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

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

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

SAS/STAT® 15.1 User's Guide

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

November 2018

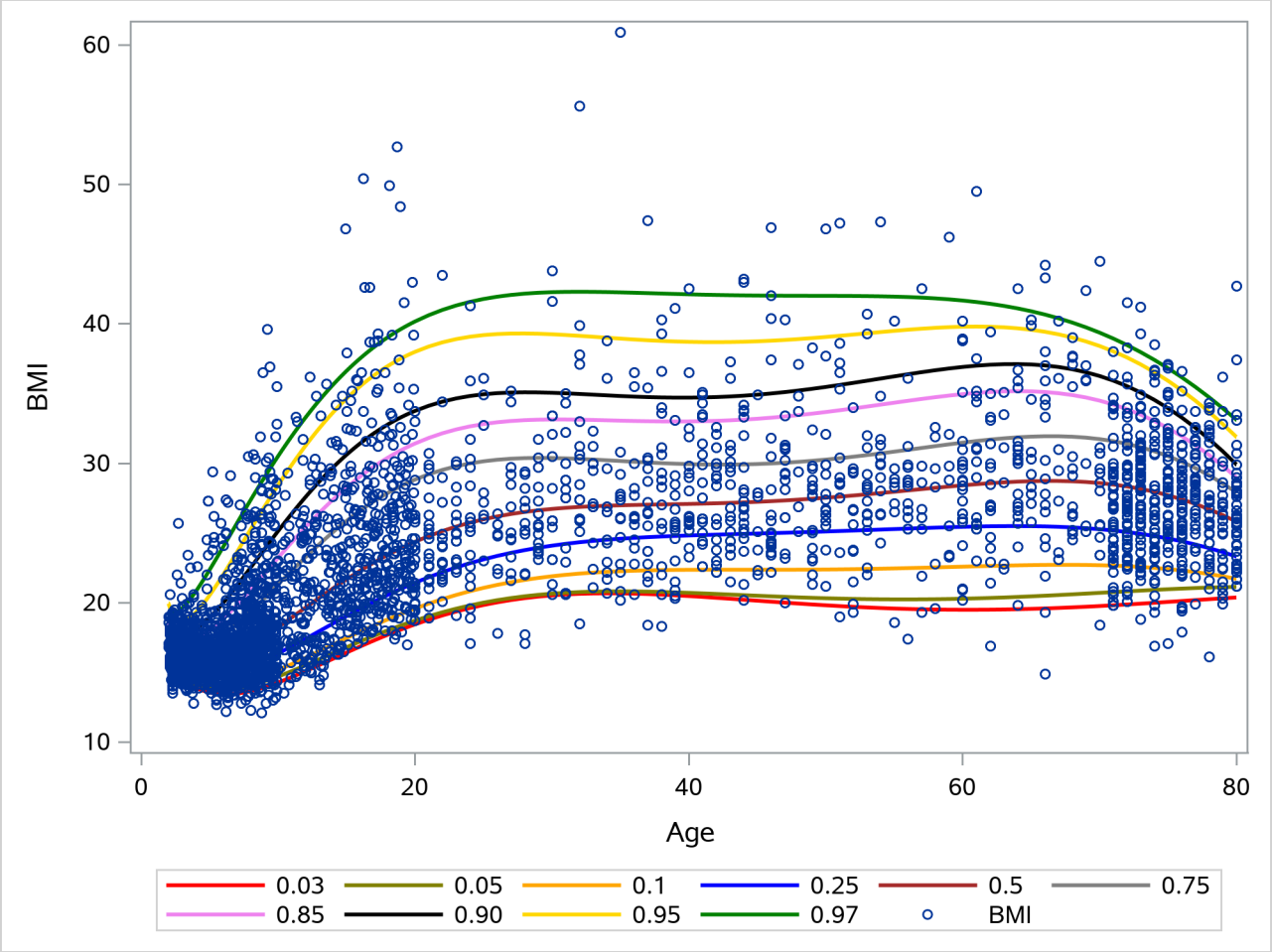
Chapter 63
The HPQUANTSELECT Procedure

Contents

Overview: HPQUANTSELECT Procedure

Quantile regression is a systematic statistical methodology for modeling conditional quantile functions of a response variable on explanatory covariate effects. Although modern quantile regression was introduced by Koenker and Bassett (1978), simple quantile regression that uses only the intercept as the explanatory effect has been practiced for much longer, because quantile is no more than a generalized notion for terms such as percentile, decile, quintile, and quartile. A conditional quantile of a response variable at quantile level τ denotes the value below which the proportion of the conditional response population is τ. Unlike linear regression, which exclusively focuses on the conditional mean, quantile regression can anatomize the entire response distribution and examine how the covariate effects influence the shape of the response distribution over the entire range of quantile levels $[0, 1]$. Therefore, quantile regression provides a more comprehensive view of the regression relationship. Figure 63.1 shows an example of quantile regression that creates growth charts for the men's body mass index (BMI) as quantile curves. Each entry in the legend shows the quantile level for the corresponding quantile curve. For example, the curve whose quantile level $\tau = 0.85$ corresponds to the 85th conditional percentile. For more information about the BMI example, see "Example 63.2: Growth Charts for Body Mass Index" on page 4944.

Figure 63.1 Growth Chart for Body Mass Index



The HPQUANTSELECT procedure is a high-performance procedure that fits and performs effect selection for quantile regression analysis. PROC HPQUANTSELECT supports continuous variables, CLASS variables, and the interactions of these variables. PROC HPQUANTSELECT supports statistical inferences on quantile regression models with or without the assumption of independently and identically distributed (iid) errors. PROC HPQUANTSELECT also offers extensive capabilities for customizing the effect selection by using a wide variety of selection and stopping criteria.

PROC HPQUANTSELECT runs in either single-machine mode or distributed mode. NOTE: Distributed mode requires SAS High-Performance Statistics.

PROC HPQUANTSELECT Features

The main features of the HPQUANTSELECT procedure are as follows:

- **Model specification**
  - supports quantile regression for single or multiple quantile levels
  - supports GLM and reference cell parameterization for classification effects
  - supports any degree of interaction (crossed effects) and nested effects
  - supports statistical inferences with or without iid errors assumption
  - supports hierarchy among effects
  - supports partitioning of data into training, validation, and testing roles
  - supports a CODE statement to write SAS DATA step code to a file or catalog entry for computing predicted quantiles
  - supports a WEIGHT statement for a weighted analysis
- **Selection control**
  - provides multiple effect-selection methods
  - offers selection of individual levels of classification effects
  - provides effect selection based on a variety of selection criteria
  - provides stopping rules based on a variety of model evaluation criteria
- **Display and output**
  - produces output data sets that contain predicted values, residuals, standardized errors, and confidence limits of predicted values

The HPQUANTSELECT procedure supports the following effect-selection methods. For more information about these methods, see the section "Methods" (Chapter 3, *SAS/STAT User's Guide: High-Performance Procedures*).

forward selection	starts with no effects in the model and adds effects.
backward elimination	starts with all effects in the model and deletes effects.
stepwise selection	is similar to the forward selection method except that effects already in the model do not necessarily stay there.

PROC HPQUANTSELECT Contrasted with Other SAS Procedures

For general contrasts between SAS High-Performance Analytics procedures and other SAS procedures, see the section "Common Features of SAS High-Performance Statistical Procedures" (Chapter 3, *SAS/STAT User's Guide: High-Performance Procedures*). The following remarks contrast the HPQUANTSELECT procedure with the QUANTSELECT and QUANTREG procedures in SAS/STAT.

The major functional differences between the HPQUANTSELECT and QUANTSELECT procedures are as follows:

- The HPQUANTSELECT procedure uses an interior point algorithm for model fitting. The QUANTSELECT procedure uses a flexible simplex algorithm for model fitting.
- The HPQUANTSELECT procedure can output confidence limits for parameter estimates.
- The HPQUANTSELECT procedure does not support the LASSO and adaptive LASSO effect-selection methods.
- The HPQUANTSELECT procedure does not support the TESTDATA= and VALDATA= options in its PROC statement.
- The HPQUANTSELECT procedure does not support graphical summaries for the effect selection processes.

Both the HPQUANTSELECT and QUANTSELECT procedures support the forward, backward, and stepwise effect-selection methods and the ability to use separate validation and test data via the PARTITION statement. For more information about the QUANTSELECT procedure, see Chapter 101, "The QUANTSELECT Procedure."

The major functional differences between the HPQUANTSELECT and QUANTREG procedures are as follows:

- The QUANTREG procedure does not support any effect-selection methods. It does not output the following fit statistics: AIC, AICC, SBS, VALIDATE, TEST, R1, and adjusted R1. And it does not support the PARTITION statement.
- The QUANTREG procedure provides three algorithms for fitting quantile regression models: simplex algorithm, interior point algorithm, and smoothing algorithm. The HPQUANTSELECT procedure supports only the interior point algorithm.
- The QUANTREG procedure supports the rank test, which is not available for the HPQUANTSELECT procedure. Both the QUANTREG and HPQUANTSELECT procedures support the Wald test and the likelihood test.
- The QUANTREG procedure supports two methods of estimating the covariance matrix of the parameter estimates: an asymptotic method and a bootstrap method. The HPQUANTSELECT procedure supports only the asymptotic method.

For more information about the QUANTREG procedure, see Chapter 100, "The QUANTREG Procedure."

The HPQUANTSELECT procedure is also different from the QUANTSELECT and QUANTREG procedures in the following respects:

- The HPQUANTSELECT procedure supports the CODE statement, which is not available in the QUANTSELECT and QUANTREG procedures.
- The HPQUANTSELECT procedure does not support quantile process regression, whereas the QUANTREG procedure does support quantile process regression. The QUANTSELECT procedure supports effect selection for quantile process regression.
- The HPQUANTSELECT procedure does not support the EFFECT statement, which provides constructed effects such as polynomial effects, spline effects, collection effects, and multimember classification effects.
- The HPQUANTSELECT procedure can output confidence limits for mean predicted quantiles; this functionality is not available in the QUANTSELECT and QUANTREG procedures.

In addition to having many similarities to the QUANTSELECT and QUANTREG procedures, the HPQUANTSELECT procedure also compares closely to the HPREG procedure. PROC HPREG is a high-performance procedure that performs effect selection in the framework of general linear models. The HPQUANTSELECT procedure inherits most of its syntax from the HPREG and QUANTREG procedures. The HPQUANTSELECT procedure provides results that are similar to those of the HPREG and QUANTSELECT procedures.

Getting Started: HPQUANTSELECT Procedure

The following example is modeled on the example in the section "Getting Started: QUANTSELECT Procedure" in the *SAS/STAT User's Guide*. The Sashelp.baseball data set contains salary and performance information for Major League Baseball (MLB) players, excluding pitchers, who played in at least one game in both the 1986 and 1987 seasons. The salaries (Time Inc. 1987) are for the 1987 season, and the performance measures are for the 1986 season (Reichler 1987).

The following statements display the variables in the data set. Figure 63.2 shows the results.

```
proc contents varnum data=sashelp.baseball;
   ods select position;
run;
```

Figure 63.2 Sashelp.Baseball Data Set

The CONTENTS Procedure

Variables in Creation Order

#	Variable	Type	Len	Label
1	Name	Char	18	Player's Name
2	Team	Char	14	Team at the End of 1986
3	nAtBat	Num	8	Times at Bat in 1986
4	nHits	Num	8	Hits in 1986
5	nHome	Num	8	Home Runs in 1986
6	nRuns	Num	8	Runs in 1986
7	nRBI	Num	8	RBIs in 1986
8	nBB	Num	8	Walks in 1986
9	YrMajor	Num	8	Years in the Major Leagues
10	CrAtBat	Num	8	Career Times at Bat
11	CrHits	Num	8	Career Hits
12	CrHome	Num	8	Career Home Runs
13	CrRuns	Num	8	Career Runs
14	CrRbi	Num	8	Career RBIs
15	CrBB	Num	8	Career Walks
16	League	Char	8	League at the End of 1986
17	Division	Char	8	Division at the End of 1986
18	Position	Char	8	Position(s) in 1986
19	nOuts	Num	8	Put Outs in 1986
20	nAssts	Num	8	Assists in 1986
21	nError	Num	8	Errors in 1986
22	Salary	Num	8	1987 Salary in $ Thousands
23	Div	Char	16	League and Division
24	logSalary	Num	8	Log Salary

Suppose you want to investigate how the MLB players' salaries for the 1987 season depend on performance measures for the players' previous season and MLB career. You might worry that some players who are outliers could dominate your least squares analysis. To address this concern, you can use the following statements to obtain a median regression model, which is equivalent to the 50th conditional percentile or the quantile regression model at quantile level 0.5:

```
proc hpquantselect data=sashelp.baseball;
   class league division;
   model Salary = nAtBat nHits nHome nRuns nRBI nBB
                  yrMajor crAtBat crHits crHome crRuns crRbi
                  crBB league division nOuts nAssts nError
         / clb;
run;
```

The CLB option in the MODEL statement requests 95% confidence limits for the parameter estimates. If you do not use the SELECTION statement, the HPQUANTSELECT procedure fits the full model that is specified by the MODEL statement without any effect selection.

