



THE
POWER
TO KNOW.

SAS/STAT[®] 14.1 User's Guide

The QUANTSELECT Procedure

This document is an individual chapter from *SAS/STAT® 14.1 User's Guide*.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2015. *SAS/STAT® 14.1 User's Guide*. Cary, NC: SAS Institute Inc.

SAS/STAT® 14.1 User's Guide

Copyright © 2015, SAS Institute Inc., Cary, NC, USA

All Rights Reserved. Produced in the United States of America.

For a hard-copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

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

July 2015

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

Chapter 96

The QUANTSELECT Procedure

Contents

Overview: QUANTSELECT Procedure	7754
Features	7754
Getting Started: QUANTSELECT Procedure	7755
Syntax: QUANTSELECT Procedure	7765
PROC QUANTSELECT Statement	7765
BY Statement	7773
CLASS Statement	7774
CODE Statement	7777
EFFECT Statement	7778
MODEL Statement	7779
OUTPUT Statement	7786
PARTITION Statement	7787
WEIGHT Statement	7788
Details: QUANTSELECT Procedure	7788
Quantile Regression	7788
Quasi-Likelihood Information Criteria	7788
Quasi-Likelihood Ratio Tests	7789
Quantile Process Regression	7791
Observation Quantile Level	7791
Quantile Regression for Extremal Quantile Levels	7792
Effect Selection Methods	7792
Full Model Fitted (NONE)	7793
Forward Selection (FORWARD)	7793
Backward Elimination (BACKWARD)	7793
Stepwise Selection (STEPWISE)	7793
LASSO Method (LASSO)	7794
Criteria Used in Model Selection Methods	7794
Macro Variables That Contain Selected Models	7796
Using Validation and Test Data	7798
Displayed Output	7800
ODS Table Names	7804
ODS Graphics	7804
Example: QUANTSELECT Procedure	7805
Example 96.1: Simulation Study	7805
Example 96.2: Econometric Growth Data	7809
Example 96.3: Pollution and Mortality	7815

Example 96.4: Surface Fitting with Many Noisy Variables	7822
Example 96.5: Quantile Process Regression	7825
References	7832

Overview: QUANTSELECT Procedure

Quantile regression, which was introduced by Koenker and Bassett (1978), is a modern method that models the effects of covariates on the conditional quantiles of a response variable. The QUANTSELECT procedure performs effect selection in the framework of quantile regression. A variety of effect selection methods are available, including greedy methods and penalty methods. The QUANTSELECT procedure offers extensive capabilities for customizing the effect selection processes with a variety of candidate selecting, effect-selection stopping, and final-model choosing criteria. PROC QUANTSELECT also provides graphical summaries for the effect selection processes.

The QUANTSELECT procedure compares most closely to the GLMSELECT and QUANTREG procedures. PROC GLMSELECT performs effect selection in the framework of general linear models. PROC QUANTREG supports a variety of estimation and inference methods for quantile regression but does not directly provide effect selection facilities. The QUANTSELECT procedure, as a counterpart of PROC GLMSELECT for quantile regression, fills this gap.

The QUANTSELECT procedure focuses on linear quantile models for univariate responses and offers great flexibility for and insight into the effect selection algorithm. The QUANTSELECT procedure inherits most of its syntax from PROC GLMSELECT and PROC QUANTREG. The QUANTSELECT procedure provides results that are similar to those of PROC GLMSELECT and PROC QUANTREG. These results (displayed tables, output data sets, and macro variables) make it easy to explore the selected models in PROC QUANTREG.

Features

The main features of the QUANTSELECT procedure are as follows:

- supports the following model specifications:
 - interaction (crossed) effects and nested effects
 - constructed effects such as regression splines
 - hierarchy among effects
 - partitioning of data into training, validation, and testing roles
- provides the following selection controls:
 - multiple methods for effect selection
 - selection for quantile process and single quantile levels
 - selection of individual or grouped effects
 - selection based on a variety of selection criteria
 - stopping rules based on a variety of model evaluation criteria

- produces the following display and output:
 - graphical representation of the selection process
 - output data sets that contain predicted values and residuals
 - an output data set that contains the design matrix
 - macro variables that contain selected effects

The QUANTSELECT procedure supports the following effect selection methods. These methods are explained in detail in the section “[Effect Selection Methods](#)” on page 7792.

- Forward selection starts with no effects or with forced-in effects in the model and adds more effects.
- Backward elimination starts with all effects in the model and deletes effects.
- Stepwise regression is similar to the forward selection method except that effects already in the model do not necessarily stay there.
- LASSO regression adds and deletes effects based on a constrained version of estimated check risk where the L1-norm of regression coefficients is penalized (Tibshirani 1996; Belloni and Chernozhukov 2011). Adaptive LASSO (Zou 2006; Wu and Liu 2009) is implemented as a special case of LASSO methods where the L1-norm of certain weighted regression coefficients is penalized. See the discussion in the section “[LASSO Method \(LASSO\)](#)” on page 7794 for additional details. The QUANTSELECT procedure uses LASSO methods only to determine the adding and dropping covariate effects at a step; a post-penalized model that is associated with the step is refitted without penalty, and the selection criteria and the parameter estimates are from the post-penalized model.

The QUANTSELECT procedure is intended primarily as an effect selection procedure and does not include regression diagnostics and hypothesis testing. The intention is that you use the QUANTSELECT procedure to select a model or a set of models, where each model contains a set of selected effects, and then you can further investigate these models by using PROC QUANTREG or other analytic tools.

Getting Started: QUANTSELECT Procedure

This example demonstrates how you can use the QUANTSELECT procedure to select covariate effects for quantile regression. The Sashelp.Baseball data set contains salary and performance information for Major League Baseball (MLB) players, excluding pitchers, who played 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 from 1986 (Reichler 1987).

The following step displays in [Figure 96.1](#) the variables in the data set:

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

Figure 96.1 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 careers. As a starting point for such an analysis, you can use the following statements to obtain a parsimonious conditional median model at $\tau = 0.5$:

```
proc quantselect data=sashelp.baseball;
  class Div;
  model Salary = nAtBat nHits nHome nRuns nRBI nBB yrMajor crAtBat
                 crHits crHome crRuns crRbi crBB nAssts nError nOuts
                 Div
    / selection=lasso(adaptive stop=aic choose=sbc sh=7);
run;
```

The SELECTION=LASSO(ADAPTIVE) option in the MODEL statement specifies the adaptive LASSO method (Zou 2006), which controls the effect selection process. The STOP=AIC option specifies that Akaike's information criterion (AIC) be used to determine the stopping condition. The CHOOSE=SBC option specifies that the Schwarz Bayesian information criterion (SBC) be used to determine the final selected model. The SH= option specifies the number of stop horizons, which requests that the selection process be stopped whenever the STOP= criterion values at step $s + 1, \dots, s + \text{SH}$ are worse than those for step s for some $s \in \{0, 1, \dots\}$.

Figure 96.2 shows the “Model Information” table, which indicates the effect selection settings. You can see that the default quantile type is single level, so this effect selection is effective only for $\tau = 0.5$.

Figure 96.2 Model Information

The QUANTSELECT Procedure

Model Information	
Data Set	SASHELP.BASEBALL
Dependent Variable	Salary
Selection Method	Adaptive LASSO
Quantile Type	Single Level
Stop Criterion	AIC
Choose Criterion	SBC

Figure 96.3 summarizes the effect selection process, which starts with an intercept-only model at step 0. At step 1, the effect that corresponds to the career runs is added to the model that reduced the AIC value from 2691.6511 to 2510.7297. You can see that step 10 has the minimum AIC and that step 7 has the minimum SBC. Common sense also tells you that the SBC favors a smaller model than the AIC.

Figure 96.3 Selection Summary

The QUANTSELECT Procedure
Quantile Level = 0.5

Selection Summary					
Step	Effect Entered	Effect Removed	Number Effects In	AIC	SBC
0	Intercept		1	2691.6511	2695.2232
1	CrRuns		2	2510.7297	2517.8740
2	nHits		3	2470.4807	2481.1971
3	CrHome		4	2463.5953	2477.8839
4	nBB		5	2463.7806	2481.6414
5	nOuts		6	2455.6212	2477.0541
6	Div AW		7	2451.4609	2476.4660
7	nAtBat		8	2445.0446	2473.6218*
8	CrBB		9	2445.5432	2477.6926
9	nHome		10	2443.4818	2479.2033
10	nRuns		11	2442.6036*	2481.8973
11	Div NE		12	2444.2409	2487.1067
12	CrAtBat		13	2444.5049	2490.9429
13		Div NE	12	2442.8387	2485.7046
14	YrMajor		13	2443.5374	2489.9754
15	nError		14	2445.2085	2495.2187
16	Div NE		15	2446.4042	2499.9865
* Optimal Value Of Criterion					

Figure 96.4 shows that the selection process stopped at a local minimum of the STOP= criterion, which is step 10. According to the SH=7 option, the effect selection process is stopped at step 10 because all the AIC

values for step 11 through step 17 are no less than the AIC at step 10. Step 17 is ignored in the selection summary table because it is the last step.

Figure 96.4 Stop Reason

Selection stopped at a local minimum of the AIC criterion.

Figure 96.5 shows how the final selected model is determined. CHOOSE=SBC is specified in this example, so the model at step 7 is chosen as the final selected model.

Figure 96.5 Selection Reason

The model at step 7 is selected where SBC is 2473.622.

Figure 96.6 shows the final selected effects and Figure 96.7 shows the parameter estimates for the final selected model.

Figure 96.6 Selected Effects

Selected Effects: Intercept nAtBat nHits nBB CrHome CrRuns nOuts Div AW

Figure 96.7 Parameter Estimates

Parameter Estimates			
Parameter	DF	Estimate	Standardized Estimate
Intercept	1	-18.187539	0
nAtBat	1	-1.582714	-0.500417
nHits	1	7.044354	0.686968
nBB	1	2.053726	0.097911
CrHome	1	1.429926	0.272726
CrRuns	1	0.425955	0.316167
nOuts	1	0.282803	0.175489
Div AW	1	-57.671778	-0.056862

Quantile regression can fit a conditional quantile model at any quantile level $\tau \in (0, 1)$, so it can describe the entire distribution of a response variable conditional on covariate effects. To further investigate the effects that might affect the MLB players' salaries, you can also conduct effect selection at $\tau = 0.1$ and $\tau = 0.9$, which correspond to low-end salaries and high-end salaries respectively. The following statements use the same selection settings that are used in the previous program:

```
proc quantselect data=sashelp.baseball;
  class Div;
  model Salary = nAtBat nHits nHome nRuns nRBI nBB yrMajor crAtBat
                 crHits crHome crRuns crRbi crBB nAssts nError nOuts
                 Div
    / quantiles=0.1 0.9 selection=lasso(adaptive stop=aic choose=sbc sh=7);
run;
```


Figure 96.8 shows the effect selection summary with $\tau = 0.1$.

Figure 96.8 Selection Summary: $\tau = 0.1$

The QUANTSELECT Procedure
Quantile Level = 0.1

Selection Summary					
Step	Effect Entered	Effect Removed	Number Effects In	AIC	SBC
0	Intercept		1	2008.3489	2011.9211
1	CrRuns		2	1918.7675	1925.9118
2	nHits		3	1897.2425	1907.9590*
3	YrMajor		4	1897.2476	1911.5362
4	CrBB		5	1896.1765	1914.0373
5	nBB		6	1894.1257	1915.5587
6	CrHome		7	1895.6765	1920.6816
7	nAtBat		8	1890.4051	1918.9824
8	nHome		9	1891.3527	1923.5020
9	Div NE		10	1891.7566	1927.4781
10	nRBI		11	1893.7319	1933.0256
11	CrAtBat		12	1893.9432	1936.8090
12		nRBI	11	1891.9716	1931.2653
13	nAssts		12	1888.6870	1931.5529
14	nRBI		13	1890.5300	1936.9680
15	Div AE		14	1889.4234	1939.4336
16		nRBI	13	1887.6644*	1934.1024
17	CrRbi		14	1888.0966	1938.1068
18	Div AW		15	1890.0322	1943.6145
19	nError		16	1891.7949	1948.9494
20	nRuns		17	1893.2801	1954.0067
21	nRBI		18	1894.7805	1959.0793
22	CrHits		19	1896.6868	1964.5578
* Optimal Value Of Criterion					

Figure 96.9 shows the parameter estimates for the final selected model with $\tau = 0.1$. You can see from Figure 96.9 that low-end salaries for MLB players depend mainly on career runs and hits in 1986.

