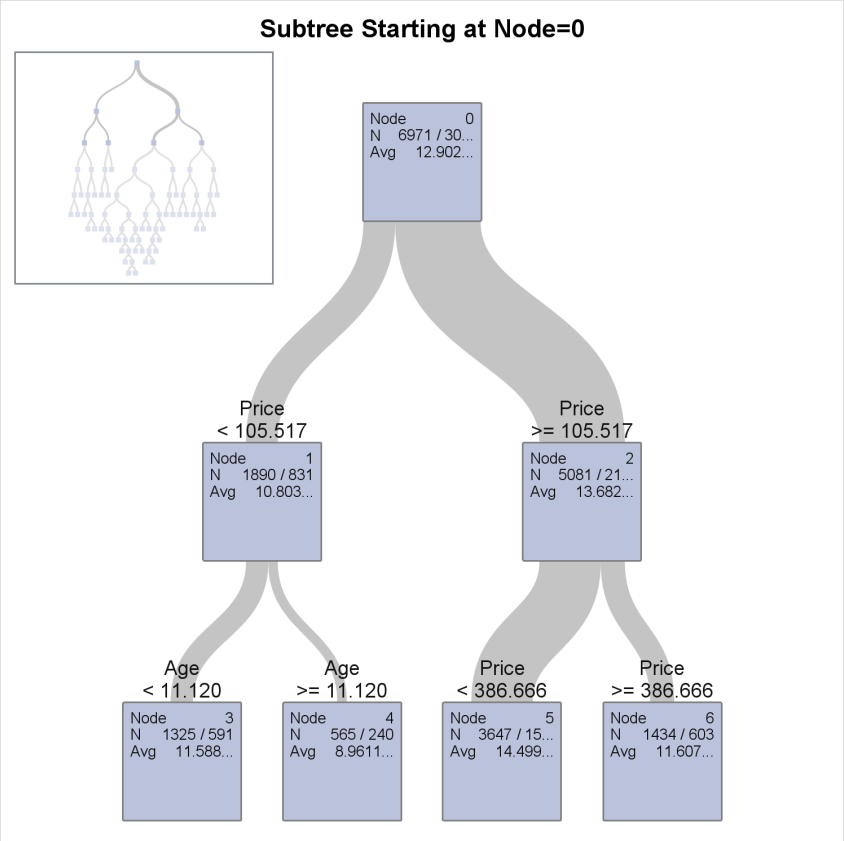


Figure 34 Zoomed Plot Starting at Node 0



The diagram in Figure 34, which is requested using the ZOOMEDTREE option, displays the root node (Node 0) and the next two levels of the final tree. Node 0 contains all of the 6,971 observations in the training data. The first split assigns the 1,890 observations where **Price** < 105.517 to Node 1, and the remaining 5,081 observations where **Price** $\geq$ 105.517 to Node 2. The next split assigns the observations in Node 1 where **Age** < 11.120 to Node 3. A second diagram, which is requested using the ZOOMEDTREE option and is not shown, displays Node 3 and the two levels that follow Node 3.

Figure 35 shows fit statistics for the final tree.

Figure 35 Fit Statistics

The HPSPLIT Procedure

Fit Statistics for Selected Tree

	N Leaves	ASE	RSS
Training	39	12.6203	87975.8
Validation	39	13.4979	40885.3

Figure 36 shows measures of variable importance. The variables **Price** and **Age** are the most useful predictors.

Figure 36 Variable Importance

Variable Importance

		Training		Validation			
Variable	**Variable Label**	**Relative**	**Importance**	**Relative**	**Importance**	**Relative Ratio**	**Count**
Price	Wtd Avg of House Price at Loan Origination	1.0000	157.4	1.0000	104.9	1.0000	14
Age	Age of Loan in Years	0.7728	121.6	0.7502	78.7345	0.9709	16
UnempRate	Wtd Avg of Unemployment Rates	0.0719	11.3167	0.1023	10.7354	1.4226	1
PayIncmRatio	Wtd Avg of Payment to Income Ratios	0.1582	24.8905	0.0892	9.3573	0.5638	6
RefInctvRatio	Wtd Avg of Refinance Incentive Ratios	0.0458	7.2074	0.0098	1.0288	0.2141	1

This example illustrates a limitation of regression tree models: they are adequate for fitting response surfaces that are constant over rectangular regions of the predictor space, but they lack the flexibility necessary to capture smooth relationships between the predictors and the response. In these situations, regression models with continuous effects will outperform tree models—and, in fact, for the claim rate prediction problem, the approaches discussed in the previous example provide better solutions. On the other hand, tree models offer the advantages of being easy to explain and handling missing values efficiently through the use of surrogate variables. For a comprehensive discussion of tree-based methods, see Hastie, Tibshirani, and Friedman (2009).

Summary: Benefits of Modern Approaches for Model Building

Table 8 provides a high-level comparison of the five approaches discussed in this paper. All these approaches share a common goal of delivering good predictive ability with future data, but they differ in the benefits that they offer and the assumptions that they require you to make.

All these approaches avoid overfitting the training data by giving you methods of choosing tuning parameters and computing model fit statistics that are based on information criteria and validation techniques. When you have sufficient data for partitioning, you should use validation data for choosing the tuning parameter and test data for assessing predictive ability.

The ability to score future data is an essential aspect of predictive modeling. All the procedures that are illustrated in this paper provide ways to score data with the final model, as summarized in Table 9.