Figure 63.3 Performance, Data Access, and Model Information

The HPQUANTSELECT Procedure

Performance Information	
Execution Mode	Single-Machine
Number of Threads	4

Data Access Information			
Data	**Engine**	**Role**	**Path**
SASHELP.BASEBALL	V9	Input	On Client

Model Information	
Data Source	SASHELP.BASEBALL
Dependent Variable	Salary
Class Parameterization	GLM
Optimization Algorithm	Interior

Figure 63.3 displays the "Performance Information," "Data Access Information," and "Model Information" tables.

The "Performance Information" table shows that the HPQUANTSELECT procedure executes in client mode—that is, the model is fit on the machine where the SAS session executes. This step was performed on a multicore machine that contained four CPUs; one computational thread was spawned per CPU.

The "Data Access Information" table shows that the input data set is accessed with the V9 (base) engine on the client machine.

The "Model Information" table identifies the data source and response and shows that the CLASS variables are parameterized in the GLM parameterization, which is the default.

Figure 63.4 Number of Observations, Class Level Information, and Dimensions Tables

Number of Observations Read	322
Number of Observations Used	263

Class Level Information		
Class	**Levels**	**Values**
League	2	American National
Division	2	East West

Dimensions	
Number of Effects	19
Number of Parameters	21

Figure 63.4 displays the "Number of Observations," "Class Level Information," and "Dimensions" tables.

The "Number of Observations" table shows that, of the 322 observations, PROC HPQUANTSELECT uses only 263 observations for model fitting and ignores 59 incomplete observations.

The "Class Level Information" table shows level information for two CLASS effects that the CLASS statement identifies: League and Division. League has two levels: American League and National League. Division also has two levels: East Division and West Division.

The "Dimensions" table shows that the MODEL statement identifies 19 effects for model fitting besides the intercept effect. Because the 19 effects include two CLASS effects and each level of a CLASS effect corresponds to a parameter, the 19 effects contain a total of 21 parameters.

Figure 63.5 Fit Statistics

The HPQUANTSELECT Procedure

Quantile Level = 0.5

Fit Statistics	
Objective Function	25977
R1	0.40584
Adj R1	0.36200
AIC	2453.81587
AICC	2456.94344
SBC	2521.68680
ACL	98.77118

Figure 63.5 displays the "Fit Statistics" table, which shows the values of model fitting criteria for the fitted median model. For more information about model fitting criteria for quantile regression, see the section "Details: HPQUANTSELECT Procedure" on page 4925.

Figure 63.6 Parameter Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		t Value	Pr > |t|
Intercept	1	-67.75322	39.95908	-146.46197	10.95553	-1.70	0.0912
nAtBat	1	-1.57112	0.44700	-2.45160	-0.69064	-3.51	0.0005
nHits	1	8.82192	1.94990	4.98114	12.66270	4.52	<.0001
nHome	1	-5.91757	4.91015	-15.58926	3.75412	-1.21	0.2293
nRuns	1	-5.17076	2.14914	-9.40400	-0.93753	-2.41	0.0169
nRBI	1	0.77547	2.15469	-3.46870	5.01963	0.36	0.7192
nBB	1	5.28866	1.67603	1.98732	8.59000	3.16	0.0018
YrMajor	1	6.61877	6.61798	-6.41690	19.65444	1.00	0.3182
CrAtBat	1	-0.04463	0.15485	-0.34964	0.26038	-0.29	0.7734
CrHits	1	0.07896	0.73594	-1.37064	1.52857	0.11	0.9146
CrHome	1	3.78231	1.90065	0.03854	7.52607	1.99	0.0477
CrRuns	1	1.23105	0.77137	-0.28833	2.75044	1.60	0.1118
CrRbi	1	-0.70695	0.76888	-2.22144	0.80754	-0.92	0.3588
CrBB	1	-0.68911	0.41382	-1.50423	0.12601	-1.67	0.0971
League American	1	-34.39136	24.37175	-82.39724	13.61451	-1.41	0.1595
League National	0	0	.	.	.	.	.
Division East	1	60.30856	27.28730	6.55984	114.05728	2.21	0.0280
Division West	0	0	.	.	.	.	.
nOuts	1	0.23273	0.12110	-0.00581	0.47126	1.92	0.0558
nAssts	1	0.09824	0.18888	-0.27381	0.47029	0.52	0.6035
nError	1	-0.81574	3.51436	-7.73810	6.10661	-0.23	0.8166

Figure 63.6 displays the "Parameter Estimates" table, which shows the parameter estimates of the fitted median model. You can see that, of the 19 effective parameters whose degrees of freedom are not zero, the fitted model contains 13 insignificant parameters whose 95% confidence intervals cover zeros. Because more than half of the 19 effective parameters are insignificant, you might worry that the model is overfitted.

It is well known that both overfitting and underfitting harm the prediction performance of a model. You can prevent overfitting and underfitting by using a good effect-selection technique. The following statements apply the forward selection method and the SL (significance level) criterion to choose a parsimonious model for the Sashelp.baseball data set:

```
proc hpquantselect data=sashelp.baseball;
   class league division;
   model Salary = nAtBat nHits nHome nRuns nRBI nBB
                  yrMajor crAtBat crHits crHome crRuns crRbi
                  crBB league division nOuts nAssts nError
         / clb;
   selection method=forward(select=sl sle=0.1);
run;
```

The SLE=0.1 option in the SELECTION statement specifies the significance level for entry. A candidate effect can enter the model at a certain selection step only if the following conditions are met:

- Its *p*-value is the smallest among all the valid candidate effects.
- Its *p*-value is smaller than 0.1 (the significance level for entry).

For more information about using significance levels in effect selection, see the section "Statistical Tests for Significance Level" on page 4929.

Figure 63.7 Selection Information

The HPQUANTSELECT Procedure

Selection Information	
Selection Method	Forward
Select Criterion	Significance Level
Stop Criterion	Significance Level
Effect Hierarchy Enforced	None
Entry Significance Level (SLE)	0.1
Stop Horizon	1

Figure 63.7 displays the "Selection Information" table. The "Selection Information" provides details about the method and criteria used to perform the model selection. The requested selection method is the forward selection method where the decisions about what effects to add at any step and when to terminate the selection are both based on the significance level criterion.

Figure 63.8 Selection Summary

The HPQUANTSELECT Procedure

Quantile Level = 0.5

Selection Summary

Step	Effect Entered	Number Effects In	p Value
0	Intercept	1	.
1	CrHome	2	<.0001
2	nHits	3	<.0001
3	CrHits	4	<.0001
4	nOuts	5	0.0185
5	nAtBat	6	0.0182
6	Division	7	0.0118
7	nBB	8	0.0647
8	nRuns	9	0.0558

Figure 63.8 displays the "Selection Summary" table. Each row in the "Selection Summary" table shows the effect that enters the model at the corresponding step of the effect selection process together with its *p*-value for adding the effect into the model at that step.

Figure 63.9 Stopping and Selection Reasons

Selection stopped because no candidate for entry is significant at the 0.1 level.

The model at step 8 is selected.

The HPQUANTSELECT Procedure

Quantile Level = 0.5
Selected Model

Selected Effects: Intercept nAtBat nHits nRuns nBB CrHits CrHome Division nOuts

Figure 63.9 displays the "Stop Reason," "Selection Reason," and "Selected Effects" tables. The "Stop Reason" and "Selection Reason" tables indicate that effect selection stopped because no candidate for entry was significant at the 0.1 level after step 8. The "Selected Effects" table lists the effects that are included in the selected model.

Figure 63.10 Details of the Selected Model

Fit Statistics

Objective Function	26568
R1	0.39232
Adj R1	0.37318
AIC	2445.64547
AICC	2446.35693
SBC	2477.79485
ACL	101.01768

Figure 63.10 *continued*

Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		t Value	Pr > |t|
Intercept	1	-130.65536	33.04546	-195.73336	-65.57735	-3.95	<.0001
nAtBat	1	-1.20522	0.41690	-2.02624	-0.38419	-2.89	0.0042
nHits	1	7.76667	1.81045	4.20127	11.33207	4.29	<.0001
nRuns	1	-3.92180	1.87567	-7.61566	-0.22795	-2.09	0.0375
nBB	1	3.92049	1.04341	1.86565	5.97532	3.76	0.0002
CrHits	1	0.17697	0.06856	0.04195	0.31198	2.58	0.0104
CrHome	1	1.66939	0.73083	0.23012	3.10866	2.28	0.0232
Division East	1	65.14327	24.48038	16.93289	113.35364	2.66	0.0083
Division West	0	0	.	.	.	.	.
nOuts	1	0.23719	0.11403	0.01261	0.46176	2.08	0.0385

The "Fit Statistics" and "Parameter Estimates" tables in Figure 63.10 give details of the final selected model. You can see that all nine effective parameters (excluding Division West) are significant at the 5% significance level, corresponding to the 95% confidence limits.

Like the sample median, a median regression model is robust to extreme observations, because it depends only on a small middle subset of all the observations in the data set. However, it is less representative of the entire conditional distribution of the response variable. You might want to further investigate the Sashelp.baseball data set at other quantile levels. The following statements select quantile regression models at the quantile levels 0.1 and 0.9, which correspond to the 10% and 90% conditional percentiles of the players' salaries:

```
proc hpquantselect data=sashelp.baseball alpha=0.1;
   class league division;
   model Salary = nAtBat nHits nHome nRuns nRBI nBB
                  yrMajor crAtBat crHits crHome crRuns crRbi
                  crBB league division nOuts nAssts nError
         / quantile=0.1 0.9 clb;
   selection method=backward(select=sl sls=0.1);
run;
```

The ALPHA=0.1 option in the PROC statement sets the significance level to 0.1. Combined with the CLB option in the MODEL statement, the ALPHA=0.1 option requests 90% confidence limits for parameter estimates. The QUANTILE= option in the MODEL statement specifies two quantile levels, 0.1 and 0.9, for fitting quantile regression models. The METHOD=BACKWARD option in the SELECTION statement specifies the backward elimination method of effect selection.

Figure 63.11 Parameter Estimates at Quantile Level 0.1

The HPQUANTSELECT Procedure

Quantile Level = 0.1
Selected Model

Selected Effects:	Intercept nAtBat nHits nBB CrRuns CrBB Division nAssts

Figure 63.11 *continued*

Parameter Estimates

Parameter	DF	Estimate	Standard Error	90% Confidence Limits		t Value	Pr > |t|
Intercept	1	4.75224	18.94983	-26.53111	36.03558	0.25	0.8022
nAtBat	1	-0.73670	0.17958	-1.03316	-0.44024	-4.10	<.0001
nHits	1	2.69396	0.63510	1.64551	3.74241	4.24	<.0001
nBB	1	1.81807	0.40595	1.14791	2.48823	4.48	<.0001
CrRuns	1	0.65476	0.09560	0.49694	0.81259	6.85	<.0001
CrBB	1	-0.44622	0.16254	-0.71454	-0.17790	-2.75	0.0065
Division East	1	28.35406	10.68922	10.70774	46.00038	2.65	0.0085
Division West	0	0	.	.	.	.	.
nAssts	1	0.14958	0.05897	0.05223	0.24694	2.54	0.0118

Figure 63.11 displays the "Selected Effects" and "Parameter Estimates" tables at quantile level 0.1.

Figure 63.12 Parameter Estimates at Quantile Level 0.9

The HPQUANTSELECT Procedure

Quantile Level = 0.9
Selected Model

Selected Effects: Intercept nHits nBB CrAtBat CrHits CrHome CrRbi League Division nOuts

Parameter Estimates

Parameter	DF	Estimate	Standard Error	90% Confidence Limits		t Value	Pr > |t|
Intercept	1	20.39804	58.17164	-75.63745	116.43353	0.35	0.7261
nHits	1	2.30897	0.55640	1.39042	3.22752	4.15	<.0001
nBB	1	3.09799	1.44414	0.71386	5.48213	2.15	0.0329
CrAtBat	1	-0.44914	0.14651	-0.69101	-0.20727	-3.07	0.0024
CrHits	1	2.48064	0.51725	1.62672	3.33457	4.80	<.0001
CrHome	1	6.29896	1.37134	4.03502	8.56291	4.59	<.0001
CrRbi	1	-2.12293	0.76546	-3.38662	-0.85923	-2.77	0.0060
League American	1	-103.28955	33.68480	-158.89975	-47.67935	-3.07	0.0024
League National	0	0	.	.	.	.	.
Division East	1	107.46694	50.82797	23.55512	191.37876	2.11	0.0355
Division West	0	0	.	.	.	.	.
nOuts	1	0.39766	0.12820	0.18601	0.60931	3.10	0.0021

Figure 63.11 displays the "Selected Effects" and "Parameter Estimates" tables at quantile level 0.9.