Figure 96.9 Parameter Estimates: $\tau = 0.1$

The QUANTSELECT Procedure
Quantile Level = 0.1

Parameter Estimates				
Parameter	DF	Estimate	Standardized Estimate	
Intercept	1	-4.397043	0	
nHits	1	0.878564	0.085678	
CrRuns	1	0.327350	0.242977	

Figure 96.10 shows the effect selection summary with $\tau = 0.9$.

Figure 96.10 Selection Summary: $\tau = 0.9$

The QUANTSELECT Procedure
Quantile Level = 0.9

Selection Summary					
Step	Effect Entered	Effect Removed	Number Effects In	AIC	SBC
0	Intercept		1	2436.7289	2440.3011
1	CrHits		2	2197.4349	2204.5792
2	CrRbi		3	2183.6148	2194.3313
3	nHits		4	2113.2757	2127.5643
4		CrRbi	3	2127.8632	2138.5797
5	CrRbi		4	2113.2757	2127.5643
6		CrRbi	3	2127.8632	2138.5797
7	CrRbi		4	2113.2757	2127.5643
8	CrHome		5	2099.2203	2117.0811
9		CrRbi	4	2099.3891	2113.6777
10	CrRbi		5	2099.2203	2117.0811
11		CrRbi	4	2099.3891	2113.6777
12	nOuts		5	2067.1926	2085.0533
13	Div AW		6	2048.2393	2069.6723
14	CrRuns		7	2028.8040	2053.8090
15	nAtBat		8	2012.8195	2041.3968
16		CrHits	7	2017.0290	2042.0341
17	CrRbi		8	2009.3551	2037.9324
18	CrAtBat		9	2011.2415	2043.3908
19		CrRbi	8	2011.4053	2039.9825
20	CrRbi		9	2011.2415	2043.3908
21		CrAtBat	8	2009.3551	2037.9324
22	CrAtBat		9	2011.2415	2043.3908
23	nBB		10	2004.5033	2040.2249
24		CrAtBat	9	2003.1023	2035.2517
25	CrAtBat		10	2004.5033	2040.2249
26		CrAtBat	9	2003.1023	2035.2517*
27	CrAtBat		10	2004.5033	2040.2249
28	nError		11	2004.2230	2043.5167
29	CrHits		12	2003.0544	2045.9203
30	Div NE		13	2001.9603	2048.3983
31	Div AE		14	2001.8349*	2051.8451
32	nRuns		15	2003.5961	2057.1784
33	nHome		16	2004.2721	2061.4266
34	nRBI		17	2006.0023	2066.7289
35	YrMajor		18	2007.9975	2072.2963
36	CrBB		19	2009.9514	2077.8223
37	nAssts		20	2011.9095	2083.3525
* Optimal Value Of Criterion					

Figure 96.11 shows the parameter estimates for the final selected model with $\tau = 0.9$.

Figure 96.11 Parameter Estimates: $\tau = 0.9$

Parameter Estimates			
Parameter	DF	Estimate	Standardized Estimate
Intercept	1	92.893875	0
nAtBat	1	-1.858170	-0.587509
nHits	1	8.155573	0.795335
nBB	1	3.392794	0.161751
CrHome	1	3.191472	0.608700
CrRuns	1	1.394317	1.034939
CrRbi	1	-0.913371	-0.664951
nOuts	1	0.437241	0.271323
Div AW	1	-167.110005	-0.164764

To visually illustrate how the model evolves through the selection process, the QUANTSELECT procedure provides the coefficient plot, the average check loss plot, and several criterion plots in either packed or unpacked forms. You can request these plots by using the **PLOTS=** option. The following statements request all the plots for the baseball data at $\tau = 0.1$; they also use the **STOP=AIC** criterion, the **CHOOSE=SBC** criterion, and the **SH=7** option:

```
ods graphics on;
proc quantselect data=sashelp.baseball plots=all;
  class Div;
  model Salary = nAtBat nHits nHome nRuns nRBI nBB yrMajor crAtBat
                 crHits crHome crRuns crRbi crBB nAssts nError nOuts
                 Div
    / quantiles=0.1 selection=lasso(adaptive stop=aic choose=sbc sh=7);
run;
```

Figure 96.12 shows the progression of the parameter estimates as the selection process proceeds.

Figure 96.12 Coefficient Panel: $\tau = 0.1$

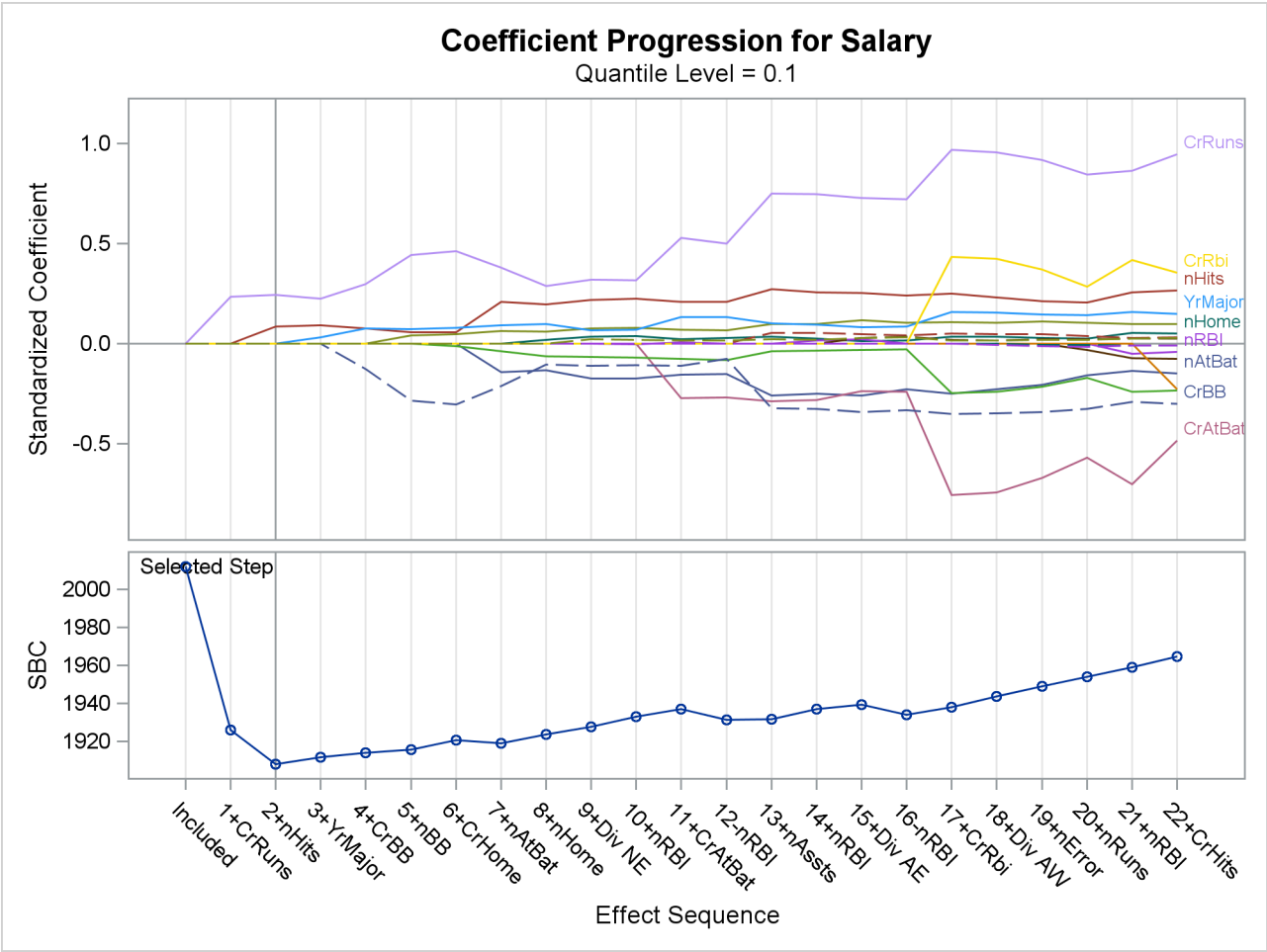


Figure 96.13 shows the progression of the average check losses as the selection process proceeds.

Figure 96.13 Average Check Loss Plot: $\tau = 0.1$

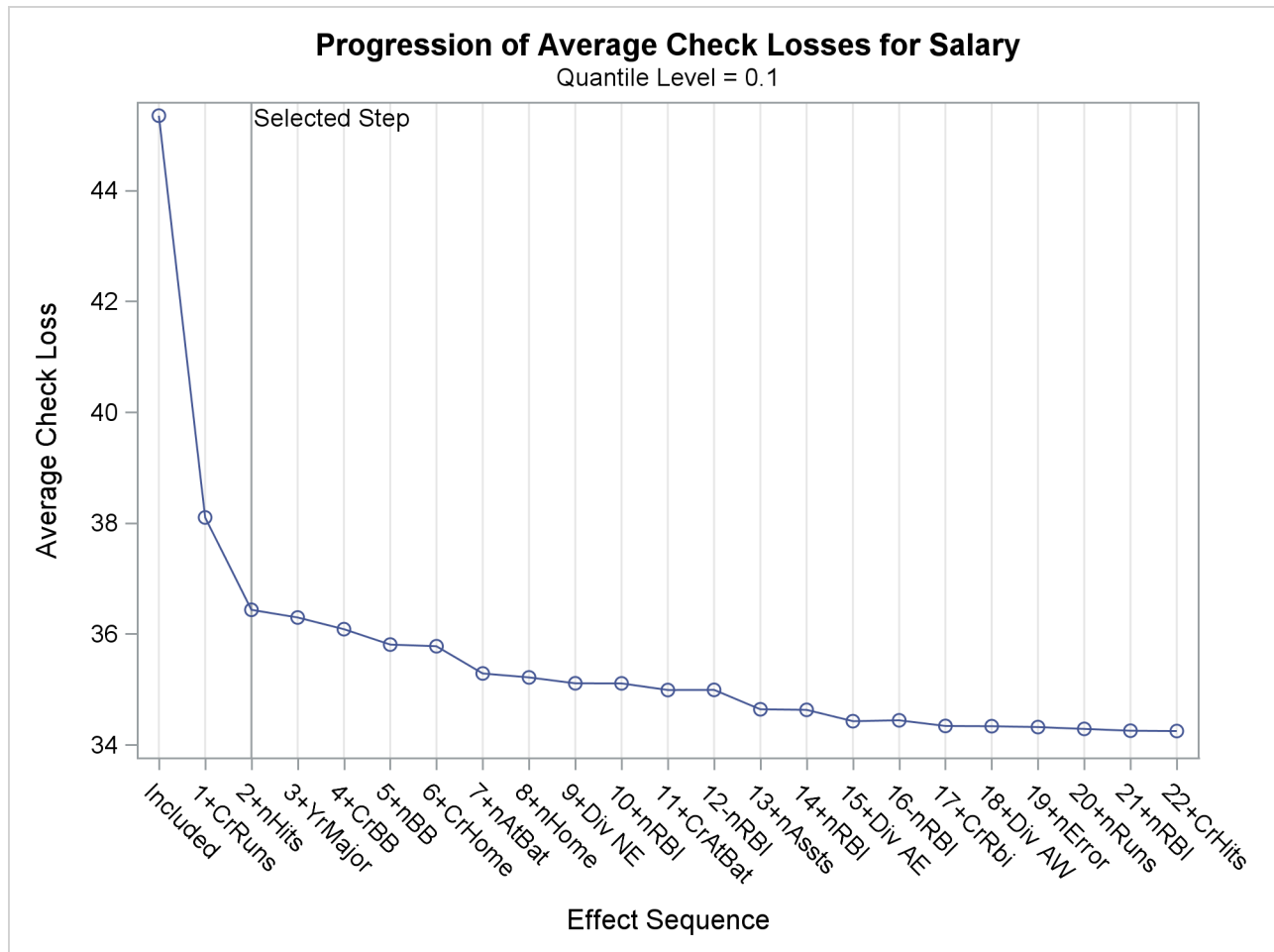
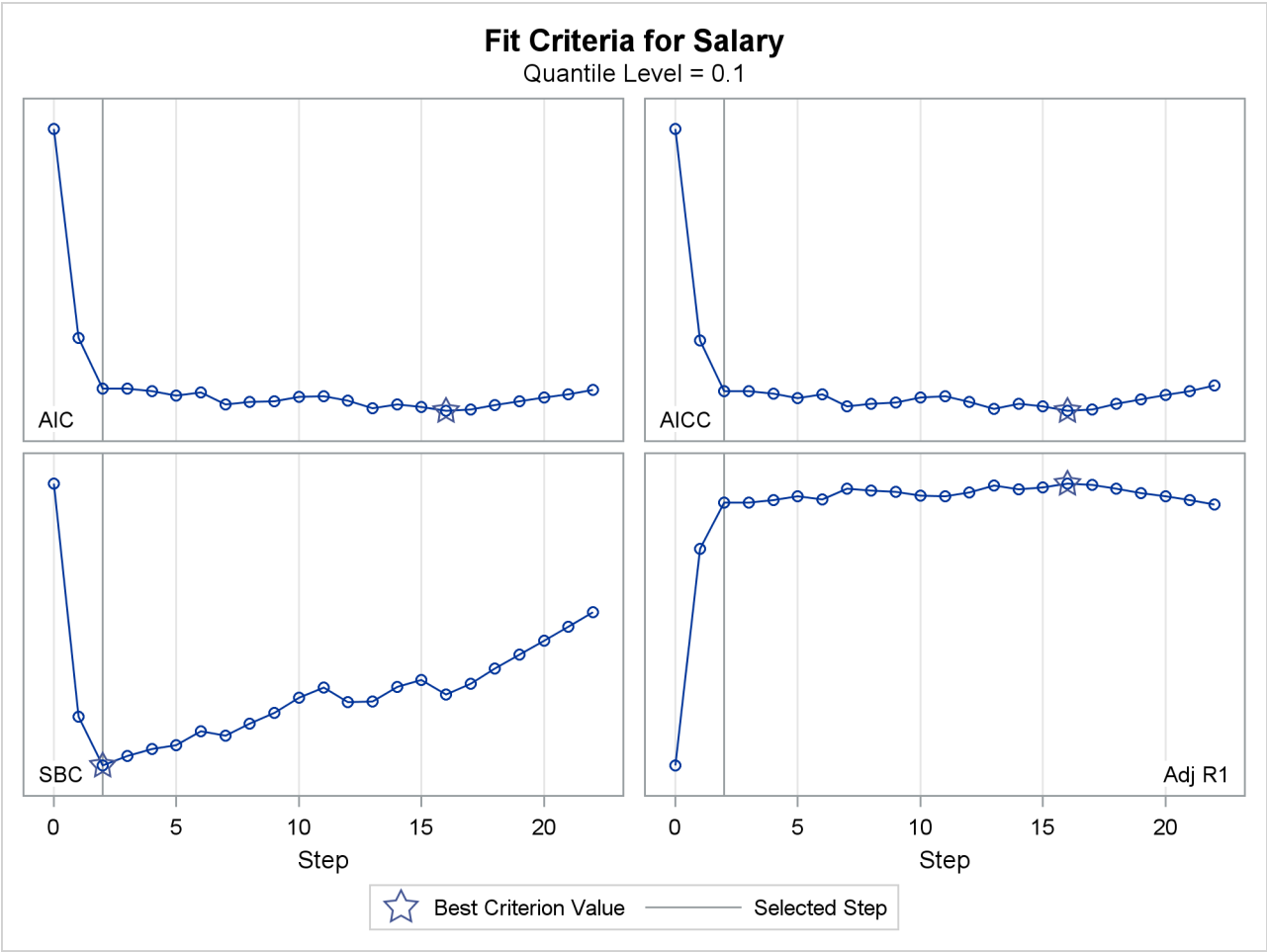