In order to decide which modeling approaches are appropriate for your work, you should understand their underlying assumptions, characteristics, and relative benefits. These aspects are explained in the “Details” sections of the procedure chapters in the *SAS/STAT 14.1 User’s Guide*.

Table 8 New Tools for Regression Modeling in Recent Releases of SAS/STAT Software

Approach	Benefits	Model Type	Availability
Lasso methods for selecting regression effects	Sparse models for high-dimensional data; potentially more interpretable	Parametric	GLMSELECT, HPGENSELECT, QUANTSELECT
Effect selection for generalized linear models	Wide variety of response distributions	Parametric	HPGENSELECT, QUANTSELECT
Effect selection for quantile regression	Ability to model the entire conditional response distribution	Parametric	QUANTSELECT
Generalized additive models with penalization	Flexibility for capturing complex dependency relationships	Semiparametric	GAMPL
Classification and regression trees	Interpretability of small trees, handling of missing values	Nonparametric	HPSPLIT

Table 9 Functionality for Scoring

Procedure	Feature	Description
GLMSELECT	SCORE statement	Creates SAS data set that contains predicted values for new data
	CODE statement	Writes SAS DATA step code for computing predicted values
HPGENSELECT	CODE statement	Writes SAS DATA step code for computing predicted values
QUANTSELECT	CODE statement	Writes SAS DATA step code for computing predicted values
GAMPL	OUTPUT statement	Computes predicted values for observations with missing responses
HPSPLIT	CODE statement	Writes SAS DATA step code for computing predicted values

Keeping Up with New Releases of SAS/STAT

The model building approaches that are described in this paper are five of the many enhancements in recent releases of SAS/STAT software. The best place to find out about these enhancements is the chapter "What's New in SAS/STAT" in the online documentation at http://support.sas.com/documentation/onlinedoc/stat/. Also, be sure to visit the Statistics and Operations Research focus area at http://support.sas.com/statistics. There you can watch helpful videos, download overview papers, and subscribe to a quarterly e-newsletter.

REFERENCES

Breiman, L., Friedman, J., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees.* Belmont, CA: Wadsworth.

Cohen, R., and Rodriguez, R. N. (2013). "High-Performance Statistical Modeling." In *Proceedings of the SAS Global Forum 2013 Conference*. Cary, NC: SAS Institute Inc. http://support.sas.com/resources/papers/proceedings13/401-2013.pdf.

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

Fan, J., and Lv, J. (2008). "Sure Independence Screening for Ultrahigh Dimensional Feature Space." *Journal of the Royal Statistical Society, Series B* 70:849–911.

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.

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. Boca Raton, FL: CRC Press.

Hastie, T. J., Tibshirani, R. J., and Friedman, J. H. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. New York: Springer-Verlag.

Johnston, G., and Rodriguez, R. N. (2015). "Introducing the HPGENSELECT Procedure: Model Selection for Generalized Linear Models and More." In *Proceedings of the SAS Global Forum 2015 Conference*. Cary, NC: SAS Institute Inc. http://support.sas.com/resources/papers/proceedings15/SAS1742-2015.pdf.

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

Liu, J., Zhao, Z., Wang, J., and Ye, J. (2014). "Safe Screening with Variational Inequalities and Its Application to Lasso." In *JMLR Workshop and Conference Proceedings, Vol. 32: Proceedings of the Thirty-First International Conference on Machine Learning, Second Cycle*. http://jmlr.org/proceedings/papers/v32/liuc14.pdf.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. San Francisco: Morgan Kaufmann.

Rodriguez, R. N. (2014). "SAS/STAT 13.1: Round-Up." In *Proceedings of the SAS Global Forum 2014 Conference*. Cary, NC: SAS Institute Inc. http://support.sas.com/resources/papers/proceedings14/SAS181-2014.pdf.

Stokes, M. (2013). "Current Directions in SAS/STAT Software Development." In *Proceedings of the SAS Global Forum 2013 Conference*. Cary, NC: SAS Institute Inc. http://support.sas.com/resources/papers/proceedings13/432-2013.pdf.

Stokes, M., Chen, F., Yuan, Y., and Cai, W. (2012). "Look Out: After SAS/STAT 9.3 Comes SAS/STAT 12.1!" In *Proceedings of the SAS Global Forum 2012 Conference*. Cary, NC: SAS Institute Inc. http://support.sas.com/resources/papers/proceedings12/313-2012.pdf.

Stokes, M., and Statistical R&D Staff (2015). "SAS/STAT 14.1: Methods for Massive, Missing, or Multifaceted Data." In *Proceedings of the SAS Global Forum 2015 Conference*. Cary, NC: SAS Institute Inc. http://support.sas.com/resources/papers/proceedings15/SAS1940-2015.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.

Yuan, M., and Lin, L. (2006). "Model Selection and Estimation in Regression with Grouped Variables." *Journal of the Royal Statistical Society, Series B* 68:49–67.

Zou, H., and Hastie, T. (2005). "Regularization and Variable Selection via the Elastic Net." *Journal of the Royal Statistical Society, Series B* 67:301–320.

Acknowledgments

The following SAS/STAT developers contributed to this paper: Weijie Cai, Gordon Johnston, Jun Liu, Warren Kuhfeld, and Yonggang Yao. The author also thanks Ed Huddleston for editorial assistance.

Contact Information

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

Robert N. Rodriguez
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
Bob.Rodriguez@sas.com