You might want to compute the 90th percentile predictions for players' salaries and find out which players were overpaid based on the quantile regression model at quantile level 0.9. The following statements repeat the backward elimination method at quantile level 0.9, compute and sort the overpaid players' salaries, and output the observations for the top 10 overpaid players in the Sashelp.baseball data set:

```
proc hpquantselect data=sashelp.baseball alpha=0.1;
   id Name;
   class league division;
   model Salary = nAtBat nHits nHome nRuns nRBI nBB
                  yrMajor crAtBat crHits crHome crRuns crRbi
                  crBB league division nOuts nAssts nError
         / quantile=0.9 clb;
   selection method=backward(select=sl sls=0.1);
   output out=BaseballOverpaid copyvar=Salary r=Overpaid
          p=PredictedSalary lclm uclm;
run;

proc sort data=BaseballOverpaid;
   by descending Overpaid;
run;

proc print data=BaseballOverpaid(obs=10);
   var Name Salary Overpaid PredictedSalary lclm uclm;
run;
```

The ID statement adds the variable Name from the input data set to the BaseballOverpaid data set that is produced by the OUTPUT statement. The LCLM and UCLM options, respectively, request lower and upper bounds of $100(1-\alpha)\%$ confidence intervals for the expected conditional quantile predictions of players' salaries at quantile level 0.9.

Figure 63.13 Top 10 Overpaid Baseball Players at Quantile Level 0.9

Obs	Name	Salary	Overpaid	PredictedSalary	LCLM	UCLM
1	Smith, Ozzie	1940.0	1084.54	855.46	611.09	1099.83
2	Wiggins, Alan	700.0	220.74	479.26	345.93	612.60
3	Murray, Eddie	2460.0	213.10	2246.90	1982.06	2511.74
4	Strawberry, Darryl	1220.0	187.64	1032.36	914.99	1149.72
5	Gibson, Kirk	1300.0	177.24	1122.76	1011.97	1233.56
6	Trevino, Alex	512.5	149.70	362.80	273.17	452.43
7	Ramirez, Rafael	875.0	141.09	733.91	599.52	868.31
8	Romero, Ed	375.0	128.71	246.29	144.88	347.70
9	Mattingly, Don	1975.0	124.82	1850.18	1557.81	2142.55
10	Puhl, Terry	900.0	104.34	795.66	625.51	965.81

Output 63.13 shows the information about the top 10 overpaid players according to the final selected quantile regression model at quantile level 0.9. Ozzie Smith is in first place. This might be because, although Smith was known for his defensive brilliance, the model weights offensive performance measures much more than defensive performance measures.

Syntax: HPQUANTSELECT Procedure

The following statements are available in the HPQUANTSELECT procedure:

PROC HPQUANTSELECT < *options* > ;
BY *variables* ;
CLASS *variable* < **(***options***)** >. . . < *variable* < **(***options***)** > > < / *global-options* > ;
CODE < *options* > ;
ID *variables* ;
MODEL *dependent* **=** < *effects* > < / *model-options* > ;
OUTPUT < **OUT=***SAS-data-set* >
< *keyword* < **=***name* > >. . .
< *keyword* < **=***name* > > < / *options* > ;
PARTITION < *partition-options* > ;
PERFORMANCE *performance-options* ;
SELECTION *selection-options* ;
WEIGHT *variable* ;

The PROC HPQUANTSELECT statement and a single MODEL statement are required. All other statements are optional. The CLASS statement can appear multiple times. If a CLASS statement is specified, it must precede the MODEL statement.

The rest of this section provides detailed syntax information about each of the preceding statements, beginning with the PROC HPQUANTSELECT statement. The remaining statements are described in alphabetical order.

PROC HPQUANTSELECT Statement

PROC HPQUANTSELECT < *options* > ;

The PROC HPQUANTSELECT statement invokes the HPQUANTSELECT procedure. Table 63.1 summarizes the options in the PROC HPQUANTSELECT statement by function.

Table 63.1 PROC HPQUANTSELECT Statement Options

Option	Description
Basic Options	
DATA=	Specifies the input data set
MAXMACRO=	Specifies the maximum number of macro variables to produce
NAMELEN=	Limits the length of effect names
Options Related to Output	
NOCLPRINT	Limits or suppresses the display of CLASS levels
NOPRINT	Suppresses ODS output
User-Defined Formats	
FMTLIBXML=	Specifies a file reference for a format stream

Table 63.1 *continued*

Option	Description
Other Options	
ALPHA=	Sets the significance level to use for the construction of confidence intervals
SEED=	Sets the seed used for pseudorandom number generation

The following list describes these *options* in alphabetical order:

ALPHA=*number*
sets the significance level to use for the construction of confidence intervals. The value must be between 0 and 1; the default value of 0.05 results in 95% intervals. This option affects the STDP, LCLM, and UCLM keywords in the OUTPUT statement and the CLB option in the MODEL statement.

DATA=*SAS-data-set*
names the input SAS data set to be used by PROC HPQUANTSELECT. The default is the most recently created data set.

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

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

MAXMACRO=*n*
specifies the total maximum number of macro variables to produce. Each macro variable contains selected effects for a selected model. For more information about the macro variables, see the section "Macro Variables That Contain Selected Effects" on page 4933. By default, MAXMACRO=100.

NAMELEN=*number*
specifies the length to which long effect names are to be shortened. The default and minimum value is 20.

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

NOPRINT
suppresses the generation of ODS output.

SEED=*number*
specifies an integer to be used to start the pseudorandom number generator for random partitioning of data for training, testing, and validation. If you do not specify a seed, or if you specify a value less than or equal to 0, the seed is generated from reading the time of day from the computer's clock.

BY Statement

BY *variables* **;**

You can specify a BY statement in PROC HPQUANTSELECT to obtain separate analyses of observations in groups that are defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. If you specify more than one BY statement, only the last one specified is used.

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

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the NOTSORTED or DESCENDING option in the BY statement in the HPQUANTSELECT procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

Processing of BY statements is not supported when the HPQUANTSELECT procedure runs alongside the database or alongside the Hadoop Distributed File System (HDFS). These modes are used if the input data are stored in a database or HDFS and the grid host is the appliance that houses the data.

For more information about BY-group processing, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*.

CLASS Statement

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

The CLASS statement names the classification variables to be used as explanatory variables in the analysis. The CLASS statement must precede the MODEL statement.

The CLASS statement for SAS high-performance analytical procedures is documented in the section "CLASS Statement" (Chapter 3, *SAS/STAT User's Guide: High-Performance Procedures*). The HPQUANTSELECT procedure also supports the following *global-option* in the CLASS statement:

UPCASE
uppercases the values of character-valued CLASS variables before levelizing them. For example, if you specify the UPCASE option and a CLASS variable can take the values "a," "A," and "b," then "a" and "A" represent the same level and the CLASS variable is treated as having only two values: "A" and "B."

CODE Statement

CODE < *options* > **;**

The CODE statement writes SAS DATA step code for computing predicted values of the fitted model either to a file or to a catalog entry. This code can then be included in a DATA step to score new data.

Table 63.2 summarizes the *options* available in the CODE statement.

Table 63.2 CODE Statement Options

Option	Description
CATALOG=	Names the catalog entry where the generated code is saved
DUMMIES	Retains the dummy variables in the data set
ERROR	Computes the error function
FILE=	Names the file where the generated code is saved
FORMAT=	Specifies the numeric format for the regression coefficients
GROUP=	Specifies the group identifier for array names and statement labels
IMPUTE	Imputes predicted values for observations with missing or invalid covariates
LINESIZE=	Specifies the line size of the generated code
LOOKUP=	Specifies the algorithm for looking up CLASS levels
RESIDUAL	Computes residuals

For details about the syntax of the CODE statement, see the section "CODE Statement" on page 400 in Chapter 19, "Shared Concepts and Topics."

ID Statement

ID *variables* **;**

The ID statement lists one or more variables from the input data set that are transferred to output data sets created by SAS High-Performance Analytics procedures, provided that the output data set produces one (or more) records per input observation.

For information about the common ID statement in SAS high-performance analytical procedures, see the section "ID Statement" (Chapter 3, *SAS/STAT User's Guide: High-Performance Procedures*).

MODEL Statement

MODEL *dependent*=< *effects* > / < *options* > **;**

The MODEL statement names the dependent variable and the explanatory effects, including covariates, main effects, interactions, and nested effects. If you omit the explanatory effects, PROC HPQUANTSELECT fits an intercept-only model.

After the keyword MODEL, the dependent (response) variable is specified, followed by an equal sign. The explanatory effects follow the equal sign. For information about constructing the model effects, see the section "Specification and Parameterization of Model Effects" (Chapter 3, *SAS/STAT User's Guide: High-Performance Procedures*).

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

CLB
requests the $100(1-\alpha)\%$ upper and lower confidence limits for the parameter estimates. By default, the 95% limits are computed; you can use the ALPHA= option in the PROC HPQUANTSELECT statement to change the α level.

INCLUDE=*n*

INCLUDE=*single-effect*

INCLUDE=(*effects***)**
forces effects to be included in all models. If you specify INCLUDE=*n*, then the first *n* effects that are listed in the MODEL statement are included in all models. If you specify INCLUDE=*single-effect* or if you specify a list of effects within parentheses, then the specified effects are forced into all models. The effects that you specify in the INCLUDE= option must be explanatory effects that are defined in the MODEL statement.

NOINT
suppresses the intercept term that is otherwise included in the model.

ORDERSELECT
specifies that, for the selected model, effects be displayed in the order in which they first entered the model. If you do not specify this option, then effects in the selected model are displayed in the order in which they appear in the MODEL statement.

QUANTILES=*number-list*

QUANTILE=*number-list*
specifies the quantile levels for the quantile regression. You can specify any number of quantile levels in $(0, 1)$. If you do not specify this option, the HPQUANTSELECT procedure performs median regression effect selection that corresponds to QUANTILE=0.5.

SPARSITY(< BF | HS > < IID >)
specifies the suboptions for estimating the sparsity function. You can specify the Bofinger method by using the BF suboption or the Hall-Sheather method by using the HS suboption. By default, the Hall-Sheather method is used. You can also specify the IID suboption to assume that the quantile regression errors satisfy the independently and identically distributed (iid) assumption. Let f_i and F_i, respectively, denote the probability density function and the cumulative distribution function of the ith error for $i = 1, \ldots, n$. The iid assumption means that there exist f and F such that $f = f_1 = \cdots = f_n$ and

$F = F_1 = \cdots = F_n$. If you specify the IID option, the covariance matrix of the parameter estimates, $\omega^2(\tau, F)(\mathbf{X}'\mathbf{X})^-$, is adopted for computing the confidence limits and the Wald statistics, where $\omega^2(\tau, F) = \tau(1-\tau)/f^2(F^{-1}(\tau))$. By default, the covariance matrix of the parameter estimates is non-iid and takes the sandwich form: $n^{-2}\tau(1-\tau)\mathbf{H}_n^-(\mathbf{X}'\mathbf{X})\mathbf{H}_n^-$, where $\mathbf{H}_n = n^{-1}\sum_{i=1}^n f_i(F_i^{-1}(\tau))\mathbf{x}_i\mathbf{x}_i'$. For more information, see the section "Details: HPQUANTSELECT Procedure" on page 4925.

START=*n*

START=*single-effect*

START=(*effects***)**

begins the effect-selection process in the forward and stepwise selection methods from the initial model that you designate. If you specify START=*n*, then the starting model consists of the first *n* effects listed in the MODEL statement. If you specify START=*single-effect* or if you specify a list of effects within parentheses, then the starting model consists of these specified effects. The effects that you specify in the START= option must be explanatory effects defined in the MODEL statement. The START= option is not available when you specify METHOD=BACKWARD in the SELECTION statement.