Figure 96.14 shows the progression of four effect selection criteria as the selection process proceeds.

Figure 96.14 Criterion Panel: $\tau = 0.1$



Syntax: QUANTSELECT Procedure

The following statements are available in PROC QUANTSELECT:

```

PROC QUANTSELECT < options > ;
  BY variables ;
  CLASS variable < (v-options) > < variable < (v-options ... ) > > < / v-options > < options > ;
  CODE < options > ;
  EFFECT name = effect-type (variables < / options > ) ;
  MODEL variable = < effects > < / options > ;
  OUTPUT < OUT=SAS-data-set > < keyword < =name > > < ... keyword < =name > > > ;
  PARTITION < options > ;
  WEIGHT variable ;

```

The PROC QUANTSELECT statement invokes the procedure. All statements other than the **MODEL** statement are optional. **CLASS** and **EFFECT** statements, if present, must precede the **MODEL** statement.

PROC QUANTSELECT Statement

```
PROC QUANTSELECT < options > ;
```

Table 96.1 lists the *options* available in the PROC QUANTSELECT statement.

Table 96.1 PROC QUANTSELECT Statement Options

<i>option</i>	Description
Data Set Options	
DATA=	Names a data set to use for the regression
MAXMACRO=	Sets the maximum number of macro variables to produce
TESTDATA=	Names a data set that contains test data
VALDATA=	Names a data set that contains validation data
ODS Graphics Options	
PLOTS=	Produces ODS Graphics displays
Other Options	
ALGORITHM=	Specifies an algorithm for estimating the regression parameters
NAMELEN=	Specifies the maximum length of effect names in tables and output data sets
NOPRINT	Suppresses displayed output (including plots)
OUTDESIGN=	Names a data set that contains the design matrix
PARMLABELSTYLE=	Sets the style of parameter names and labels for nested and crossed effects
SEED=	Sets the seed used for pseudorandom number generation

You can specify the following *options* (shown in alphabetical order) in the PROC QUANTSELECT statement.

ALGORITHM=SIMPLEX | SMOOTH

specifies either the simplex algorithm (ALGORITHM=SIMPLEX) or the smoothing algorithm (ALGORITHM=SMOOTH) for estimating the regression parameters. The smoothing algorithm is computationally much more efficient than the simplex algorithm for fitting models on large data sets. You might consider specifying the ALGORITHM=SMOOTH if your DATA= data set contains more than 5,000 observations and more than 50 regressors. The smoothing algorithm does not support quantile process effect selection or the LASSO selection method. By default, ALGORITHM=SIMPLEX.

DATA=SAS-data-set

names the SAS data set to be used by PROC QUANTSELECT. If the DATA= option is not specified, PROC QUANTSELECT uses the most recently created SAS data set. If the data set contains a variable named `_ROLE_`, then this variable is used to assign observations for training, validation, and testing roles. See the section “[Using Validation and Test Data](#)” on page 7798 for more information about using the `_ROLE_` variable.

MAXMACRO=*n*

specifies the maximum number of macro variables with selected effects to create. By default, MAXMACRO=100.

PROC QUANTSELECT saves the list of selected effects in a macro variable, `&_QRSIND`. For example, suppose your input effect list consists of `x1–x10`. Then `&_QRSIND` would be set to `x1 x3 x4 x10` if the first, third, fourth, and tenth effects were selected for the model. This list can be used in the MODEL statement of a subsequent procedure.

If you specify the OUTDESIGN= option in the PROC QUANTSELECT statement, then PROC QUANTSELECT saves the list of columns in the design matrix in a macro variable named `&_QRSMOD`.

With multiple quantile levels and BY processing, one macro variable is created for each combination of quantile level and BY group, and the macro variables are indexed by the BY-group number and the quantile-level index. You can use the MAXMACRO= option to either limit or increase the number of these macro variables when you are processing data sets with many combinations of quantile level and BY group.

With a single quantile level and no BY-group processing, PROC QUANTSELECT creates the macro variables shown in [Table 96.2](#).

Table 96.2 Macro Variables Created for a Single Quantile Level and No BY Processing

Macro Variable Name	Contains
<code>_QRSIND</code>	Selected effects
<code>_QRSIND1</code>	Selected effects
<code>_QRSINDT1</code>	Selected effects
<code>_QRSIND1T1</code>	Selected effects
<code>_QRSMOD</code>	Selected design matrix columns
<code>_QRSMOD1</code>	Selected design matrix columns
<code>_QRSMODT1</code>	Selected design matrix columns
<code>_QRSMOD1T1</code>	Selected design matrix columns

With multiple quantile levels and BY-group processing, PROC QUANTSELECT creates the macro variables shown in [Table 96.3](#).

Table 96.3 Macro Variables Created for a Multiple Quantile Levels and BY-Group Processing

Macro Variable Name	Contains
_QRSIND	Selected effects for quantile 1 and BY group 1
_QRSINDT1	Selected effects for quantile 1 and BY group 1
_QRSINDT2	Selected effects for quantile 2 and BY group 1
.	
.	
.	
_QRSIND1	Selected effects for quantile 1 and BY group 1
_QRSIND1T1	Selected effects for quantile 1 and BY group 1
_QRSIND1T2	Selected effects for quantile 2 and BY group 1
.	
.	
.	
_QRSIND2	Selected effects for quantile 1 and BY group 2
_QRSIND2T1	Selected effects for quantile 1 and BY group 2
_QRSIND2T2	Selected effects for quantile 2 and BY group 2
.	
.	
.	
_QRSIND mTn	Selected effects for quantile n and BY group m

If you specify the OUTDESIGN= option, PROC QUANTSELECT also creates the macro variables shown in [Table 96.4](#).

Table 96.4 Macro Variables Created When the OUTDESIGN= Option Is Specified

Macro Variable Name	Contains
_QRSMOD	Selected design matrix columns for BY group 1
_QRSMOD1	Selected design matrix columns for BY group 1
_QRSMOD2	Selected design matrix columns for BY group 2
.	
.	
.	
_QRSMOD mTn	Selected design matrix columns for quantile n and BY group m

The macros variables in Table 96.5 show the number of quantiles and BY groups:

Table 96.5 Macro Variables Showing the Number of Quantiles and BY Groups

Macro Variable Name	Contains
_QRSNUMBYS	The number of BY groups
_QRSNUMTAUS	The number of quantiles
_QRSBY1NUMTAUS	The number of _QRSIND1Tj macro variables actually made
_QRSBY2NUMTAUS	The number of _QRSIND2Tj macro variables actually made
.	
.	
.	
_QRSNUMBYTAUS	The number of _QRSINDiTj macro variables actually made. This value can be less than $_QRSNUMBYS \times _QRSNUMTAUS$, and it is less than or equal to $MAXMACRO=n$.

See the section “[Macro Variables That Contain Selected Models](#)” on page 7796 for more information.

NAMELEN=number

specifies the maximum length of effect names. By default, NAMELEN=20. If you specify a value less than 20, the default is used.

NOPRINT

suppresses all displayed output (including plots).

OUTDESIGN<(options)><=SAS-data-set>

creates a data set that contains the design matrix. By default, the QUANTSELECT procedure includes in the OUTDESIGN data set the X matrix that corresponds to all the effects in the selected models. Two schemes for naming the columns of the design matrix are available:

- In the first scheme, names of the parameters are constructed from the parameter labels that appear in the parameter estimates table. This naming scheme is the default when you do not request BY processing, or when you specify the FULLMODEL option with BY processing.
- In the second scheme, the design matrix column names consist of a prefix followed by an index. The default name prefix is _X. This scheme is used when you specify the PREFIX= option, or when you specify a BY statement without using the FULLMODEL option; otherwise the first scheme is used.

You can specify the following *options* in parentheses to control the contents of the OUTDESIGN= data set:

ADDINPUTVARS

includes all the input data set variables in the OUTDESIGN= data set.

ADDVALDATA

includes the VALDATA= data set observations in the OUTDESIGN= data set. This option is ignored if the VALDATA= data set is not specified.

ADDTESTDATA

includes the TESTDATA= data set observations in the OUTDESIGN= data set. This option is ignored if TESTDATA= data set is not specified.

FULLMODEL

includes in the OUTDESIGN= data set parameters that correspond to all effects that are specified in the MODEL statement. By default, only parameters that correspond to the selected model are included.

NAMES

produces a table that associates columns in the OUTDESIGN= data set with the labels of the parameters they represent.

PREFIX<=prefix>

creates the design matrix column names from a prefix followed by an index. The default *prefix* is X.

PARMLABELSTYLE=options

specifies how parameter names and labels are constructed for nested and crossed effects.

The following *options* are available:

INTERLACED <(SEPARATOR=*quoted string*)>

forms parameter names and labels by positioning levels of classification variables and constructed effects adjacent to the associated variable or constructed effect name and using “ * ” as the delimiter for both crossed and nested effects. This style of naming parameters and labels is used in the TRANSREG procedure. You can request truncation of the classification variable names used in forming the parameter names and labels by using the CPREFIX= and LPREFIX= options in the CLASS statement. You can use the SEPARATOR= suboption to change the delimiter between the crossed variables in the effect. PARMLABELSTYLE=INTERLACED is not supported if you specify the SPLIT option in an EFFECT statement or a CLASS statement. The following are examples of the parameter labels in this style (Age is a continuous variable, Gender and City are classification variables):

```
Age
Gender male * City Beijing
City London * Age
```

SEPARATE

specifies that in forming parameter names and labels, the effect name appears before the levels associated with the classification variables and constructed effects in the effect. You can control the length of the effect name by using the NAMELEN= option in the PROC GLMSELECT statement. In forming parameter labels, the first level that is displayed is positioned so that it starts at the same offset in every parameter label—this enables you to easily distinguish the effect name from the levels when the parameter labels are displayed in a column in the “Parameter Estimates” table. The following are examples of the parameter labels in this style (Age is a continuous variable, Gender and City are classification variables):

```

Age
Gender*City male Beijing
Age*City      London

```

SEPARATECOMPACT

requests the same parameter naming and labeling scheme as PARMLABELSTYLE=SEPARATE except that the first level in the parameter label is separated from the effect name by a single blank. This style of labeling is used in the PLS procedure and is the default if you do not specify the PARMLABELSTYLE option. The following are examples of the parameter labels in this style (Age is a continuous variable, Gender and City are classification variables):

```

Age
Gender*City male Beijing
Age*City London

```

PLOTS | PLOT *<(global-plot-options)> <=plot-request <(options)>>*

PLOTS | PLOT *<(global-plot-options)> <=(plot-request <(options)> <... plot-request <(options)>>)>*

controls the plots that are produced through ODS Graphics. When you specify only one *plot-request*, you can omit the parentheses around it. Here are some examples:

```

plots=all
plots=coefficients(unpack)
plots(unpack)=(coef acl crit)

```

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

```

ods graphics on;
proc quantselect plots=all;
  class temp sex / split;
  model depVar = sex sex*temp;
run;

```

For more information about enabling and disabling ODS Graphics, see the section “[Enabling and Disabling ODS Graphics](#)” on page 609 in Chapter 21, “[Statistical Graphics Using ODS](#).”

You can specify the following *global-plot-options*, which apply to all plots generated by the QUANTSELECT procedure, unless they are altered by specific plot *options*.

ENDSTEP=*n*

specifies that the step ranges shown on the horizontal axes of plots terminate at the specified step. By default, the step range shown terminates at the final step of the selection process. If you specify the ENDSTEP= option as both a *global-plot-option* and as an *option* for a specific *plot-request*, then PROC QUANTSELECT uses the ENDSTEP=*n* option for the specific *plot-request*.

LOGP | LOGPVALUE

displays the natural logarithm of the entry and removal significance levels when the [SELECT=SL](#) option is specified in the MODEL statement.

MAXSTEPLABEL=*n*

specifies the maximum number of characters beyond which labels of effects on plots are truncated. The default is MAXSTEPLABEL=256.

MAXPARMLABEL=*n*

specifies the maximum number of characters beyond which parameter labels on plots are truncated. The default is MAXPARMLABEL=256.

STARTSTEP=*n*

specifies that the step ranges shown on the horizontal axes of plots start at the specified step. By default, the step range shown starts at the initial step of the selection process. If you specify the STATSTEP= option as both a *global-plot-option* and as an *option* for a specific *plot-request*, then PROC QUANTSELECT uses the STARTSTEP=*n* option for the specific *plot-request*. The default is STARTSTEP=0.

STEPAXIS=EFFECT | NORMB | NUMBER

specifies the method for labeling the horizontal plot axis. This axis represents the sequence of entering or departing effects. The default is STEPAXIS=EFFECT.

STEPAXIS=EFFECT

labels each step by a prefix followed by the name of the effect that enters or leaves at that step. The prefix consists of the step number followed by a “+” sign or a “–” sign, depending on whether the effect enters or leaves at that step.

STEPAXIS=NORMB

labels the horizontal axis value at step *i* with the penalty on the parameter estimates at step *i*, normalized by the penalty on the parameter estimates at the final step. This option is valid only with regularization selection methods.

STEPAXIS=NUMBER

labels each step with the step number.

UNPACK

displays each graph separately. (By default, some graphs can appear together in a single panel.) You can also specify UNPACK as a suboption with CRITERIA and COEFFICIENTS options for specific *plot-requests*.

The following list describes the specific *plot-requests* and their *options*.

ALL

displays all appropriate graphs.

ACL | ACLPLOT <(aclplot-option)>

plots the progression of the average check losses on the training data, and on the test and validation data when these data are provided with the TESTDATA= or VALDATA= options or are produced by using a PARTITION statement. When the PROC QUANTSELECT procedure is applied on multiple quantile levels, the ACL option and its suboptions apply to the ACL plots for each of the quantile levels.

You can specify the following *aclplot-option*:

STEPAXIS=EFFECT | NORMB | NUMBER

specifies the method for labeling the horizontal plot axis. See the [STEPAXIS=](#) option in the *global-plot-options* for more information.

COEF | COEFFICIENTS | COEFFICIENTPANEL <(coefficient-panel-options)>

displays a panel of two plots for each quantile level. The upper plot shows the progression of the parameter values as the selection process proceeds. The lower plot shows the progression of the [CHOOSE=](#) criterion. If no [CHOOSE=](#) criterion is in effect, then the AICC criterion is displayed. You can specify the following *coefficient-panel-options*:

LABELGAP=percentage

specifies the percentage of the vertical axis range that forms the minimum gap between successive parameter labels at the final step of the coefficient progression plot. If the values of more than one parameter at the final step are closer than this gap, then the labels on all but one of these parameters are suppressed. The default is LABELGAP=5.

LOGP | LOGPVALUE

displays the natural logarithm of the entry and removal significance levels when the [SELECT=SL](#) option is specified in the MODEL statement.

STEPAXIS=EFFECT | NORMB | NUMBER

specifies the horizontal axis to be used. See the [STEPAXIS=](#) option in the *global-options* for more information.

UNPACK | UNPACKPANEL

displays the coefficient progression and the [CHOOSE=](#) criterion progression in separate plots.

CRIT | CRITERIA | CRITERIONPANEL <(criterion-panel-options)>

plots a panel of model fit criteria. If multiple quantile levels apply, the CRITERIA option plots a panel of model fit criteria for each quantile level. The criteria that are displayed are AIC, AICC, and SBC, in addition to any other criteria that are named in the [CHOOSE=](#), [SELECT=](#), [STOP=](#), and [STATS=](#) options in the MODEL statement. You can specify the following *criterion-panel-options*:

STEPAXIS=EFFECT | NORMB | NUMBER

specifies the horizontal axis to be used. See the [STEPAXIS=](#) option in the *global-options* for more information.

UNPACK | UNPACKPANEL

displays each criterion progression on a separate plot.

NONE

suppresses all plots.

SEED=number

specifies an integer that is 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 by reading the time of day from the computer's clock.

TESTDATA=SAS-data-set

names a SAS data set that contains test data. This data set must contain all the effects that are specified in the **MODEL** statement. Furthermore, when you also specify a **BY** statement and the **TESTDATA=** data set contains any of the **BY** variables, then the **TESTDATA=** data set must also contain all the **BY** variables sorted in the order of the **BY** variables. In this case, only the test data for a specific **BY** group are used with the corresponding **BY** group in the analysis data. If the **TESTDATA=** data set contains none of the **BY** variables, then the entire **TESTDATA=** data set is used with each **BY** group of the analysis data.

If you specify both a **TESTDATA=** data set and the **PARTITION** statement, then the testing observations from the **DATA=** data set are merged with the **TESTDATA=** data set for testing purposes.

VALDATA=SAS-data-set

names a SAS data set that contains validation data. This data set must contain all the effects that are specified in the **MODEL** statement. Furthermore, when a **BY** statement is used and the **VALDATA=** data set contains any of the **BY** variables, then the **VALDATA=** data set must also contain all the **BY** variables sorted in the order of the **BY** variables. In this case, only the validation data for a specific **BY** group are used with the corresponding **BY** group in the analysis data. If the **VALDATA=** data set contains none of the **BY** variables, then the entire **VALDATA=** data set is used with each **BY** group of the analysis data.

If you specify both a **VALDATA=** data set and the **PARTITION** statement, then the validation observations from the **DATA=** data set are merged with the **VALDATA=** data set for validation purposes.

BY Statement

BY variables ;

You can specify a **BY** statement with **PROC QUANTSELECT** 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 for the **QUANTSELECT** 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).

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* < (*v-options*) > < *variable* < (*v-options* ...) > > < / *v-options* > < *options* > ;

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

Table 96.6 summarizes the *options* and *v-options* available in the CLASS statement.

Table 96.6 CLASS Statement Options

<i>option or v-option</i>	Description
DELIMITER=	Specifies the delimiter
DESCENDING	Reverses the sort order
MISSING	Allows for missing values
ORDER=	Specifies the sort order
PARAM=	Specifies the parameterization method
REF=	Specifies the reference level
SHOW	Requests a table for each CLASS variable
SPLIT	Splits CLASS variables into independent effects

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

DELIMITER='c'

specifies the delimiter character, 'c', to be used between levels of classification variables when parameter names and lists of class level values are built. The default delimiter is a space. This option is useful if the levels of a classification variable contain embedded blanks.

SHOW | SHOWCODING

requests a table that shows the coding used for each classification variable.

You can specify various *v-options* for each variable by enclosing them in parentheses after the variable name; these are called individual *v-options*. You can also specify global *v-options* by placing them after a slash (/) at the end of the CLASS statement. Global *v-options* are applied to all the variables specified in the CLASS statement. If you specify more than one CLASS statement, the global *v-options* specified in any one CLASS statement apply to all CLASS statements. However, individual CLASS variable *v-options* override the global *v-options* except for the **PARAM=GLM** option. The global **PARAM=GLM** option overrides all individual **PARAM=** options.

You can specify the following *v-options*:

CPREFIX=n

specifies that, at most, the first *n* characters of a CLASS variable name be used in creating names for the corresponding design variables. The default is $32 - \min(32, \max(2, f))$, where *f* is the formatted length of the CLASS variable. This option applies only when you specify the **PARMLABELSTYLE=INTERLACED** option in the PROC QUANTSELECT statement.

DESCENDING**DESC**

reverses the sort order of the classification variable.

LPREFIX=*n*

specifies that, at most, the first *n* characters of a CLASS variable label be used in creating labels for the corresponding design variables. The default is $256 - \min(256, \max(2, f))$, where *f* is the formatted length of the CLASS variable. This option applies only when you specify the PARMLABELSTYLE=INTERLACED option in the PROC QUANTSELECT statement.

MISSING

allows missing values, such as ‘.’ for a numeric variable or a blank for a character variable, as valid values for the CLASS variable.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sort order for the levels of classification variables. If ORDER=FORMATTED for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values. The following table shows how PROC QUANTSELECT interprets values of the ORDER= option.

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

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

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

PARAM=keyword

specifies the parameterization method for the classification variable or variables. Design matrix columns are created from CLASS variables according to the following coding schemes. If the PARAM= option is not specified with any individual CLASS variable, by default, PARAM=GLM. Otherwise, the default is PARAM=EFFECT. If PARAM=ORTHOPOLY or PARAM=POLY, and the CLASS levels are numeric, then the ORDER= option in the CLASS statement is ignored, and the internal, unformatted values are used. See the section “[CLASS Variable Parameterization and the SPLIT Option](#)” on page 3859 in Chapter 49, “[The GLMSELECT Procedure](#),” for more information.

EFFECT

specifies effect coding.

GLM

specifies less-than-full-rank coding. This option can be used only as a global *v-option* (after the slash in the CLASS statement).

You can specify the SPLIT option for individual classification variables. For example, consider the following PROC QUANTSELECT statements:

```
proc quantselect;
  class temp(split) sex;
  model depVar = sex sex*temp;
run;
```

In this case, the effect 'sex' is not split, and the effect 'sex*temp' is split into three effects labeled 'sex*temp_hot', 'sex*temp_warm', and 'sex*temp_cold'. Furthermore each of these three split effects now has two parameters that correspond to the two levels of 'sex, ' and the previous PROC QUANTSELECT statements are equivalent to the following statements for the split version of the DATA= data set:

```
proc quantselect;
  class sex;
  model depVar = sex sex*temp_hot sex*temp_warm sex*temp_cold;
run;
```

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 96.7 summarizes the *options* available in the CODE statement.

Table 96.7 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 399 in Chapter 19, “Shared Concepts and Topics.”

EFFECT Statement

EFFECT *name=effect-type (variables < / options>)* ;

The EFFECT statement enables you to construct special collections of columns for design matrices. These collections are referred to as *constructed effects* to distinguish them from the usual model effects that are formed from continuous or classification variables, as discussed in the section “GLM Parameterization of Classification Variables and Effects” on page 391 in Chapter 19, “Shared Concepts and Topics.”