STB

produces standardized regression coefficients. A standardized regression coefficient is computed by dividing a parameter estimate by the ratio of the sample standard deviation of the dependent variable to the sample standard deviation of the regressor. If you use the INCLUDE= option to force some effects to be in the model, then the QTRSELECT procedure computes the sample standard deviation against all the effects that are forced in, as follows. Let $\mathbf{X}_1$ denote the design submatrix of p_1 regressors that consists of all the effects that are forced in, and let $\mathbf{z}$ denote the dependent variable or any regressor. Then the sample standard deviation of $\mathbf{z}$ is computed as

$$s_z = \sqrt{\frac{\mathbf{z}'\left[\mathbf{I} - \mathbf{X}_1(\mathbf{X}_1{}'\mathbf{X}_1)^{-1}\mathbf{X}_1{}'\right]\mathbf{z}}{n - p_1}}$$

TOL

produces tolerance values for the estimates. Tolerance for a parameter is defined as $1 - R^2$, where R^2 is obtained from the ordinary least squares regression of the parameter on all other parameters in the model.

VIF

produces variance inflation factors in the parameter estimates table. Variance inflation is the reciprocal of tolerance.

OUTPUT Statement

OUTPUT <**OUT=***SAS-data-set*>
<**COPYVARS=(***variables***)**>
<*keyword* <**=***name*>>...<*keyword* <**=***name*>> **;**

The OUTPUT statement creates a data set that contains observationwise statistics, which are computed after the final selected model is fit. To avoid data duplication for large data sets, the variables in the input data set are *not* included in the output data set; however, variables that are specified in the ID statement or COPYVARS= option are included.

If the input data are in distributed form, where access of data in a particular order cannot be guaranteed, the HPQUANTSELECT procedure copies the distribution or partition key to the output data set so that its contents can be joined with the input data.

The output statistics are computed based on the parameter estimates for the selected model. If you specify multiple quantile levels by using the QUANTILE= option in the MODEL statement, then for each appropriate keyword that is specified in the OUTPUT statement, one variable is generated for each specified quantile level. These variables appear in the sorted order of the specified quantile levels. For example, the following statements generate the Out data set, which contains the two predicted quantile variables p1 and p2:

```
proc hpquantselect data=one;
   model y = x1-x4 /quantile=0.5 0.3;
   output out=out pred=p;
run;
```

The variable p1 is for quantile level 0.3, and the variable p2 is for quantile level 0.5, because the sorted quantile levels are (0.3 0.5), not (0.5 0.3).

You can specify the following options in the OUTPUT statement:

OUT=*SAS-data-set*

DATA=*SAS-data-set*

specifies the name of the output data set. If you omit the OUT= (or DATA=) option, PROC HPQUANTSELECT uses the DATA*n* convention to name the output data set.

COPYVAR=*variable*

COPYVARS=(*variables***)**

transfers one or more *variables* from the input data set to the output data set. Variables that you name in an ID statement are also copied from the input data set to the output data set.

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

specifies the statistics to include in the output data set and optionally names the new variables that contain the statistics. Specify a *keyword* for each desired statistic (see the following list of *keywords*), followed optionally by an equal sign and a variable to contain the statistic.

If you specify *keyword***=***name*, the new variable that contains the requested statistic has the specified name. If you omit the optional **=***name* after a *keyword*, then a default name is used.

You can specify the following *keywords* to request statistics that are available with for selection methods:

LCLM

requests the lower bound of a $100(1-\alpha)\%$ confidence interval for the expected quantile of the dependent variable. The default variable name is LCLM.

PREDICTED

PRED

P

requests predicted values for the response variable. The default variable name is Pred.

RESIDUAL

RESID

R

requests the residual, calculated as ACTUAL – PREDICTED. The default variable name is Residual.

ROLE

requests a numeric variable that indicates the role that each observation plays in fitting the model. The default variable name is _ROLE_. For each observation, the interpretation of this variable is shown in Table 63.3.

Table 63.3 Role Interpretation

Value	Observation Role
0	Not used
1	Training
2	Validation
3	Testing

If you do not partition the input data by using a PARTITION statement, then the role variable value is 1 for observations that are used in fitting the model, and 0 for observations that have at least one missing or invalid value for the response, regressor, or weight variables.

You can use the following statements to display the values of the _ROLE_ variable in the output data set:

```
proc format;
   value role 0 = 'Not Used'
              1 = 'Training'
              2 = 'Validation'
              3 = 'Testing';
run;

proc freq;
   tables _role_;
   format _role_ role.;
run;
```

STDP

requests standard error of the mean predicted quantiles. The default variable name is STDP.

UCLM

requests the upper bound of a $100(1-\alpha)\%$ confidence interval for the expected quantile of the dependent variable. The default variable name is UCLM.

PARTITION Statement

PARTITION *partition-options* **;**

The PARTITION statement specifies how observations in the input data set are logically partitioned into disjoint subsets for model training, validation, and testing. Either you can designate a variable in the input data set and a set of formatted values of that variable to determine the role of each observation, or you can specify proportions to use for random assignment of observations for each role.

You can specify either of the following mutually exclusive *partition-options*:

FRACTION(< TEST=*fraction* **> < VALIDATE=***fraction* **>)**
randomly assigns the specified proportions of observations in the input data set to testing, validation, and training roles. You specify the proportions for testing and validation by using the TEST= and VALIDATE= suboptions. If you specify both the TEST= and VALIDATE= suboptions, then the sum of the specified fractions must be less than 1 and the remaining fraction of the observations are assigned to the training role.

ROLEVAR | ROLE=*variable***(< TEST='***value***' > < TRAIN='***value***' > < VALIDATE='***value***' >)**
names the variable in the input data set whose values are used to assign roles to each observation. Use the TEST=, TRAIN=, and VALIDATE= suboptions to specify the formatted values of this variable that are used to assign observation roles are specified in the TEST=, TRAIN=, and VALIDATE= suboptions. If you do not specify the TRAIN= suboption, then all observations whose roles are not determined by the TEST= and VALIDATE= suboptions are assigned to training.

To create an output data set variable that indicates the role assignment for either *partition-option*, specify the ROLE=*variable* option in the OUTPUT statement.

PERFORMANCE Statement

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

The PERFORMANCE statement defines performance parameters for multithreaded and distributed computing, passes variables about the distributed computing environment, and requests detailed results about the performance characteristics of a SAS high-performance analytical procedure.

You can also use the PERFORMANCE statement to control whether a SAS high-performance analytical procedure executes in single-machine or distributed mode.

The PERFORMANCE statement for SAS high-performance analytical procedures is documented in the section "PERFORMANCE Statement" (Chapter 2, *SAS/STAT User's Guide: High-Performance Procedures*).

SELECTION Statement

SELECTION < *options* > **;**

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

The HPQUANTSELECT procedure supports the following suboptions in the METHOD= option in the SELECTION statement to specify the corresponding effect selection methods:

NONE specifies no model selection.

FORWARD specifies the forward selection method, which starts with no effects in the model and adds effects.

BACKWARD specifies the backward elimination method, which starts with all effects in the model and deletes effects.

STEPWISE specifies the stepwise regression method, which is similar to the forward selection method except that effects already in the model do not necessarily stay there.

By default, the METHOD=STEPWISE option is used in the SELECTION statement. If you do not use the SELECTION statement, the HPQUANTSELECT procedure fits the full model that is specified by the MODEL statement; this is equivalent to specifying the METHOD=NONE option in the SELECTION statement. For information about all the selection criteria that are used in PROC HPQUANTSELECT, see the section "Criteria Used in Model Selection" on page 4928.

The DETAILS=ALL and DETAILS=STEPS options produce "Fit Statistics" and "Parameter Estimates" tables, which provide information about the model that is selected at each step of the selection process.

SHOWLASTSTEP

SLS

displays the last stop horizon step in the "Selection Summary" table, and includes this step for choosing the final model.

TEST=*value*

specifies a method to compute significance levels for the selection process. Table 63.4 summarizes these methods.

Table 63.4 Options for Significance Levels

Value of TEST=	Method
LR1	Likelihood ratio test Type I
LR2	Likelihood ratio test Type II
WALD	Wald score test

By default, PROC HPQUANTSELECT uses the Wald score to compute significance levels. If you specify the IID suboption in the SPARSITY option of the MODEL statement, the Wald score test uses the iid form of the covariance matrix to compute the Wald score and the associated significance levels. Otherwise, the non-iid form of the covariance matrix is used. The sparsity functions for both Type I and Type II likelihood ratio tests are estimated under the iid assumption no matter whether you specify the IID suboption.

WEIGHT Statement

WEIGHT *variable* **;**

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

The HPQUANTSELECT procedure mainly uses each valid weight as the scale factor of its relevant observation. Let $\mathbf{X}$ denote the design matrix before using weights, and let $\mathbf{W}$ denote the diagonal matrix whose diagonal elements are the weights. Then the HPQUANTSELECT procedure mainly uses the weighted design matrix $\mathbf{WX}$ for its computation. For example, the weighted version of $\mathbf{X}'\mathbf{X}$ is $\mathbf{X}'\mathbf{WWX}$ but not $\mathbf{X}'\mathbf{WX}$. The only exception is in computing the standardized parameter estimates, which computation uses the weighted standard deviation of the dependent variable and all regressors. These standard deviations are based effectively on the diagonal of $\mathbf{X}'\mathbf{WX}$.

Details: HPQUANTSELECT Procedure

Quantile Regression

This section describes the basic concepts and notations for quantile regression and quantile regression model selection.

Let $\{(y_i, \mathbf{x}_i) : i = 1, \ldots, n\}$ denote a data set of observations, where y_i are responses and $\mathbf{x}_i$ are regressors. Koenker and Bassett (1978) define the *regression quantile* at quantile level $\tau \in (0, 1)$ as any solution to the minimization problem

$$\min_{\boldsymbol{\beta} \in \mathbf{R}^p} \sum_{i=1}^{n} \rho_\tau \left(y_i - \mathbf{x}_i' \boldsymbol{\beta} \right)$$

where $\rho_\tau(r) = \tau r^+ + (1 - \tau) r^-$ is a check loss function in which $r^+ = \max(r, 0)$ and $r^- = \max(-r, 0)$.

If you specify weights $w_i, i = 1, \ldots, n$, in the WEIGHT statement, then weighted quantile regression is carried out by solving

$$\min_{\boldsymbol{\beta} \in \mathbf{R}^p} \sum_{i=1}^{n} \rho_\tau \left(w_i (y_i - \mathbf{x}_i' \boldsymbol{\beta}) \right)$$

The HPQUANTSELECT procedure fits a quantile regression model by using a predictor-corrector interior point algorithm, which was originally designed to solve support vector machine classifiers for large data sets (Gertz and Griffin 2005, 2010).

Linear Model with iid Errors

You can specify the SPARSITY(IID) option in the MODEL statement to assume that the distribution of Y_i conditional on $\mathbf{x}_i$ follows the linear model

$$Y_i = \mathbf{x}_i'\boldsymbol{\beta} + \epsilon_i$$

where ϵ_i for $i = 1, \ldots, n$ are iid in the distribution function F. Let $f = F'$ denote the density function of F. Further assume that $f(F^{-1}(\tau)) > 0$ in a neighborhood of τ. Then, under some mild conditions, Koenker and Bassett (1982) prove that the asymptotic distribution of the quantile regression estimates is

$$\sqrt{n}(\hat{\boldsymbol{\beta}}(\tau) - \boldsymbol{\beta}(\tau)) \rightarrow N(0, \omega^2(\tau, F)\boldsymbol{\Omega}^{-1})$$

where $\omega^2(\tau, F) = \tau(1-\tau)/f^2(F^{-1}(\tau))$ and $\boldsymbol{\Omega} = \lim_{n\rightarrow\infty} n^{-1}\sum \mathbf{x}_i\mathbf{x}_i'$. The reciprocal of the density function, $s(\tau) = 1/f(F^{-1}(\tau))$, is called the *sparsity function*.

Accordingly, the covariance matrix of $\hat{\boldsymbol{\beta}}(\tau)$ can be estimated as

$$\hat{\Sigma}(\tau) = \tau(1-\tau)\hat{s}^2(\tau)(\mathbf{X}'\mathbf{X})^-$$

where $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)'$ is the design matrix and $\hat{s}(\tau)$ is an estimate of $s(\tau)$. Under the iid assumption, the algorithm for computing $\hat{s}(\tau)$ is as follows:

1. Fit a quantile regression model and compute the residuals. Each residual $r_i = y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}(\tau)$ can be viewed as an estimated realization of the corresponding error ϵ_i.
2. Compute the quantile level bandwidth h_n. The HPQUANTSELECT procedure provides two bandwidth methods:
   - The Bofinger bandwidth is an optimizer of mean squared error for standard density estimation:

     $$h_n = n^{-1/5}(4.5v^2(\tau))^{1/5}$$

   - The Hall-Sheather bandwidth is based on Edgeworth expansions for studentized quantiles,

     $$h_n = n^{-1/3}z_\alpha^{2/3}(1.5v(\tau))^{1/3}$$

     z_α satisfies $T(z_\alpha, df) = 1 - \alpha/2$ for the construction of $1-\alpha$ confidence intervals, where T is the cumulative distribution function for the t distribution and df is the residual degrees of freedom.