You can specify the following *effect-types*:

COLLECTION	specifies a collection effect that defines one or more variables as a single effect with multiple degrees of freedom. The variables in a collection are considered as a unit for estimation and inference.
LAG	specifies a classification effect in which the level that is used for a particular period corresponds to the level in the preceding period.
MULTIMEMBER MM	specifies a multimember classification effect whose levels are determined by one or more variables that appear in a CLASS statement.
POLYNOMIAL POLY	specifies a multivariate polynomial effect in the specified numeric variables.
SPLINE	specifies a regression spline effect whose columns are univariate spline expansions of one or more variables. A spline expansion replaces the original variable with an expanded or larger set of new variables.

Table 96.8 summarizes the *options* available in the EFFECT statement.

Table 96.8 EFFECT Statement Options

Option	Description
Collection Effects Options	
DETAILS	Displays the constituents of the collection effect
Lag Effects Options	
DESIGNROLE=	Names a variable that controls to which lag design an observation is assigned
DETAILS	Displays the lag design of the lag effect
NLAG=	Specifies the number of periods in the lag
PERIOD=	Names the variable that defines the period. This option is required.
WITHIN=	Names the variable or variables that define the group within which each period is defined. This option is required.
Multimember Effects Options	
NOEFFECT	Specifies that observations with all missing levels for the multimember variables should have zero values in the corresponding design matrix columns

Table 96.8 *continued*

Option	Description
WEIGHT=	Specifies the weight variable for the contributions of each of the classification effects
Polynomial Effects Options	
DEGREE=	Specifies the degree of the polynomial
MDEGREE=	Specifies the maximum degree of any variable in a term of the polynomial
STANDARDIZE=	Specifies centering and scaling suboptions for the variables that define the polynomial
Spline Effects Options	
BASIS=	Specifies the type of basis (B-spline basis or truncated power function basis) for the spline effect
DEGREE=	Specifies the degree of the spline effect
KNOTMETHOD=	Specifies how to construct the knots for the spline effect

For more information about the syntax of these *effect-types* and how columns of constructed effects are computed, see the section “[EFFECT Statement](#)” on page 401 in Chapter 19, “[Shared Concepts and Topics](#).”

MODEL Statement

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

The MODEL statement names the dependent variable and the covariate effects, including covariates, main effects, constructed effects, interactions, and nested effects; see the section “[Specification of Effects](#)” on page 3593 in Chapter 46, “[The GLM Procedure](#),” for more information. If you omit the explanatory effects, PROC QUANTSELECT fits an intercept-only model.

After the keyword MODEL, specify the dependent (response) variable, followed by an equal sign, followed by the explanatory effects.

Table 96.9 summarizes the *options* available in the MODEL statement.

Table 96.9 MODEL Statement Options

Option	Description
DETAILS=	Specifies the level of effect selection detail to display
HIERARCHY=	Specifies hierarchy of effects to impose
NOINT	Specifies models without an explicit intercept
QUANTILES=	Specifies quantile levels to be applied
SELECTION=	Specifies effect selection method
STATS=	Specifies additional statistics to be displayed
TEST=	Specifies the test type for computing significance levels

The following list provides details about the *options* that you can specify in the MODEL statement after a slash (/):

DETAILS=*level* | **STEPS** <(step options)>

specifies the level of effect selection detail that is displayed, where *level* can be ALL, STEPS, or SUMMARY. The default if the DETAILS= option is omitted is DETAILS=SUMMARY that produces only the selection summary table. The DETAILS=ALL option produces the following:

- entry and removal statistics for each variable that is selected in the model building process
- fit statistics and parameter estimates
- entry and removal statistics for the top five candidates for inclusion or exclusion at each step
- a selection summary table

The option DETAILS=STEPS <(step options)> provides the step information and the selection summary table. The following suboptions can be specified within parentheses after the DETAILS=STEPS option:

FITSTATISTICS | **FITSTATS** | **FIT**

requests fit statistics at each selection step.

PARAMETERESTIMATES | **PARMEST**

requests parameter estimates at each selection step.

CANDIDATES <(ALL | *n*)>

requests entry or removal statistics for the best *n* candidate effects for inclusion or exclusion at each step. If you specify the CANDIDATES(ALL) option, then all candidates are shown. If the CANDIDATES(*n*) is not specified, then the best 10 candidates are shown. The entry or removal statistic is the statistic named in the **SELECT=** option that is specified in the MODEL statement **SELECTION=** option.

HIERARCHY=*keyword*

HIER=*keyword*

specifies whether and how the model hierarchy requirement is applied. This option also controls whether a single effect or multiple effects are allowed to enter or leave the model in one step. You can specify that only CLASS effects, or both CLASS and continuous effects, be subject to the hierarchy requirement. This option is ignored unless you also specify one of the following options: **SELECTION=FORWARD**, **SELECTION=BACKWARD**, or **SELECTION=STEPWISE**.

Model hierarchy refers to the requirement that for any term to be in the model, all model effects contained in the term must be present in the model. For example, in order for the interaction A*B to enter the model, the main effects A and B must be in the model. Likewise, neither effect A nor effect B can leave the model while the interaction A*B is in the model.

You can specify the following *keywords*:

NONE

specifies that model hierarchy not be maintained. Any single effect can enter or leave the model at any given step of the selection process.

SINGLE

specifies that only one effect enter or leave the model at one time, subject to the model hierarchy requirement. For example, suppose that the model contains the main effects A and B and the interaction A*B. In the first step of the selection process, either A or B can enter the model. In the second step, the other main effect can enter the model. The interaction effect can enter the model only when both main effects have already entered. Also, before A or B can be removed from the model, the A*B interaction must first be removed. All effects (CLASS and interval) are subject to the hierarchy requirement.

SINGLECLASS

is the same as HIERARCHY=SINGLE except that only CLASS effects are subject to the hierarchy requirement.

The default is HIERARCHY=NONE.

NOINT

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

QUANTILES=*number-list* | **PROCESS** <(option)>

QUANTILE=<*number-list* | **PROCESS** <(option)>>

specifies the quantile levels for the quantile regression. You can specify any number of quantile levels in (0, 1). You can also specify QUANTILE=0 or QUANTILE=1 for the ALGORITHM=SIMPLEX algorithm. See the section “[Quantile Regression for Extremal Quantile Levels](#)” on page 7792 for more information about extremal quantile levels.

If you do not specify this option, the QUANTSELECT procedure performs median regression effect selection that corresponds to QUANTILES=0.5.

If you specify the QUANTILES=PROCESS option, the QUANTSELECT procedure performs effect selection for quantile process regression. See the section “[Quantile Process Regression](#)” on page 7791 for more information about quantile process regression. The QUANTILES=PROCESS option cannot be used with LASSO selection methods. You can specify the following *option* in parentheses after QUANTILES=PROCESS.

NTAU=*n* | **ALL**

specifies how many quantile levels that you expect to cover for the quantile process. If you specify NTAU=ALL, the QUANTSELECT procedure performs effect selection for accurate quantile process regression. If you specify NTAU=*n*, the QUANTSELECT procedure performs effect selection for approximate quantile process regression. The approximate quantile process is computed at *n* equally spaced quantile levels: $\{\frac{1}{n+1}, \dots, \frac{n}{n+1}\}$ besides three control quantile levels {0, 0.5, 1}. If the number of observations for training is more than 1000, by default, NTAU=500. Otherwise, the default is NTAU=ALL.

SELECTION=*method* <(method-options)>

specifies the *method* used to select the model, optionally followed by parentheses that enclose *method-options* that apply to the specified method. The default is SELECTION=STEPWISE.

You can specify the following *methods*, which are explained in detail in the section “[Effect Selection Methods](#)” on page 7792.

NONE	specifies full model fitting without effect selection.
FORWARD	specifies forward selection. This method starts with no effects in the model and adds effects.
BACKWARD	specifies backward elimination. This method starts with all effects in the model and deletes effects.
STEPWISE	specifies stepwise regression. This is similar to the FORWARD method except that effects already in the model do not necessarily stay there.
LASSO	specifies a method that adds and deletes parameters based on a version of estimated check risk where the weighted L1-norm of certain weighted regression coefficients is penalized. For more information, see the section “ LASSO Method (LASSO) ” on page 7794. If the model contains CLASS variables or constructed effects, these CLASS variables or constructed effects are split into separate covariates.

Table 96.10 lists the applicable *method-options* for each *method*.

Table 96.10 Applicable *method-options* for Each *method*

<i>method-option</i>	FORWARD	BACKWARD	STEPWISE	LASSO
ADAPTIVE				X
CHOOSE=	X	X	X	X
INCLUDE=	X	X	X	X
MAXSTEP=	X	X	X	X
SELECT=	X	X	X	
SLENTY=	X		X	
SLSTAY=		X	X	
STOP=	X	X	X	X
STOPHORIZON=	X	X	X	X

You can specify the following *method-option* in parentheses after the *method*. As described in Table 96.10, not all *method-options* apply to every SELECTION= method.

ADAPTIVE

ADAPT

specifies the adaptive LASSO selection method. The ADAPTIVE option can be used only with the SELECTION=LASSO option.

CHOOSE=*criterion*

chooses from the list of models (with one model at each step of the selection process) the model that yields the best value of the specified *criterion* as the final selected model. If the optimal value of the specified *criterion* occurs for more than one model, then the model with the smallest number of parameters is chosen. If you do not specify the CHOOSE= option, then the model selected is the model at the final step in the selection process for the SELECT=SL criterion, or the STOP= option is applied as the CHOOSE= option for all the other cases.

You can specify the following values for *criterion* in the CHOOSE= option. See the section “[Criteria Used in Model Selection Methods](#)” on page 7794 for more information about these criteria.

ADJR1	chooses the model with the largest adjusted quantile regression R statistic.
AIC	chooses the model with the smallest Akaike's information criterion.
AICC	chooses the model with the smallest corrected Akaike's information criterion.
SBC	chooses the model with the smallest Schwarz Bayesian information criterion.
VALIDATE	chooses the model with the smallest average check loss for the validation data. You can specify <code>CHOOSE=VALIDATE</code> only if you have specified a <code>VALIDATA=</code> data set in the PROC QUANTSELECT statement or if you have reserved part of the input data for validation by using either a <code>PARTITION</code> statement or a <code>_ROLE_</code> variable in the input data.

INCLUDE=*n*

forces the first *n* effects listed in the MODEL statement to be included in all models. The selection methods are performed on the other effects in the MODEL statement.

MAXSTEP=*n*

specifies the maximum number of selection steps. The default value of *n* is the number of effects in the MODEL statement when `SELECTION=FORWARD` or `SELECTION=BACKWARD` and is three times the number of effects when `SELECTION=STEPWISE` or `SELECTION=LASSO`.

SELECT=*criterion*

specifies the criterion that PROC QUANTSELECT uses to determine the order in which effects enter or leave at each step of the specified selection method. This option is not valid when `SELECTION=LASSO`. You can specify the following values for *criterion*: ADJR1, AIC, AICC, SBC, SL, and VALIDATE. See the section “[Criteria Used in Model Selection Methods](#)” on page 7794 for more information about these criteria.

When `SELECT=SL`, the effect selection depends on the selection method and is described in the relevant subsection of the section “[Effect Selection Methods](#)” on page 7792. Otherwise, the effect that is selected to enter or leave at a step of the selection process is the effect whose addition to or removal from the current model produces the maximum improvement in the specified criterion.

If validation data exist, the default is `SELECT=VALIDATE`; otherwise, the default is `SELECT=SBC`.

SLENTY=*value***SLE=*value***

specifies the significance level for entry, used when the `SELECT=SL` option is in effect. The defaults are 0.50 when `SELECTION=FORWARD` and 0.15 when `SELECTION=STEPWISE`.

SLSTAY=*value***SLS=*value***

specifies the significance level for staying in the model, used when the `SELECT=SL` option is in effect. The defaults are 0.10 when `SELECTION=BACKWARD` and 0.15 when `SELECTION=STEPWISE`.

STOP=*criterion*

specifies the *criterion* for stopping the selection process. If the maximum number of steps is specified in the `MAXSTEP=` option and the *criterion* does not stop the selection process before the maximum number of steps for the selection method, then the selection process terminates at the maximum number of steps.

You can specify the following values for *criterion*. See the section “Criteria Used in Model Selection Methods” on page 7794 for more detailed descriptions of these criteria.

NONE	enables the model selection process to go through all possible steps.
ADJR1	stops selection at the step where the next SH= steps (or all remaining steps) would yield models with smaller values of the adjusted quantile regression R (ADJR1) statistic.
AIC	stops selection at the step where the next SH= steps (or all remaining steps) would yield models with larger values of Akaike’s information criterion.
AICC	stops selection at the step where the next SH= steps (or all remaining steps) would yield models with larger values of the corrected Akaike’s information criterion.
SBC	stops selection at the step where the next SH= steps (or all remaining steps) would yield models with larger values of the Schwarz Bayesian information criterion.
VALIDATE	stops selection at the step where the next SH= steps (or all remaining steps) would yield models with larger values of the average check loss for the validation data. You can specify STOP=VALIDATE only if you have specified a VALIDATA= data set in the PROC QUANTSELECT statement or if you have reserved part of the input data for validation by using either a PARTITION statement or a _ROLE_ variable in the input data.

The default *criterion* depends on other factors as follows:

- If validation data exist, STOP=VALIDATE by default.
- If validation data do not exist and you specify SELECTION=LASSO, STOP=SBC by default. The SELECTION=LASSO option does not support the SELECT=*method-option*.
- If validation data do not exist and you specify SELECTION= STEPWISE, FORWARD, or BACKWARD, the default is one of the following:
 - When you specify SELECT=SL, the entry and stay significance levels terminate the effect selection process.
 - When you do not specify SELECT=SL, the default is the criterion that is specified in the SELECT= option.

If you specify both the STOP= option and SELECT=SL, the following rules apply:

- When you specify SELECTION=STEPWISE, the entry and stay significance levels can terminate the effect selection process when no candidate effect is available to be deleted from or added to the model. This extra check can result in the selection terminating before a local minimum of the STOP= criterion is found.
- When you specify SELECTION=FORWARD, the effect selection process ignores the entry significance level even if you use the SLE= option to specify the entry significance level.
- When you specify SELECTION=BACKWARD, the effect selection process ignores the stay significance level even if you use the SLS= option to specify the stay significance level.

STOPHORIZON=*n***SH=*n***

looks ahead for the specified number of steps to decide whether an extremum of the stop criterion is achieved. This option applies only to the STOP= criterion. The default is STOPHORIZON=1.

For example, suppose that the stop criterion values at steps 1 through 5 are 4, 3, 5, 6, and 2, respectively. If you specify STOPHORIZON=1, then the selection process terminates after looking at the model at step 3, and the final selected model is the model at step 2. If you specify STOPHORIZON=2, the selection process stops after looking at the model at step 4, and the final selected model is the model at step 2. However, if you specify STOPHORIZON=3 or higher, then the local minimum in the stop value sequence at step 2 cannot stop the selection process because a lower value is achieved at step 5, which is within 3 steps beyond this local minimum step.

STAT=*name* | (*names*)**STATS=*name* | (*names*)**

specifies which model fit statistics to display in the selection summary table. To specify multiple model fit statistics, specify a list of *names* in parentheses. If you omit this option, the default set of statistics that are displayed in these tables includes all the criteria that are specified in any of the **CHOOSE=**, **SELECT=**, and **STOP= method-options**.

You can specify the following values for *name*:

ADJR1	displays the adjusted quantile regression R statistic.
AIC	displays the Akaike's information criterion.
AICC	displays the corrected Akaike's information criterion.
ACL	displays the average check losses for the training, test, and validation data. The ACL statistics for the test and validation data are reported only if you have specified the TESTDATA= option or the VALDATA= option in the PROC QUANTSELECT statement or if you have reserved part of the input data for testing or validation by using either a PARTITION statement or a _ROLE_ variable in the input data.
R1	displays the quantile regression R statistic.
SBC	displays the Schwarz Bayesian information criterion.

The statistics ADJR1, AIC, AICC, and SBC can be computed with little computation cost. However, computing ACL for test and validation data when these are not used in any of the **CHOOSE=**, **SELECT=**, and **STOP= method-options** can hurt performance.

TEST=*name*

specifies the test type for computing significance levels.

You can specify the following values for *name*:

LR1 specifies the likelihood ratio test Type I. The LR1 test score is

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

where $D_1(\tau) = \sum \rho_\tau(y_i - \mathbf{x}_i \hat{\beta}_1(\tau))$ is the sum of check losses for the reduced model, $D_2(\tau) = \sum \rho_\tau(y_i - \mathbf{x}_i \hat{\beta}(\tau))$ is the sum of check losses for the extended

model, and \hat{s} is the estimated sparsity function. See the section “[Quasi-Likelihood Ratio Tests](#)” on page 7789 for more information.

LR2

specifies the likelihood ratio test Type II. The LR2 test score is

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

See the section “[Quasi-Likelihood Ratio Tests](#)” on page 7789 for more information.

OUTPUT Statement

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

The OUTPUT statement creates a new SAS data set that saves diagnostic measures that are calculated for the selected model. If you do not specify a *keyword*, then the only diagnostic included is the predicted response.

All the variables in the original data set are included in the new data set, along with variables that are created by the *keyword* options in the OUTPUT statement. These new variables contain the values of a variety of statistics and diagnostic measures that are calculated for each observation in the data set.

The OUTPUT data set is created in row-wise form, and the variable `_QUANTILE_` is optional. For each appropriate *keyword* specified in the OUTPUT statement, one variable for each specified quantile level is generated. These variables appear in the sorted order of the specified quantile levels.

If you specify a BY statement, then a variable `_BY_` that indexes the BY groups is included. For each observation, the value of `_BY_` is the index of the BY group to which this observation belongs. This variable is useful for matching BY groups with macro variables that PROC QUANTSELECT creates. See the section “[Macro Variables That Contain Selected Models](#)” on page 7796 for more information.

If you have partitioned the input data with a [PARTITION](#) statement, then a character variable `_ROLE_` is included in the output data set. The following table shows the value of `_ROLE_` for each observation:

<code>_ROLE_</code> Value	Observation Role
TEST	Testing
TRAIN	Training
VALIDATE	Validation

If you want to create a permanent SAS data set, you must specify a two-level name. For more information about permanent SAS data sets, see the discussion in *SAS Language Reference: Concepts*.

You can specify the following arguments in the OUTPUT statement:

keyword <=name>

specifies the statistics to include in the output data set and optionally names the new variables that contain the statistics. Specify one of the following *keywords* for each desired statistic, followed optionally by an equal sign, and the *name* of 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 the new variable name is formed by using a prefix of one or more characters that identify the statistic, followed by an underscore (_), followed by the dependent variable name.

PREDICTED | PRED | P includes predicted values in the output data set. The prefix for the default name is p.

QUANTLEVEL | QL includes observation quantile levels in the output data set. The prefix for the default name is ql. The QL= option is available only when you specify QUANTILE=PROCESS in the **MODEL** statement. For more information about observation quantile level, see the section “[Observation Quantile Level](#)” on page 7791.

RESIDUAL | RESID | R includes residuals, calculated as ACTUAL – PREDICTED, in the output data set. The prefix for the default name is r.

OUT=SAS-data-set

names the output data set. By default, PROC QUANTSELECT uses the DATA*n* convention to name the new data set.

PARTITION Statement

PARTITION < *option* > ;

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.

An alternative to using a PARTITION statement is to provide a variable named `_ROLE_` in the input data set to define roles of observations in the input data. If you specify a PARTITION statement, then any `_ROLE_` variable in the input data set is ignored. If you do not use a PARTITION statement and the input data do not contain a variable named `_ROLE_`, then all observations in the input data set are assigned to model training.

You can specify either (but not both) of the following *options*:

ROLEVAR=*variable* (< **TEST=***value*> < **TRAIN=***value*> < **VALIDATE=***value*>)

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. The TEST=, TRAIN=, and VALIDATE= suboptions specify the formatted values of this variable that are used to assign observations roles. If you do not specify the TRAIN= suboption, then all observations whose role is not determined by the TEST= or VALIDATE= suboptions are assigned to training.

FRACTION(< **TEST=***fraction*> < **VALIDATE=***fraction*>)

requests that specified proportions of the observations in the input data set be randomly assigned training and validation roles. You specify the proportions for testing and validation by using the TEST= and VALIDATE= suboptions. If you specify both the TEST= and the 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.

WEIGHT Statement

WEIGHT *variable* ;

A WEIGHT statement names a variable in the input data set with values that are relative weights for a weighted quantile regression fit.

Values of the weight variable must be nonnegative. If an observation's weight is 0, the observation is deleted from the analysis. If a weight is negative or missing, it is set to 0, and the observation is excluded from the analysis.

Details: QUANTSELECT Procedure

Quantile Regression

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

Let $\{(y_i, \mathbf{x}_i) : i = 1, \dots, n\}$ denote a data set of observations, where y_i are responses, and \mathbf{x}_i are regressors. Koenker and Bassett (1978) defined the *regression quantile* at quantile level $\tau \in (0, 1)$ as any solution that minimizes the following objective function in $\boldsymbol{\beta}$:

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

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 $w_i, i = 1, \dots, n$, in the WEIGHT statement, weighted quantile regression is carried out by solving

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

Quasi-Likelihood Information Criteria

Given 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 $i = 1, \dots, n$ 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 associated -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

$$\hat{\boldsymbol{\beta}}(\tau) = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{i=1}^n \rho_{\tau}(y_i - \mathbf{x}_i' \boldsymbol{\beta})$$

The MLE for σ is

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

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

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

$$\text{AIC}(\tau) = 2n \ln(\text{ACL}(\tau)) + 2p$$

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

Similarly, the quasi-likelihood corrected AIC and Schwarz Bayesian information criterion can be formulated respectively as follows:

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

$$\text{SBC}(\tau) = 2n \ln(\text{ACL}(\tau)) + p \ln(n)$$

In fact, the quasi-likelihood AIC, AICC, and SBC are fairly robust, and they can be used to select effects for data sets without the iid assumption in asymmetric Laplace distribution. See “[Example 96.1: Simulation Study](#)” on page 7805 for a simulation study that applies SBC for effect selection on a data set that is generated from a naive instrumental model (Chernozhukov and Hansen 2008).

Quasi-Likelihood Ratio Tests

Under the iid assumption, Koenker and Machado (1999) proposed 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 \hat{s} is the estimated sparsity function, $D_1(\tau) = \sum \rho_{\tau}(y_i - \mathbf{x}_i' \hat{\boldsymbol{\beta}}_1(\tau))$ is the sum of check losses for the reduced model, and $D_2(\tau) = \sum \rho_{\tau}(y_i - \mathbf{x}_i' \hat{\boldsymbol{\beta}}(\tau))$ is the sum of check losses for the extended model. The Type II test score, LR2, is defined as

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

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

If you specify the TEST=LR1 option in the **MODEL** statement, the QUANTSELECT procedure uses LR1 score to compute the significance level. Or you can use the substitutable TEST=LR2 option for computing the significance level on Type II quasi-likelihood ratio test.

Under the iid assumption, the sparsity function is defined as $s(\tau) = 1/f(F^{-1}(\tau))$. Here the distribution of errors F is flexible but not limited to the asymmetric Laplace distribution. The algorithm for estimating $s(\tau)$ 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 . Then \hat{s} is computed on the reduced model for testing the entry effect and on the extended model for testing the removal effect.
2. Compute quantile-level bandwidth h_n . The QUANTSELECT procedure computes the Bofinger bandwidth, which is an optimizer of mean squared error for standard density estimation:

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

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

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

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

$$\hat{F}^{-1}(t) = \begin{cases} r_{(1)} & \text{if } t \in [0, 1/2n) \\ \lambda r_{(i+1)} + (1 - \lambda) r_{(i)} & \text{if } t \in [(i - 0.5)/n, (i + 0.5)/n) \\ r_{(n)} & \text{if } t \in [(2n - 1), 1] \end{cases}$$

where $r_{(i)}$ is the i th smallest residual and $\lambda = t - (i - 0.5)/n$.

- 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

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

Because a real data set might not follow the null hypothesis and the iid assumptions, the LR1 and LR2 scores that are used for quantile regression effect selection often do not follow a χ^2 distribution. Hence, the SLENTY and SLSTAY values cannot reliably be viewed as probabilities. One way to address this difficulty is to treat the SLENTY and SLSTAY values only as criteria for comparing importance levels of effect candidates at each selection step, and not to explain these values as probabilities.

Quantile Process Regression

You can specify `QUANTILE=PROCESS` in the `MODEL` statement to perform quantile process regression. Quantile process regression fits quantile regression models for the entire range of quantile levels from 0 to 1. Because a quantile function is the inverse of its cumulative distribution function, quantile process regression can estimate the entire distribution of a response variable conditional on its covariates.