   The quantity

   $$v(\tau) = \frac{s(\tau)}{s^{(2)}(\tau)} = \frac{f^2}{2(f^{(1)}/f)^2 + [(f^{(1)}/f)^2 - f^{(2)}/f]}$$

   is not sensitive to f and can be estimated by assuming f is Gaussian as

   $$\hat{v}(\tau) = \frac{\exp(-q^2)}{2\pi(q^2+1)}$$

   where $q = \Phi^{-1}(\tau)$.

3. Compute residual quantiles $\hat{F}^{-1}(\tau_0)$ and $\hat{F}^{-1}(\tau_1)$ as follows:

   a) Set $\tau_0 = \max(0, \tau - h_n)$ and $\tau_1 = \min(1, \tau + h_n)$.

   b) Use the equation

$$\hat{F}^{-1}(t) = \begin{cases} r_{(1)} & \text{if } t \in [0, 0.5/n) \\ \frac{0.5+(nt-i)}{n} r_{(i+1)} + \frac{0.5-(nt-i)}{n} r_{(i)} & \text{if } t \in [(i-0.5)/n, (i+0.5)/n) \\ r_{(n)} & \text{if } t \in [(n-0.5)/n, 1] \end{cases}$$

   where $r_{(i)}$ is the ith smallest residual.

   c) If $\hat{F}^{-1}(\tau_0) = \hat{F}^{-1}(\tau_1)$, find i that satisfies $r_{(i)} < \hat{F}^{-1}(\tau_0)$ and $r_{(i+1)} \geq \hat{F}^{-1}(\tau_0)$. If such an i exists, reset $\tau_0 = (i - 0.5)/n$ so that $\hat{F}^{-1}(\tau_0) = r_{(i)}$. Also find j that satisfies $r_{(j)} > \hat{F}^{-1}(\tau_1)$ and $r_{(j-1)} \leq \hat{F}^{-1}(\tau_1)$. If such a j exists, reset $\tau_1 = (j - 0.5)/n$ so that $\hat{F}^{-1}(\tau_1) = r_{(j)}$.

4. Estimate the sparsity function $s(\tau)$ as

$$\hat{s}(\tau) = \frac{\hat{F}^{-1}(\tau_1) - \hat{F}^{-1}(\tau_0)}{\tau_1 - \tau_0}$$

Linear-in-Parameter Model with Non-iid Settings

The general form of a linear quantile regression model is

$$Q_Y(\tau|\mathbf{x}) = \mathbf{x}'\boldsymbol{\beta}(\tau)$$

where the iid assumption is not necessary. Under some regularity conditions, the asymptotic distribution of the general form of quantile regression estimates is

$$\sqrt{n}(\hat{\boldsymbol{\beta}}(\tau) - \boldsymbol{\beta}(\tau)) \to N(0, \tau(1-\tau)\mathbf{H}_n^{-}\boldsymbol{\Omega}\mathbf{H}_n^{-})$$

where $\mathbf{H}_n = \lim_{n\to\infty} n^{-1} \sum \mathbf{x}_i \mathbf{x}_i' f_i(F_i^{-1}(\tau))$.

Accordingly, the covariance matrix of $\hat{\boldsymbol{\beta}}(\tau)$ can be estimated as

$$\hat{\Sigma}(\tau) = n^{-2}\tau(1-\tau)\hat{\mathbf{H}}_n^{-}(\mathbf{X}'\mathbf{X})\hat{\mathbf{H}}_n^{-}$$

where $\hat{\mathbf{H}}_n = n^{-1} \sum \left(\mathbf{x}_i \mathbf{x}_i' / \hat{s}_i(\tau)\right)$.

The sparsity function of the ith observation, $\hat{s}_i(\tau)$, can be estimated as

$$\hat{s}_i(\tau) = \frac{\hat{F}_i^{-1}(\tau + h_n) - \hat{F}_i^{-1}(\tau - h_n)}{2h_n}$$

where $\hat{F}_i^{-1}(\tau \pm h_n) = \mathbf{x}_i'\hat{\boldsymbol{\beta}}(\tau \pm h_n)$ are the ith predicted quantile values at quantile levels $(\tau \pm h_n)$.

More Statistics for Parameter Estimates

Let COV(·) denote the covariance function matrix of a random vector. Then, the sparsity-function estimates of COV $\left(\hat{\boldsymbol{\beta}}(\tau)\right)$ is

$$\widehat{\mathrm{COV}}\left(\hat{\boldsymbol{\beta}}(\tau)\right) = \begin{cases} \omega^2(\tau, F)\boldsymbol{\Omega}^{-1}/n & \text{for a linear model with iid errors} \\ \tau(1-\tau)\mathbf{H}_n^{-}\boldsymbol{\Omega}\mathbf{H}_n^{-}/n & \text{for a linear-in-parameter model with non-iid settings} \end{cases}$$

where $\hat{\boldsymbol{\beta}}(\tau) = \left(\hat{\beta}_1(\tau), \ldots, \hat{\beta}_p(\tau)\right)$ is the vector of the parameter estimates.

If you specify the CLB option in the MODEL statement, PROC HPQUANTSELECT outputs the standard error, confidence limits, t value, and Pr > $|t|$ probability for each $\hat{\beta}_j(\tau)$ in the parameter estimates table. Table 63.5 summarizes these statistics for $\hat{\beta}_j(\tau)$.

Table 63.5 More Statistics for $\hat{\beta}_j(\tau)$

Statistic	Definition
Standard error: $\hat{\sigma}_j$	$\sqrt{\widehat{\mathrm{COV}}\left(\hat{\boldsymbol{\beta}}(\tau)\right)_{jj}}$
$(1-\alpha)$% confidence limits	$\left(\hat{\beta}_j(\tau) \pm t_{1,1-\frac{\alpha}{2}}\hat{\sigma}_j\right)$
t value	$\hat{\beta}_j(\tau)/\hat{\sigma}_j$
Pr > $\lvert t\rvert$ probability	p-value of the t value

Here COV $\left(\hat{\boldsymbol{\beta}}(\tau)\right)_{jj}$ is the (j, j) element of COV $\left(\hat{\boldsymbol{\beta}}(\tau)\right)$, and $t_{1,1-\frac{\alpha}{2}}$ denotes the $\left(1-\frac{\alpha}{2}\right)$-level student's t score with 1 degree of freedom.

Criteria Used in Model Selection

The HPQUANTSELECT procedure supports the following fit statistics that you can use as criteria for the CHOOSE=, SELECT=, and STOP= options in the SELECTION statement:

ADJR1	specifies the adjusted R1 statistic.
AIC	specifies Akaike's information criterion (Akaike 1969; Koenker 2005).
AICC	specifies the corrected Akaike's information criterion (Hurvich and Tsai 1989).
BIC | SBC	specifies the Schwarz Bayesian information criterion (Schwarz 1978; Koenker 2005).
R1	specifies the R1 statistic (Koenker and Machado 1999). The R1 statistic is not valid for the STOP= or CHOOSE= option.
SL	specifies the significance level that is used to assess an effect's contribution to the fit when it is added to or removed from a model. SL is not valid for the CHOOSE= option.
VALIDATE	specifies the average check loss over the validation data.

Quasi-likelihood Information Criteria

Given the quantile level τ, assume that the distribution of Y_i conditional on $\mathbf{x}_i$ follows the linear model

$$Y_i = \mathbf{x}_i'\boldsymbol{\beta} + \epsilon_i$$

where ϵ_i for $i = 1, \ldots, n$ are iid in distribution F. Further assume that F is an asymmetric Laplace distribution whose density function is

$$f_\tau(r) = \frac{\tau(1-\tau)}{\sigma} \exp\left(-\frac{\rho_\tau(r)}{\sigma}\right)$$

where σ is the scale parameter. Then, the negative log-likelihood function is

$$l_\tau(\boldsymbol{\beta}, \sigma) = n \log(\sigma) + \sigma^{-1} \sum_{i=1}^{n} \rho_\tau(y_i - \mathbf{x}_i'\boldsymbol{\beta}) - n \log(\tau(1-\tau))$$

Under these settings, the maximum likelihood estimate (MLE) of $\boldsymbol{\beta}$ is the same as the relevant level-τ quantile regression solution $\hat{\boldsymbol{\beta}}(\tau)$, and the MLE for σ is

$$\hat{\sigma}(\tau) = n^{-1} \sum_{i=1}^{n} \rho_\tau\left(y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}(\tau)\right)$$

where $\hat{\sigma}(\tau)$ equals the level-τ average check loss $\mathrm{ACL}(\tau)$ for the quantile regression solution.

Because the general form of Akaike's information criterion (AIC) is $\mathrm{AIC} = (-2l + 2p)$, the quasi-likelihood AIC for quantile regression is

$$\mathrm{AIC}(\tau) = 2n \ln\left(\mathrm{ACL}(\tau)\right) + 2p$$

where p is the degrees of freedom for the fitted model.

Similarly, the quasi-likelihood AICC (corrected AIC) and SBC (Schwarz Bayesian information criterion) can be formulated as follows:

$$\mathrm{AICC}(\tau) = 2n \ln\left(\mathrm{ACL}(\tau)\right) + \frac{2pn}{n-p-1}$$

$$\mathrm{SBC}(\tau) = 2n \ln\left(\mathrm{ACL}(\tau)\right) + p \ln(n)$$

In fact, the quasi-likelihood AIC, AICC, and SBC are fairly robust, and you can use them to select effects for data sets without the iid assumption in asymmetric Laplace distribution. For a simulation study that applies SBC for effect selection, see "Example 63.1: Simulation Study" on page 4942. The study generates a data set by using a naive instrumental model (Chernozhukov and Hansen 2008).

Statistical Tests for Significance Level

The HPQUANTSELECT procedure supports the significance level (SL) criterion for effect selection. Consider the general form of a linear quantile regression model:

$$Q_Y(\tau|\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1'\boldsymbol{\beta}_1(\tau) + \mathbf{x}_2'\boldsymbol{\beta}_2(\tau)$$

At each step of an effect-selection process, a candidate effect can be represented as $\mathbf{x}_2$, and the significance level of the candidate effect can be calculated by testing the null hypothesis: $H_0 : \boldsymbol{\beta}_2(\tau) = \mathbf{0}$.

When you use SL as a criterion for effect selection, you can further use the TEST= option in the SELECTION statement to specify a statistical test method to compute the significance-level values as follows:

- The TEST=WALD option specifies the Wald test. Let $\hat{\boldsymbol{\beta}}(\tau) = \left(\hat{\boldsymbol{\beta}}_1'(\tau), \hat{\boldsymbol{\beta}}_2'(\tau)\right)'$ be the parameter estimates for the extended model, and denote the estimated covariance matrix of $\hat{\boldsymbol{\beta}}(\tau)$ as

  $$\hat{\Sigma}(\tau) = \begin{bmatrix} \hat{\Sigma}_{11}(\tau) & \hat{\Sigma}_{12}(\tau) \\ \hat{\Sigma}_{21}(\tau) & \hat{\Sigma}_{22}(\tau) \end{bmatrix}$$

  where $\hat{\Sigma}_{22}(\tau)$ is the covariance matrix for $\hat{\boldsymbol{\beta}}_2(\tau)$. Then the Wald test score is defined as

  $$\hat{\boldsymbol{\beta}}_2'(\tau)\hat{\Sigma}_{22}^{-1}(\tau)\hat{\boldsymbol{\beta}}_2(\tau)$$

  If you specify the SPARSITY(IID) option in the MODEL statement, $\hat{\Sigma}(\tau)$ is estimated under the iid errors assumption. Otherwise, $\hat{\Sigma}(\tau)$ is estimated by using non-iid settings. For more information about the linear model with iid errors and non-iid settings, see the section "Quantile Regression" on page 4925.