Because of the piecewise linearity of the check loss function, the optimal quantile regression solution $\hat{\beta}(\tau)$ is a step function in $\tau \in [0, 1]$. In other words, given any optimal solution $\hat{\beta}(\tau^*)$, there exists an optimal quantile-level range $[\tau_1, \tau_2]$ such that $\hat{\beta}(\tau) = \hat{\beta}(\tau^*)$ is optimal for any $\tau \in [\tau_1, \tau_2]$. This step-function property can simplify integration computation in quantile process regression. For example, to estimate conditional mean by using quantile process regression, you can substitute integration by using the summation

$$E(Y|\mathbf{X} = \mathbf{x}) = \int_0^1 \mathbf{x} \hat{\beta}(\tau) d\tau = \mathbf{x} \sum_{i=1}^s (\tau_{i+1} - \tau_i) \hat{\beta}_i$$

where $\tau_1 = 0$, $\tau_{s+1} = 1$, and $\hat{\beta}_i$ is the optimal solution for quantile range $[\tau_i, \tau_{i+1}]$.

If you specify the `NTAU=ALL` suboption in the `QUANTILE=PROCESS` option, `PROC QUANTSELECT` outputs $\hat{\beta} = \sum_{i=1}^s (\tau_{i+1} - \tau_i) \hat{\beta}_i$ as the mean parameter estimates in the parameter estimates table. If you request the “Parameter Estimates for Quantile Process” table, `PROC QUANTSELECT` outputs parameter estimates in the following quantile-level grid:

$$\left\{ 0, \frac{\tau_1 + \tau_2}{2}, \frac{\tau_2 + \tau_3}{2}, \dots, \frac{\tau_s + \tau_{s+1}}{2}, 1 \right\}$$

For more information about the “Parameter Estimates for Quantile Process” table, see “[Parameter Estimates for Quantile Process](#)” on page 7803. `PROC QUANTSELECT` also uses this grid to estimate observation quantile levels.

If you specify `NTAU=n`, `PROC QUANTSELECT` approximates the quantile process regression in the following quantile-level grid:

$$\left\{ 0, \frac{1}{n+1}, \frac{2}{n+1}, \dots, 0.5, \dots, \frac{n}{n+1}, 1 \right\}$$

When `NTAU=n`, `PROC QUANTSELECT` also approximates integrations by using the linear interpolation method, which defines

$$\hat{\beta}(\tau) = \frac{\tau - \tau_1}{\tau_2 - \tau_1} \hat{\beta}(\tau_2) + \frac{\tau_2 - \tau}{\tau_2 - \tau_1} \hat{\beta}(\tau_1)$$

Here, τ_1 and τ_2 denote two consecutive quantile levels in the quantile-level grid that satisfy $\tau \in [\tau_1, \tau_2]$.

Observation Quantile Level

The observation quantile level of a valid observation, (y, \mathbf{x}) , is defined as $\tau_{(y, \mathbf{x})} = F_{Y|\mathbf{x}}(y)$, where $F_{Y|\mathbf{x}}(\cdot)$ denotes the cumulative distribution function (CDF) for the y 's underlying distribution conditional on \mathbf{x} . For the CDF that is continuous at y , the equation $y = Q_{Y|\mathbf{x}}(\tau_{(y, \mathbf{x})})$ holds because the quantile function is inversely related to the CDF. Ideally, if $y = \mathbf{x} \hat{\beta}(\tau^*)$ for a unique $\tau^* \in [0, 1]$ and some quantile-regression optimal solution $\hat{\beta}(\tau^*)$, then τ^* is a reasonable estimation for $\tau_{(y, \mathbf{x})}$, written as $\hat{\tau}_{(y, \mathbf{x})} = \tau^*$. However, such a τ^* might not exist or is nonunique in practice. The following steps show how the `QUANTSELECT` procedure estimates the observation quantile level $\tau_{(y, \mathbf{x})}$ via quantile process regression:

1. Fit the quantile process regression model and label its quantile-level grid as follows:

$$\{0 = \tau_{(0)} \leq \tau_{(1)} \leq \cdots \leq \tau_{(s)} \leq \tau_{(s+1)} = 1\}$$

2. Compute quantile predictions conditional on \mathbf{x} in the quantile-level grid: $\{q_i = \mathbf{x}\hat{\boldsymbol{\beta}}_i : i = 0, \dots, s+1\}$.
3. Sort q_i 's to avoid crossing, such that $q_{(0)} \leq q_{(1)} \leq \cdots \leq q_{(s+1)}$.
4. $\hat{\tau}_{(y,\mathbf{x})} = 0$ if $y < q_{(0)}$, or $\hat{\tau}_{(y,\mathbf{x})} = 1$ if $y > q_{(s+1)}$.
5. Otherwise, search index j such that $q_{(j)} < y < q_{(j+1)}$. If such a j exists,

$$\hat{\tau}_{(y,\mathbf{x})} = \left(\frac{y - q_{(j)}}{q_{(j+1)} - q_{(j)}} \right) \tau_{(j+1)} + \left(\frac{q_{(j+1)} - y}{q_{(j+1)} - q_{(j)}} \right) \tau_{(j)}$$

6. Otherwise, search j and k such that $q_{(j-1)} < y = q_{(j)} = \cdots = q_{(j+k)} < q_{(j+k+1)}$, and set $\hat{\tau}_{(y,\mathbf{x})} = \frac{\tau_{(j)} + \tau_{(j+k)}}{2}$. Here, define $q_{(-1)} = -\infty$ and $q_{(s+2)} = \infty$.

Quantile Regression for Extremal Quantile Levels

A quantile level τ is extremal if τ is equal to or approaching 0 or 1. The solution for an extremal quantile-level quantile regression problem can be nonunique because the parameter estimate of the intercept effect can be arbitrarily small or large. In a quantile process regression toward the direction of the specified extremal quantile level, the tightest solution refers to the first solution whose quantile-level range includes the specified extremal quantile level. Among all the valid solutions for an extremal quantile-level quantile regression problem, the tightest solution can generalize the terminology of sample minimum and sample maximum.

The QUANTSELECT procedure computes the tightest solution for an extremal quantile-level quantile regression problem by using the ALGORITHM=SIMPLEX algorithm. If $\tau \in \left[\frac{1}{4n}, 1 - \frac{1}{4n} \right]$, τ is not extremal. Otherwise, follow these steps:

1. Set $\tau_0 = \frac{1}{4n}$ (or $\tau_0 = \left(1 - \frac{1}{4n}\right)$).
2. Compute $\hat{\boldsymbol{\beta}}(\tau_0) = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{i=1}^n \rho_{\tau_0}(y_i - \mathbf{x}_i' \boldsymbol{\beta})$.
3. Find the quantile-level lower limit (or upper limit), τ_1 , such that $\hat{\boldsymbol{\beta}}(\tau_0)$ is still optimal at τ_1 .
4. If $\tau_1 \leq \tau$ (or $\tau_1 \geq \tau$), return $\hat{\boldsymbol{\beta}}(\tau_0)$. Otherwise, update $\tau_0 = \tau_1 - c$ (or $\tau_0 = \tau_1 + c$) for a small tolerance $c > 0$, and go to step 2.

Effect Selection Methods

The effect selection methods implemented in PROC QUANTSELECT are specified with the SELECTION= option in the MODEL statement.

Full Model Fitted (NONE)

The complete model specified in the **MODEL** statement is used to fit the model, and no effect selection is done. You request this method by specifying **SELECTION=NONE** in the **MODEL** statement.

Forward Selection (FORWARD)

The forward selection technique begins with just the forced-in covariates and then sequentially adds the effect that most improves the fit. The process terminates when no significant improvement can be obtained by adding any effect. You request this method by specifying **SELECTION=FORWARD** in the **MODEL** statement.

If you specify the **SELECT=SL method-option**, you can use the **TEST= method-option** to specify a test statistic for gauging improvement in fit. For example, if **TEST=LR1**, at each step the effect that yields the most significant likelihood ratio statistic is added and the process continues until all effects that are not in the model have LR1 statistics that are not significant at the entry significance level (which is specified in the **SLE= option**). Because effects can contribute different degrees of freedom to the model, it is necessary to compare the *p*-values that correspond to these statistics.

Backward Elimination (BACKWARD)

The backward elimination technique starts from the full model, which includes all independent effects. Then effects are deleted one by one until a stopping condition is satisfied. At each step, the effect that shows the smallest contribution to the model is deleted. You request this method by specifying **SELECTION=BACKWARD** in the **MODEL** statement.

Suppose you specify the **SELECT=SL method-option** and the **TEST=LR1 method-option** to gauge improvement in quantile regression fit. At any step, the predictor that produces the least significant LR1 statistic is dropped and the process continues until all effects that remain in the model have LR1 statistics that are significant at the stay significance level (which is specified in the **SLS= option**).

Stepwise Selection (STEPWISE)

The stepwise method is a modification of the forward selection technique in which effects already in the model do not necessarily stay there. You request this method by specifying **SELECTION=STEPWISE** in the **MODEL** statement.

In the implementation of the stepwise selection method, the same entry and removal approaches for the forward selection and backward elimination methods are used to assess contributions of effects as they are added to or removed from a model. Suppose you specify **SELECT=SL**. If, at a step of the stepwise method, any effect in the model is not significant at the level specified by the **SLSTAY= method-option**, then the least significant of these effects is removed from the model and the algorithm proceeds to the next step. This ensures that no effect can be added to a model while some effect currently in the model is not deemed significant. Only after all necessary deletions have been accomplished can another effect be added to the model. In this case, the effect whose addition yields the most significant statistic value is added to the model and the algorithm proceeds to the next step. The stepwise process ends when none of the effects outside the model is significant at the level specified by the **SLENTY= method-option** and every effect in the model is significant at the level specified by the **SLSTAY= method-option**. In some cases, neither of these two conditions for stopping is met and the sequence of models cycles. In this case, the stepwise method terminates at the end of the second cycle.

Just as with forward selection and backward elimination, you can use the **SELECT=** *method-option* to change the criterion used to assess effect contributions. You can also use the **STOP=** *method-option* to specify a stopping criterion and use the **CHOOSE=** *method-option* to specify a criterion used to select among the sequence of models produced.

LASSO Method (LASSO)

The standard LASSO method uses a standardized design matrix that orthogonalizes selectable covariates against forced-in covariates, and then scales the orthogonalized selectable covariates so that they all have the same sum of squares. See the information about the standard parameter estimate in the section “[Parameter Estimates](#)” on page 7802 for more information about design matrix orthogonalization. The LASSO method initializes all the selectable coefficients into 0 at step 0. The predictor that reduces the average check loss the fastest relative to the L1-norm of the selectable coefficient increment is determined, and a step is taken in the direction of this predictor.

The difference between adaptive LASSO and standard LASSO methods is in the prescaling of the selectable coefficients. After orthogonalization against forced-in covariates, the adaptive LASSO method first fits a full model without penalty, and then scales the orthogonalized selectable covariates with the corresponding coefficients from the full model. This adaptive scaling can be equivalently substituted by using a weighted L1-norm penalty, where the weights are the reciprocals of the corresponding coefficients from the full model.

The length of this step determines the coefficient of this predictor and is chosen when some residual changes its sign or some predictor that is not used in the model can reduce the average check loss more efficiently. This process continues until all predictors are in the model.

As with other selection methods, the issue of when to stop the selection process is crucial. You can use the **CHOOSE=** *method-option* to specify a criterion for choosing among the models at each step. You can also use the **STOP=** *method-option* to specify a stopping criterion. See the section “[Criteria Used in Model Selection Methods](#)” on page 7794 for more information and [Table 96.11](#) for the formulas for evaluating these criteria.

Criteria Used in Model Selection Methods

PROC QUANTSELECT supports a variety of fit statistics that you can specify as criteria for the **CHOOSE=**, **SELECT=**, and **STOP=** *method-options* in the **MODEL** statement.

Single Quantile Effect Selection

The following fit statistics are available for single quantile effect selection:

AIC	applies the Akaike’s information criterion (Akaike 1981; Darlington 1968; Judge et al. 1985).
AICC	applies the corrected Akaike’s information criterion (Hurvich and Tsai 1989).

SBC	applies the Schwarz Bayesian information criterion (Schwarz 1978; Judge et al. 1985).
SL<(LR1 LR2)>	specifies the significance level of a statistic used to assess an effect's contribution to the fit when it is added to or removed from a model. LR1 specifies likelihood ratio Type I, and LR2 specifies the likelihood ratio Type II. By default, the LR1 statistic is applied.
ADJR1	applies the adjusted quantile regression R statistic.
VALIDATE	applies the average check loss for the validation data.