- The TEST=LR1 or TEST=LR2 option specifies the Type I or Type II quasi-likelihood ratio test, respectively. Under the iid assumption, Koenker and Machado (1999) propose two types of quasi-likelihood ratio tests for quantile regression, where the error distribution is flexible but not limited to the asymmetric Laplace distribution. The Type I test score, LR1, is defined as

  $$\frac{2(D_1(\tau) - D_2(\tau))}{\tau(1-\tau)\hat{s}}$$

  where $D_1(\tau) = \sum \rho_\tau \left(y_i - \mathbf{x}_{1i}\hat{\boldsymbol{\beta}}_{1_1}(\tau)\right)$ is the sum of check losses for the reduced model, $D_2(\tau) = \sum \rho_\tau \left(y_i - \mathbf{x}_{1i}\hat{\boldsymbol{\beta}}_{1_2}(\tau) - \mathbf{x}_{2i}\hat{\boldsymbol{\beta}}_2(\tau)\right)$ is the sum of check losses for the extended model, and $\hat{s}$ is the estimated sparsity function. The Type II test score, LR2, is defined as

  $$\frac{2D_2(\tau)\left(\log(D_1(\tau)) - \log(D_2(\tau))\right)}{\tau(1-\tau)\hat{s}}$$

Under the null hypothesis that the reduced model is the true model, the Wald score, LR1 score, and LR2 score all follow a χ^2 distribution with degrees of freedom $df = df_2 - df_1$, where df_1 and df_2 are the degrees of freedom for the reduced model and the extended model, respectively .

When you use SL as a criterion for effect selection, the algorithm for estimating sparsity function depends on whether an effect is being considered as an add or a drop candidate. For testing an add candidate effect, the sparsity function, which is $s(\tau)$ under the iid error assumption or $s_i(\tau)$ for non-iid settings, is estimated on the reduced model that does not include the add candidate effect. For testing a drop candidate effect, the sparsity function is estimated on the extended model that does not exclude the drop candidate effect. Then, these estimated sparsity function values are used to compute LR1 or LR2 and the covariance matrix of the parameter estimates for the extended model. However, for the model that is selected at each step, the sparsity function for estimating standard errors and confidence limits of the parameter estimates is estimated on that model itself, but not on the model that was selected at the preceding step.

Because the null hypotheses usually do not hold, the SLENTRY and SLSTAY values cannot reliably be viewed as probabilities. One way to address this difficulty is to replace hypothesis testing as a means of selecting a model with information criteria or out-of-sample prediction criteria.

Table 63.6 provides formulas and definitions for these fit statistics.

Table 63.6 Formulas and Definitions for Model Fit Summary Statistics for Single Quantile Effect Selection

Statistic	Definition or Formula
n	Number of observations
p	Number of parameters, including the intercept
$r_i(\tau)$	Residual for the ith observation; $r_i(\tau) = y_i - \mathbf{x}_i\hat{\boldsymbol{\beta}}(\tau)$
$D(\tau)$	Total sum of check losses; $D(\tau) = \sum_{i=1}^{n} \rho_\tau(r_i)$. $D(\tau)$ is labeled as Objective Function in the "Fit Statistics" table.
$D_0(\tau)$	Total sum of check losses for intercept-only model if the intercept is a forced-in effect; otherwise for empty model.
$\mathrm{ACL}(\tau)$	Average check loss; $\mathrm{ACL}(\tau) = \dfrac{D(\tau)}{n}$
$\mathrm{R1}(\tau)$	Counterpart of linear regression R square for quantile regression; $\mathrm{R1}(\tau) = 1 - \dfrac{D(\tau)}{D_0(\tau)}$
$\mathrm{ADJR1}(\tau)$	Adjusted R1; $1 - \dfrac{(n-1)D(\tau)}{(n-p)D_0(\tau)}$ if intercept is a forced-in effect; otherwise $1 - \dfrac{nD(\tau)}{(n-p)D_0(\tau)}$.
$\mathrm{AIC}(\tau)$	$2n\ln(\mathrm{ACL}(\tau)) + 2p$
$\mathrm{AICC}(\tau)$	$2n\ln(\mathrm{ACL}(\tau)) + \dfrac{2pn}{n-p-1}$
$\mathrm{SBC}(\tau)$	$2n\ln(\mathrm{ACL}(\tau)) + p\ln(n)$

The $\mathrm{ADJR1}(\tau)$ criterion is equivalent to the generalized approximate cross validation (GACV) criterion for quantile regression (Yuan 2006). The GACV criterion is defined as

$$\mathrm{GACV}(\tau) = D(\tau)/(n-p)$$

which is proportional to $1 - \mathrm{ADJR1}(\tau)$.

Diagnostic Statistics

This section gathers the formulas for the statistics available in the OUTPUT statement. All the statistics available in the OUTPUT statement are conditional on the selected model and do not take into account the variability that is introduced by doing model selection.

The model to be fit is $Q_Y(\tau|\mathbf{x}) = \mathbf{x}'\boldsymbol{\beta}(\tau)$, and the parameter estimate $\hat{\boldsymbol{\beta}}(\tau)$ is the solution that minimizes $\sum_{i=1}^{n} \rho_\tau\left(y_i - \mathbf{x}_i'\boldsymbol{\beta}\right)$. The subscript i is for the ith observation. The subscript j is for the jth-smallest quantile level among all the specified QUANTILE= levels in the MODEL statement. $\hat{\Sigma}(\tau)$ denotes the covariance estimation for $\hat{\boldsymbol{\beta}}(\tau)$.

The ALPHA= option in the PROC HPQUANTSELECT statement sets the α value for the confidence limit statistics. The degrees of freedom for $t_{\frac{\alpha}{2}}$ are $n - p$.

Table 63.7 contains the diagnostic statistics and their formulas. Each statistic is computed for each observation.

Table 63.7 Formulas and Definitions for Diagnostic Statistics

MODEL Option or Statistic	Formula
PREDj	$\widehat{y}_{ji} = \mathbf{x}_i'\hat{\boldsymbol{\beta}}(\tau_j)$
RESj	$y_i - \widehat{y}_{ji}$
STDPj	$\sqrt{\mathbf{x}_i'\hat{\Sigma}(\tau_j)\mathbf{x}_i}$
LCLMj	$\widehat{y}_{ji} - t_{\frac{\alpha}{2}}\text{STDP}j_i$
UCLMj	$\widehat{y}_{ji} + t_{\frac{\alpha}{2}}\text{STDP}j_i$

Classification Variables and the SPLIT Option

PROC HPQUANTSELECT supports the ability to split classification variables when you do model selection. You use the SPLIT option in the CLASS statement to specify that the columns of the design matrix that correspond to effects that contain a split classification variable can enter or leave a model independently of the other design columns of that effect. The following statements illustrate the use of the SPLIT option:

```
data splitExample;
   length C2 $6;
   drop i;
   do i=1 to 1000;
     C1 = 1 + mod(i,6);
     if      i < 250 then C2 = 'Low';
     else if i < 500 then C2 = 'Medium';
     else                 C2 = 'High';
     x1 = ranuni(1);
     x2 = ranuni(1);
     y = x1+3*(C2 ='low')  + 10*(C1=3) +5*(C1=5) + rannor(1);
     output;
   end;
run;

proc hpquantselect data=splitExample;
   class C1(split) C2(order=data);
   model y = C1 C2 x1 x2/orderselect clb;
   selection method=forward;
run;
```

The "Class Levels" table in Figure 63.14 is produced by default whenever you specify a CLASS statement.

Figure 63.14 Class Levels

The HPQUANTSELECT Procedure

Class Level Information		
Class	Levels	Values
C1	6	* 1 2 3 4 5 6
C2	3	Low Medium High

*** Associated Parameters Split**

The SPLIT option has been specified for the classification variable C1. This permits the parameters that are associated with the effect C1 to enter or leave the model individually. The "Parameter Estimates" table in Figure 63.15 shows that for this example the parameters that correspond to only levels 3 and 5 of C1 are in the selected model. Finally, note that the ORDERSELECT option in the MODEL statement displays the parameters in the order in which they first entered the model.

Figure 63.15 Parameter Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		t Value	Pr > |t|
Intercept	1	-0.21596	0.09024	-0.39304	-0.03887	-2.39	0.0169
C1_3	1	10.08952	0.09852	9.89619	10.28285	102.41	<.0001
C1_5	1	5.04115	0.10835	4.82854	5.25376	46.53	<.0001
x1	1	1.29863	0.14014	1.02363	1.57363	9.27	<.0001

Macro Variables That Contain Selected Effects

PROC HPQUANTSELECT saves the list of selected effects in a macro variable for each selected model so that you can use other SAS procedures to perform postselection analyses. This list does not explicitly include the intercept so that you can use it in the MODEL statement of other SAS/STAT regression procedures.

When multiple quantile levels or BY processing are used, one macro variable, indexed by the quantile level order and the BY group number (as shown in Table 63.8), is created for each quantile level and BY group combination.

Table 63.8 Macro Variables Created for Subsequent Processing

Macro Variable	Description
Single Quantile Level and No BY Processing	
_HPQRSIND	Selected model
Multiple Quantile Levels and No BY Processing	
_HPQRSINDT1	Selected model for the first quantile level
_HPQRSINDT2	Selected model for the second quantile level
...	
Single Quantile Level and BY Processing	
_HPQRSIND1	Selected model for BY group 1
_HPQRSIND2	Selected model for BY group 2
...	
Multiple Quantile Levels and BY Processing	
_HPQRSIND1T1	Selected model for the first quantile level and BY group 1
_HPQRSIND1T2	Selected model for the second quantile level and BY group 1
...	
_HPQRSIND2T1	Selected model for the first quantile level and BY group 2
_HPQRSIND2T2	Selected model for the second quantile level and BY group 2
...	

The macro variables _HPQRSIND, _HPQRSINDT1, _HPQRSIND1, and _HPQRSIND1T1 are all synonyms.

The following statements generate a simulation data set, use PROC HPQUANTSELECT to select a median regression model, and print the macro variables for the selected model:

```
%let seed=321;
%let p=20;
%let n=3000;

data analysisData;
   array x{&p} x1-x&p;
   do i=1 to &n;
      do j=1 to &p;
         x{j} = ranuni(&seed);
      end;
      e  = ranuni(&seed);
      y  = x1 + x2 + x3 + e;
      output;
   end;
run;

proc hpquantselect data=analysisData;
   model y = x1-x&p;
   selection method=forward;
run;

%put _HPQRSIND    = &_hpqrsind;
%put _HPQRSIND1   = &_hpqrsind1;
%put _HPQRSIND1T1 = &_hpqrsind1t1;
```

The following statements use PROC HPREG to fit a linear regression model on the effects that are selected by the HPQUANTSELECT procedure:

```
proc hpreg data=analysisData;
   model y = &_hpqrsind;
run;
```

Using Validation and Test Data

When you have sufficient data, you can subdivide your data into three parts, called the training, validation, and test data. During the selection process, models are fit on the training data, and the prediction error for the models is found by using the validation data. This prediction error on the validation data can be used to decide when to terminate the selection process or what effects to include as the selection process proceeds. Finally, after you have obtained a selected model, you can use the test set to assess how the selected model generalizes on data that played no role in selecting the model.

In some cases, you might want to use only training and test data. For example, you might decide to use an information criterion to decide what effects to include and when to terminate the selection process. In this case, no validation data are required, but test data can still be useful in assessing the predictive performance of the selected model. In other cases, you might decide to use validation data during the selection process but forgo assessing the selected model on test data. Hastie, Tibshirani, and Friedman (2001) state that it is

difficult to give a general rule for how many observations you should assign to each role. They note that a typical split might be 50% for training and 25% each for validation and testing.

You use a PARTITION statement to logically subdivide the DATA= data set into separate roles. You can name the fractions of the data that you want to reserve as test data and validation data. For example, the following statements randomly subdivide the inData data set, reserving 50% for training and 25% each for validation and testing:

```
proc hpquantselect data=inData;
  partition fraction(test=0.25 validate=0.25);
  ...
run;
```

In some cases, you might need to exercise more control over the partitioning of the input data set. You can do this by naming both a variable in the input data set and a formatted value of that variable that correspond to each role. For example, the following statements assign roles to the observations in the inData data set based on the value of the variable Group in that data set. Observations in which the value of Group is `'group 1'` are assigned for testing, and those whose value is `'group 2'` are assigned to training. All other observations are ignored.

```
proc hpquantselect data=inData;
  partition roleVar=Group(test='group 1' train='group 2')
  ...
run;
```

After you reserve observations for training, validation, and testing, a model fit on the training data is scored on the validation and test data, and the average check loss (ACL) is computed separately for each of these subsets. The ACL for each data role is the error sum of squares for observations in that role divided by the number of observations in that role.

Using the Validation ACL as the STOP= Criterion

If you have provided observations for validation, then you can specify STOP=VALIDATE as a suboption of the METHOD= option in the SELECTION statement. At step k of the selection process, the best candidate effect to enter or leave the current model is determined. Here "best candidate" means the effect that gives the best value of the SELECT= criterion; this criterion need not be based on the validation data. The validation ACL is computed for the model in which this candidate effect is added or removed. If this validation ACL is greater than the validation ACL for the model at step k, then the selection process terminates at step k.

Using the Validation ACL as the CHOOSE= Criterion

When you specify the CHOOSE=VALIDATE suboption of the METHOD= option in the SELECTION statement, the validation ACL is computed for the models at each step of the selection process. The smallest model at any step that yields the smallest validation ACL is selected.

Using the Validation ACL as the SELECT= Criterion

You request the validation ACL as the selection criterion by specifying the SELECT=VALIDATE suboption of the METHOD= option in the SELECTION statement. At step k of the selection process, the validation ACL is computed for each model in which a candidate for entry is added or a candidate for removal is dropped. The selected candidate for entry or removal is the one that yields a model that has the minimal validation ACL.

Computational Method

Multithreading

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

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

- You can specify the CPU count by using the CPUCOUNT= SAS system option. For example, if you specify the following statement, the HPQUANTSELECT procedure schedules threads as if it were executing on a system that had at most four CPUs:

```
options cpucount=4;
```

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

The number of threads is displayed in the "Performance Information" table, which is part of the default output. The HPQUANTSELECT procedure allocates one thread per CPU.

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

- variable levelization
- effect levelization
- formation of the crossproducts matrix
- quantile regression model fitting
- estimation of covariance matrix for parameter estimates
- evaluation of predicted residual sums of check losses on validation and test data
- scoring of observations

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

Output Data Set

Many SAS procedures add the variables from the input data set when an observationwise output data set is created. The assumption of high-performance analytical procedures is that the input data sets can be large and contain many variables. For performance reasons, the output data set contains the following:

- variables that are explicitly created by the statement
- variables that are listed in the ID statement or specified by using the COPYVAR= option
- distribution keys or hash keys that are transferred from the input data set

The high-performance analytical procedures enable you to add output data set information that is necessary for subsequent SQL joins without copying the entire input data set to the output data set. For further details about output data sets when PROC HPQUANTSELECT is run in distributed mode, see the section "Output Data Sets" (Chapter 2, *SAS/STAT User's Guide: High-Performance Procedures*).

Displayed Output

The following sections describe the output that PROC HPQUANTSELECT produces. The output is organized into various tables, which are discussed in their order of appearance.

Performance Information

The "Performance Information" table is produced by default and displays information about the grid host for distributed execution and about whether PROC HPQUANTSELECT executes in single-machine mode, distributed mode, or alongside-the-database mode. The numbers of compute nodes and threads are also displayed, depending on the environment.

Data Access Information

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

Model Information

The "Model Information" table displays basic information about the model, such as the response variable, the weight variable, and the type of parameterization that is used for classification variables named in the CLASS statement.

Selection Information

When you specify the SELECTION statement, by default the HPQUANTSELECT procedure produces a series of tables that contain information about the model selection. The "Selection Information" table informs you about the model selection method; select, stop, and choose criteria; and other parameters that govern the selection. You can suppress this table by specifying DETAILS=NONE in the SELECTION statement.

Number of Observations

The "Number of Observations" table displays the number of observations that are read from the input data set and the number of observations that are used in the analysis. If you use a PARTITION statement, the table also displays the number of observations that are used for each data role.

Class Level Information

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

If the classification variables are in the reference parameterization, the "Class Level Information" table also displays the reference value for each variable. This table also indicates which, if any, of the classification variables are split by using the SPLIT option in the CLASS statement.

Dimensions

The "Dimensions" table displays information about the number of effects and the number of parameters from which the selected model is chosen. If you use split classification variables, then this table also includes the number of effects after splitting is taken into account.

Entry and Removal Candidates

When you specify the DETAILS=ALL or DETAILS=STEPS option in the SELECTION statement, the HPQUANTSELECT procedure produces "Entry Candidates" and "Removal Candidates" tables that display the effect names and values of the criterion used to select entering or departing effects at each step of the selection process. The effects are displayed in order from best to worst of the selection criterion.

Selection Summary

When you specify the SELECTION statement, the HPQUANTSELECT procedure produces the "Selection Summary" table, which contains information about the sequence of steps of the selection process. For each step, the effect that was entered or dropped is displayed along with the statistics used to select the effect, stop the selection, and choose the selected model. For all criteria that you can use for model selection, the steps at which the optimal values of these criteria occur are also indicated.

You can suppress the "Selection Summary" table by specifying DETAILS=NONE in the SELECTION statement.

Stop Reason

The "Stop Reason" table displays the reason why the selection stopped. To facilitate programmatic use of this table, an integer code is assigned to each reason and is included if you output this table by using an ODS OUTPUT statement. The reasons and their associated codes follow:

Code	Stop Reason
1	All eligible effects are in the model.
2	All eligible effects have been removed.
3	Specified maximum number of steps have been done.
4	The model contains the specified maximum number of effects.
5	The model contains the specified minimum number of effects (for backward selection).
6	The stopping criterion is at a local optimum.
7	No suitable add or drop candidate could be found.
8	Adding or dropping any effect does not improve the selection criterion.
9	No candidate meets the appropriate SLE or SLS significance level.
10	Stepwise selection is cycling.
11	The model is an exact fit.
12	Dropping an effect would result in an empty model.

You can suppress the "Stop Reason" table by specifying DETAILS=NONE in the SELECTION statement.

Selection Reason

When you specify the SELECTION statement, the HPQUANTSELECT procedure produces a simple table that contains text that explains why the final model was selected.

You can suppress the "Selection Reason" table by specifying DETAILS=NONE in the SELECTION statement.

Selected Effects

When you specify the SELECTION statement, the HPQUANTSELECT procedure produces a simple table that contains text that lists the selected effects in the final model.

Fit Statistics

The "Fit Statistics" table displays fit statistics for the selected model. The statistics include the following:

- Objective Function, total sum of check losses. Objective Function is denoted as $D(\tau)$ in Table 63.6.
- R1, a measure between 0 and 1 that indicates the portion of the (corrected) total check losses attributed to the fit rather than left to residuals. It is calculated as $1 - \frac{D(\tau)}{D_0(\tau)}$. It is the quantile regression counterpart of the linear regression R square.
- Adj R1, the adjusted R1, a version of R1 that has been adjusted for degrees of freedom. It is calculated as

$$\text{Adj R1} = 1 - \frac{n-i}{n-p}(1 - \text{R1})$$

where $i = 1$ if the intercept is forced in and $i = 0$ otherwise , n is the number of observations used to fit the model, and p is the number of parameters in the model.

- the fit criteria AIC, AICC, and SBC if they are used in the selection process. For the formulas to evaluate these criteria, see Table 63.6.
- the average check loss (ACL) on the training, validation, and test data

You can request the "Fit Statistics" tables for the model at each step of the selection process by specifying the DETAILS= option in the SELECTION statement.

Parameter Estimates

The "Parameter Estimates" table displays the parameters in the selected model and their estimates. The information that is displayed for each parameter in the selected model includes the following:

- the parameter label, which includes the effect name and level information for effects that contain classification variables
- the degrees of freedom (DF) for the parameter. There is one degree of freedom unless the model is not full rank.
- the parameter estimate
- the standardized parameter estimate. PROC HPQUANTSELECT outputs the standardized parameter estimate only if you specify the STB option in the MODEL statement.

If you specify the CLB option in the MODEL statement, PROC HPQUANTSELECT also outputs the following information:

- the standard error, which is the estimate of the standard deviation of the parameter estimate
- the $100(1-\alpha)\%$ confidence limits for the parameter estimate
- t value, the t test that the parameter is 0. This is computed as the parameter estimate divided by the standard error.
- the $\Pr > |t|$, the probability that a t statistic would obtain a greater absolute value than that observed when the true parameter is 0. This is the two-tailed significance probability.

For more information about standard errors, confidence limits, t values, and the $\Pr > |t|$ probability, see the section "More Statistics for Parameter Estimates" on page 4928.

You can request "Parameter Estimates" tables for the model at each step of the selection process by specifying the DETAILS= option in the SELECTION statement.

Timing Information

If you specify the DETAILS option in the PERFORMANCE statement, PROC HPQUANTSELECT also produces a "Timing" table that displays the elapsed time for each main task of PROC HPQUANTSELECT.

ODS Table Names

Each table that the HPQUANTSELECT procedure creates has a name associated with it. You must use this name to refer to the table when you use ODS statements. These names are listed in Table 63.9.

Table 63.9 ODS Tables Produced by PROC HPQUANTSELECT

Table Name	Description	Required Statement / Option
ClassLevels	Level information from the CLASS statement	CLASS
DataAccessInfo	Information about modes of data access	Default output
Dimensions	Model dimensions	Default output
EntryCandidates	Candidates for entry at step	SELECTION DETAILS=ALL | STEPS
FitStatistics	Fit statistics	Default output
ModelInfo	Information about the modeling environment	Default output
NObs	Number of observations read and used	Default output
ParameterEstimates	Solutions for the parameter estimates associated with effects in the MODEL statement	Default output
PerformanceInfo	Information about high-performance computing environment	Default output
RemovalCandidates	Candidates for removal at step	SELECTION DETAILS=ALL | STEPS
SelectedEffects	List of selected effects	SELECTION
SelectionInfo	Information about selection settings	Default output
SelectionReason	Reason for selecting the final model	SELECTION
SelectionSummary	Summary information about the model selection steps	SELECTION
StopReason	Reason selection was terminated	SELECTION
Timing	Timing breakdown by task	PERFORMANCE DETAILS

Examples: HPQUANTSELECT Procedure

Example 63.1: Simulation Study

This example is based on "Example 101.1: Simulation Study" on page 8400. This simulation study shows how you can use the forward selection method to select quantile regression models for single quantile levels. The following statements simulate a data set from a naive instrumental model (Chernozhukov and Hansen 2008):

```
%let seed=321;
%let p=20;
%let n=3000;

data analysisData;
   array x{&p} x1-x&p;
   do i=1 to &n;
      U  = ranuni(&seed);
      x1 = ranuni(&seed);
      x2 = ranexp(&seed);
      x3 = abs(rannor(&seed));
      y  = x1*(U-0.1) + x2*(U*U-0.25) + x3*(exp(U)-exp(0.9));
      do j=4 to &p;
         x{j} = ranuni(&seed);
      end;
      output;
   end;
run;
```

Variable U in the data set indicates the true quantile level of the response y conditional on $\mathbf{x} = (x_1, \ldots, x_p)$.

Let $Q_Y(\tau|\mathbf{x}) = \mathbf{x}\boldsymbol{\beta}(\tau)$ denote the underlying quantile regression model, where $\boldsymbol{\beta}(\tau) = (\beta_1(\tau), \ldots, \beta_p(\tau))'$. Then, the true parameter functions are

$$\begin{aligned}
\beta_1(\tau) &= \tau - 0.1 \\
\beta_2(\tau) &= \tau^2 - 0.25 \\
\beta_3(\tau) &= \exp(\tau) - \exp(0.9) \\
\beta_4(\tau) &= \cdots = \beta_p(\tau) = 0
\end{aligned}$$

It is easy to see that, at $\tau = 0.1$, only $\beta_2(0.1) = -0.24$ and $\beta_3(0.1) = \exp(0.1) - \exp(0.9) \approx -1.354432$ are nonzero parameters. Therefore, an effective effect-selection method should select x2 and x3 and drop all the other effects in this data set at $\tau = 0.1$. By the same rationale, x1 and x3 should be selected at $\tau = 0.5$ with $\beta_1(0.5) = 0.4$ and $\beta_3(0.5) \approx -0.810882$, and x1 and x2 should be selected at $\tau = 0.9$ with $\beta_1(0.9) = 0.8$ and $\beta_2(0.9) = 0.56$.

The following statements use PROC HPQUANTSELECT with the forward selection method. The STB option and the CLB option in the MODEL statement request the standardized parameter estimates and the confidence limits of parameter estimates, respectively.

```
proc hpquantselect data=analysisData;
   model y= x1-x&p / quantile=0.1 0.5 0.9 stb clb;
   selection method=forward;
   output out=out p=pred;
run;
```

Output 63.1.1 shows that, by default, the CHOOSE= and STOP= options are both set to SBC.

Output 63.1.1 Model Information

The HPQUANTSELECT Procedure

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

Output 63.1.2, Output 63.1.3, and Output 63.1.4 display the selected effects and the parameter estimates for $\tau = 0.1$, $\tau = 0.5$, and $\tau = 0.9$, respectively. You can see that the forward selection method correctly selects active effects for all three quantile levels.

Output 63.1.2 Parameter Estimates at $\tau = 0.1$

The HPQUANTSELECT Procedure

Quantile Level = 0.1
Selected Model

Selected Effects:	Intercept x2 x3

Parameter Estimates

Parameter	DF	Estimate	Standardized Estimate	Standard Error	95% Confidence Limits		t Value	Pr > |t|
Intercept	1	0.01179	0	0.01192	-0.01158	0.03516	0.99	0.3225
x2	1	-0.22871	-0.21829	0.00946	-0.24725	-0.21017	-24.19	<.0001
x3	1	-1.37991	-0.78452	0.01556	-1.41042	-1.34939	-88.67	<.0001

Output 63.1.3 Parameter Estimates at $\tau = 0.5$

The HPQUANTSELECT Procedure

Quantile Level = 0.5
Selected Model

Selected Effects:	Intercept x1 x3

Output 63.1.3 *continued*

Parameter Estimates

Parameter	DF	Estimate	Standardized Estimate	Standard Error	95% Confidence Limits		t Value	Pr > |t|
Intercept	1	0.01178	0	0.03418	-0.05524	0.07879	0.34	0.7304
x1	1	0.42584	0.11879	0.06237	0.30355	0.54814	6.83	<.0001
x3	1	-0.86332	-0.49082	0.04765	-0.95674	-0.76989	-18.12	<.0001

Output 63.1.4 Parameter Estimates at $\tau = 0.9$

The HPQUANTSELECT Procedure

Quantile Level = 0.9
Selected Model

Selected Effects: Intercept x1 x2

Parameter Estimates

Parameter	DF	Estimate	Standardized Estimate	Standard Error	95% Confidence Limits		t Value	Pr > |t|
Intercept	1	-0.00774	0	0.03292	-0.07228	0.05680	-0.24	0.8142
x1	1	0.78294	0.21841	0.05134	0.68228	0.88360	15.25	<.0001
x2	1	0.57644	0.55018	0.03422	0.50935	0.64354	16.85	<.0001

Example 63.2: Growth Charts for Body Mass Index

This example is modeled on the example in the section "Getting Started: QUANTSELECT Procedure" on page 8347. It highlights the use of the HPQUANTSELECT procedure for multiple-level quantile regression by creating growth charts for men's body mass index (BMI).

BMI, which is defined as the ratio of weight (kg) to squared height (m^2), is a standard measure for categorizing individuals as overweight or underweight. The percentiles of BMI for specified ages are of particular interest. This example draws smooth BMI quantile curves conditional on Age, which can serve as BMI growth charts in medical diagnosis to identify BMI percentiles for subjects.

The BMIMen data set is from the 1999–2000 and 2001–2002 survey results for men that are published by the National Center for Health Statistics. It contains the two variables BMI and Age with 3,264 observations.

```
data bmimen;
   input BMI Age @@;
   SqrtAge = sqrt(Age);
   InveAge = 1/Age;
   LogBMI  = log(BMI);
   datalines;
18.6  2.0 17.1  2.0 19.0  2.0 16.8  2.0 19.0  2.1  15.5   2.1
16.7  2.1 16.1  2.1 18.0  2.1 17.8  2.1 18.3  2.1  16.9   2.1
15.9  2.1 20.6  2.1 16.7  2.1 15.4  2.1 15.9  2.1  17.7   2.1

   ... more lines ...

29.0 80.0 24.1 80.0 26.6 80.0 24.2 80.0 22.7 80.0  28.4  80.0
26.3 80.0 25.6 80.0 24.8 80.0 28.6 80.0 25.7 80.0  25.8  80.0
22.5 80.0 25.1 80.0 27.0 80.0 27.9 80.0 28.5 80.0  21.7  80.0
33.5 80.0 26.1 80.0 28.4 80.0 22.7 80.0 28.0 80.0  42.7  80.0
;
```

The logarithm of BMI is used as the response. (Although this approach does not improve the quantile regression fit, it helps with statistical inference.) The following statements fit quantile regression models for the BMIMen data set at 10 quantile levels:

```
%let quantile=0.03 0.05 0.1 0.25 0.5 0.75 0.85 0.90 0.95 0.97;
%let nq=10;

proc hpquantselect data=BMIMen;
   model logBMI = InveAge SqrtAge Age SqrtAge*Age Age*Age Age*Age*Age
         / quantile=&quantile;
   code file='bmicode.sas';
   output out=Bmiout copyvars=(BMI Age) pred=P_LogBMI;
run;
```

The CODE statement enables you to write a SAS DATA step to compute quantile predictions of the fitted model. The OUTPUT statement outputs the mean predicted quantiles for the 10 specified quantile levels. The PRED= option in the OUTPUT statement specifies the variable names for the quantile predictions. For examples, p1 is for quantile level 0.03, and p2 is for quantile level 0.05.

The following statements define and apply a SAS macro function to create a quantile curves plot for the BMIMen data set:

```
%let BMIcolor=red olive orange blue brown gray violet black gold green;

%macro plotBMI;
   data BmiPred;
      set Bmiout;
      %do j=1 %to &nq;
         predBMI&j = exp(P_LogBMI&j);
      %end;
      label %do j=1 %to &nq;
               predBMI&j=%qscan(&quantile,&j,%str( ))
            %end;;
   run;
```

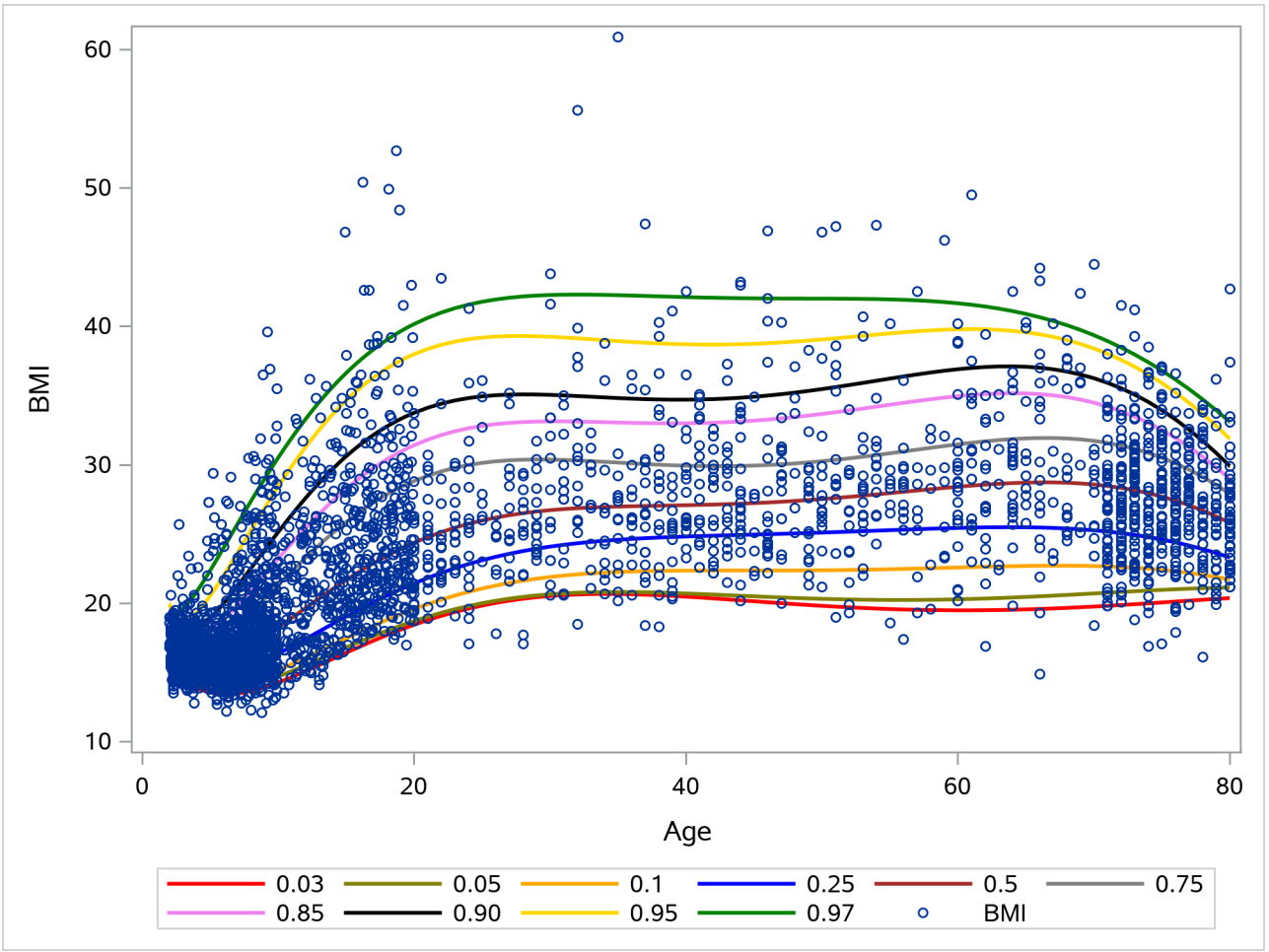
```
   proc sort data=BmiPred;
      by Age;
   run;

   proc sgplot data=BmiPred;
      %do j=1 %to &nq;
         series y=predBMI&j x=Age/lineattrs=(thickness=2
         color=%qscan(&BMIcolor,&j,%str( )));
      %end;
      scatter y=BMI x=Age/markerattrs=(size=5);
   run;
%mend;

%plotBMI;
```

Figure 63.2.1 shows the BMI quantile curves, which can serve as BMI growth charts. For example, the percentiles of any observations (small blue circles) that are located between the top 0.95 quantile (gold) curve and the 0.97 quantile (green) curve are between the 95th percentile and the 97th percentile. By using this rule, you can measure the percentile range for any observations of interest.

Output 63.2.1 Growth Chart for Body Mass Index



Other than using the OUTPUT statement, you can also calculate quantile predictions by using the CODE statement. The following statements show how to use the SAS DATA step and the SAS file *bmicode*, which the CODE statement requests, to calculate quantile predictions for the BMIMen data set:

```
data Newmen;
   set BMIMen;
   %inc bmicode;
run;
```

The SET statement in the SAS DATA step specifies a data set for computing quantile predictions. This is usually a new data set that you want to score. This example uses the BMIMen data set again, so the quantile predictions in the Newmen data set are identical to those in the Bmiout data set. The following statements compare the Bmiout data set with the Newmen data set:

```
proc compare data=Bmiout compare=Newmen criterion=0.00001;
run;
```

References

Akaike, H. (1969). "Fitting Autoregressive Models for Prediction." *Annals of the Institute of Statistical Mathematics* 21:243–247.

Chernozhukov, V., and Hansen, C. (2008). "Instrumental Variable Quantile Regression: A Robust Inference Approach." *Journal of Econometrics* 142:379–398.

Collier Books (1987). *The 1987 Baseball Encyclopedia Update*. New York: Macmillan.

Gertz, E. M., and Griffin, J. D. (2005). *Support Vector Machine Classifiers for Large Data Sets*. Technical Report ANL/MCS-TM-289, Mathematics and Computer Science Division, Argonne National Laboratory.

Gertz, E. M., and Griffin, J. D. (2010). "Using an Iterative Linear Solver in an Interior-Point Method for Generating Support Vector Machines." *Computational Optimization and Applications* 47:431–453. http://dx.doi.org/10.1007/s10589-008-9228-z.

Hastie, T. J., Tibshirani, R. J., and Friedman, J. H. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag.

Hurvich, C. M., and Tsai, C.-L. (1989). "Regression and Time Series Model Selection in Small Samples." *Biometrika* 76:297–307.

Koenker, R. (2005). *Quantile Regression*. New York: Cambridge University Press.

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

Koenker, R., and Bassett, G. W. (1982). "Tests of Linear Hypotheses and l_1 Estimation." *Econometrica* 50:1577–1583.

Koenker, R., and Machado, A. F. (1999). "Goodness of Fit and Related Inference Processes for Quantile Regression." *Journal of the American Statistical Association* 94:1296–1310.

Reichler, J. L., ed. (1987). *The 1987 Baseball Encyclopedia Update*. New York: Macmillan.

Schwarz, G. (1978). "Estimating the Dimension of a Model." *Annals of Statistics* 6:461–464.

Time Inc. (1987). "What They Make." *Sports Illustrated* (April 20): 54–81.

Yuan, M. (2006). "GACV for Quantile Smoothing Splines." *Computational Statistics and Data Analysis* 50:813–829. http://econpapers.repec.org/article/eeecsdana/v_3a50_3ay_3a2006_3ai_3a3_3ap_3a813-829.htm.

Subject Index

Syntax Index