Table 96.11 provides formulas and definitions for these fit statistics.

Table 96.11 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 i th observation; $r_i(\tau) = y_i - \mathbf{x}_i\boldsymbol{\beta}(\tau)$
$D(\tau)$	Total sum of check losses; $D(\tau) = \sum_{i=1}^n \rho_{\tau}(r_i)$
$D_0(\tau)$	Total sum of check losses for intercept-only model if intercept is a forced-in effect, otherwise for empty-model.
$ACL(\tau)$	Average check loss; $ACL(\tau) = \frac{D(\tau)}{n}$
$R1(\tau)$	Counterpart of linear regression R-square for quantile regression; $1 - \frac{D(\tau)}{D_0(\tau)}$
$ADJR1(\tau)$	Adjusted R1; $ADJR1(\tau) = 1 - \frac{(n-1)D(\tau)}{(n-p)D_0(\tau)}$
$AIC(\tau)$	Akaike's information criterion; $AIC(\tau) = 2n \ln(ACL(\tau)) + 2p$
$AICC(\tau)$	Corrected Akaike's information criterion; $AICC(\tau) = 2n \ln(ACL(\tau)) + \frac{2pn}{n-p-1}$
$SBC(\tau)$	Schwarz Bayesian information criterion; $SBC(\tau) = 2n \ln(ACL(\tau)) + p \ln(n)$

Quantile Process Effect Selection

The following statistics are available for quantile process effect selection:

AIC	specifies Akaike's information criterion (Akaike 1981; Darlington 1968; Judge et al. 1985).
AICC	specifies the corrected Akaike's information criterion (Hurvich and Tsai 1989).
SBC	specifies Schwarz Bayesian information criterion (Schwarz 1978; Judge et al. 1985).
ADJR1	specifies the adjusted quantile regression R statistic.
VALIDATE	specifies average check loss for the validation data.

Table 96.12 provides formulas and definitions for the fit statistics.

Table 96.12 Formulas and Definitions for Model Fit Summary
Statistics for Quantile Process Effect Selection

Statistic	Definition or Formula
D	Integral of total sum of check losses; $D = \int_0^1 D(\tau) d\tau$
D_0	Integral of total sum of check losses for intercept-only model or empty-model if the NOINT option is used; $D_0 = \int_0^1 D_0(\tau) d\tau$
ACL	Integral of average check loss; $ACL = \frac{D}{n}$
R1	Counterpart of linear regression R-square for quantile process regression; $R1 = 1 - \frac{D}{D_0}$
ADJR1	Adjusted R1; $ADJR1 = 1 - \frac{(n-1)D}{(n-p)D_0}$
AIC	Akaike's information criterion; $AIC = \int_0^1 AIC(\tau) d\tau$
AICC	Corrected Akaike's information criterion; $AICC = \int_0^1 AICC(\tau) d\tau$
SBC	Schwarz Bayesian information criterion; $SBC = \int_0^1 SBC(\tau) d\tau$

Macro Variables That Contain Selected Models

PROC QUANTSELECT saves the list of selected effects in a macro variable so that you can use other SAS procedures to perform post-selection 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.

Table 96.13 describes the macro variables that PROC QUANTSELECT creates. When multiple quantile levels or BY processing are used, one macro variable, indexed by the quantile-level order and the BY group number, is created for each quantile level and BY group combination.

Table 96.13 Macro Variables Created for Subsequent Processing

Macro Variable	Description
Single Quantile Level and No BY processing	
_QRSIND	Selected model
Multiple Quantile Levels and No BY Processing	
_QRSNUMTAUS	Number of quantile levels
_QRSINDT1	Selected model for the first quantile level
_QRSINDT2	Selected model for the second quantile level
...	
Single Quantile Level and BY Processing	
_QRSNUMBYS	Number of BY groups
_QRSIND1	Selected model for BY group 1
_QRSIND2	Selected model for BY group 2
...	
Multiple Quantile Levels and BY Processing	
_QRSNUMTAUS	Number of quantile levels
_QRSNUMBYS	Number of BY groups
_QRSIND1T1	Selected model for the first quantile level and BY group 1
_QRSIND1T2	Selected model for the second quantile level and BY group 1
...	
_QRSIND2T1	Selected model for the first quantile level and BY group 2
_QRSIND2T2	Selected model for the second quantile level and BY group 2
...	

The macro variables _QRSIND, _QRSINDT1, _QRSIND1, and _QRSIND1T1 are all synonyms. If you do not specify multiple quantile levels or BY processing, the macro variables _QRSNUMTAUS and _QRSNUMBYS are both set to 1.

PROC QUANTSELECT creates two output data set variables, _BY_ and _QUANTILE_, to aid in associating macro variables with output data set observations when multiple quantile levels or BY processing are used. The values of these two variables are integers that match the i,j components of the macro variable names _QRSIND i T j .

Using Validation and Test Data

When you have sufficient data, you can subdivide your data into three parts: training, validation, and test data. During the selection process, models are fit on the training data, and the prediction error for the models so obtained 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 to decide which effects to include as the selection process proceeds. Finally, after a selected model has been obtained, the test set can be used 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 which 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) note 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.

PROC QUANTSELECT provides several methods for partitioning data into training, validation, and test data. You can provide data for each role in separate data sets that you specify with the **DATA=**, **TESTDATA=**, and **VALDATA=** options in the PROC QUANTSELECT procedure. An alternative method is to 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. The following statements randomly subdivide the **inData** data set to use 25% of the data for validation and 25% for testing, leaving 50% of the data for training:

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

If you need to exercise more control over the partitioning of the input data set, you can name a variable in the input data set and a formatted value of that variable to correspond to each role. The following statements assign roles to observations in the **inData** data set based on the value of the variable named **group** in that data set:

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

Observations whose value of the variable **group** is 'group 1' are assigned for testing, and those whose value is 'group 2' are assigned to training. All other observations are ignored.

You can also combine the use of the **PARTITION** statement with named data sets for specifying data roles. For example, the following statements reserve 40% of the **inData** data set for validation, leaving the remaining 60% for training:

```
proc quantselect data=inData testData=inTest;
  partition fraction(validate=0.4);
  ...
run;
```


Data for testing are supplied in the `inTest` data set. Because a `TESTDATA=` data set is specified, additional observations for testing cannot be reserved by specifying a `PARTITION` statement.

When you use a `PARTITION` statement, the output data set that is created by an `OUTPUT` statement contains a character variable `_ROLE_` whose values 'TRAIN', 'TEST', and 'VALIDATE' indicate the role of each observation. `_ROLE_` is blank for observations that were not assigned to any of these three roles. When the input data set specified in the `DATA=` option in the `PROC QUANTSELECT` statement contains an `_ROLE_` variable, no `PARTITION` statement is used, and the `TESTDATA=` and `VALDATA=` options are not specified, then the `_ROLE_` variable is used to define the roles of each observation. This is useful when you want to rerun `PROC QUANTSELECT` but use the same data partitioning as you used in a previous `PROC QUANTSELECT` step. For example, the following statements use the same data for testing and training in both `PROC QUANTSELECT` steps:

```
proc quantselect data=inData;
  partition fraction(test=0.5);
  model y=x1-x10 / selection=forward;
  output out=outDataForward;
run;

proc quantselect data=outDataForward;
  model y=x1-x10 / selection=backward;
run;
```

When you have reserved observations for training, validation, and testing, a model that is fit on the training data is scored on the validation and test data, and the average check loss, denoted by ACL, is computed separately for each of these subsets. The ACL for each data role is the sum of check losses 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 use the `STOP=VALIDATE` *method-option* to specify the validation ACL as the `STOP=` criterion in the `SELECTION=` option in the `MODEL` statement. At step k of the selection process, the best candidate effect to enter or leave the current model is determined. The “best candidate” means the effect that gives the best value of the `SELECT=` criterion that does not need to be based on the validation data. The validation ACL for the model with this candidate effect added is computed. 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` *method-option* in the `SELECTION=` option in the `MODEL` statement, the validation ACL is computed for the models at each step of the selection process. The model that yields the smallest validation ACL and contains the fewest effects is selected.

Using the Validation ACL as the SELECT= Criterion

You request the validation ACL as the selection criterion by specifying the `SELECT=VALIDATE` *method-option* in the `SELECTION=` option in the `MODEL` statement. At step k of the selection process, the validation ACL is computed for each model where a candidate for entry is added or candidate for removal is dropped. The selected candidate for entry or removal is the one that yields a model with the minimal validation ACL.

Displayed Output

The following sections describe the output that is displayed by PROC QUANTSELECT. The output is organized into various tables, which are discussed in the order of appearance. The contents of a table might change depending on the options you specify.

Model Information

The “Model Information” table displays basic information about the data sets and the settings used to control effect selection. These settings include the following:

- the selection method
- the criteria used to select effects, stop the selection, and choose the selected model
- the effect hierarchy enforced

The ODS name of the “Model Information” table is ModelInfo.

Number of Observations

The “Number of Observations” table displays the number of observations read from the input data set and the number of observations used in the analysis. If you use a **PARTITION** statement, the table also displays the number of observations used for each data role. If you specify **TESTDATA=** or **VALDATA=** data sets in the PROC QUANTSELECT statement, then “Number of Observations” tables are also produced for these data sets. The ODS name of the “Number of Observations” table is NObs.

Class Level Information

The “Class Level Information” table lists the levels of every variable specified in the **CLASS** statement. The ODS name of the “Class Level Information” table is ClassLevelInfo.

Class Level Coding

The “Class Level Coding” table shows the coding used for every variable specified in the **CLASS** statement. The ODS name of the “Class Level Coding” table is ClassLevelCoding.

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. The ODS name of the “Dimensions” table is Dimensions.

Candidates

The “Candidates” table displays the effect name and value of the criterion used to select entering or departing effects at each step of the selection process. The effects are displayed in sorted order from best to worst of the selection criterion. You request this table with the **DETAILS=** option in the **MODEL** statement. The ODS name of the “Candidates” table is either `EntryCandidates` for addition candidates or `RemovalCandidates` for removal candidates.

Selection Summary

The “Selection Summary” table displays details about the sequence of steps of the selection process. For each step, the effect that entered or dropped out is displayed along with the statistics used to select the effect, stop the selection, and choose the selected model. You can request that additional statistics be displayed with the **STATS=** option in the **MODEL** statement. For all criteria that you can use for effect selection, the steps at which the optimal values of these criteria occur are also indicated. The ODS name of the “Selection Summary” table is `SelectionSummary`.

Stop Reason

The “Stop Reason” table displays the reason why the selection stopped. [Table 96.14](#) shows the possible stop reasons.

Table 96.14 Reasons for Stopping

Stop Reason	Description
1	The selected model is a perfect fit.
2	The specified maximum number of steps has been reached.
3	The specified maximum number of effects are in the model.
4	The specified minimum number of effects are in the model.
5	The stopping criterion found a local optimum.
6	No suitable add or drop candidate is available.
7	All effects are in the model.
8	All effects have been dropped.
9	The sequence of effect additions and removals is cycling.
10	Adding or dropping any effect does not improve the SELECT= criterion.
11	No effect is significant at the specified significance level for entry or significance level for staying levels.
12	All remaining effects are required.

The ODS name of the “Stop Reason” table is `StopReason`.

Selection Reason

The “Selection Reason” table displays how the final selected model is determined. [Table 96.15](#) shows the possible selection reasons:

Table 96.15 Selection Reasons

Selection Reason	Description
1	The last valid model that occurs in the selection process is the final model.
2	The first model with the minimum CHOOSE= criterion value in the selection process is the final model.

The ODS name of the “Selection Reason” table is SelectionReason.

Selected Effects

The “Selected Effects” table displays a string that contains the list of effects in the selected model. The ODS name of the “Selected Effects” table is SelectedEffects.

Fit Statistics

The “Fit Statistics” table displays fit statistics for the selected model. The statistics displayed include the following:

- OBJ, the sum of check losses. It is calculated as the minimized objective function value for the fit.
- R1, a measure between 0 and 1 that indicates the portion of the (corrected) total variation attributed to the fit rather than left to residual error. It is calculated as one minus OBJ(Model) divided by OBJ(Total).
- Adj R1, the adjusted R1, a version of R1 that has been adjusted for degrees of freedom. It is calculated as

$$\bar{R}1 = 1 - \frac{(n - i)(1 - R1)}{n - p}$$

where i is equal to 1 if there is an intercept and 0 otherwise, n is the number of observations used to fit the model, and p is the number of parameters in the model.

- fit criteria AIC, AICC, and SBC.
- the average check losses (ACL) on the training, validation, and test data. See the section “[Using Validation and Test Data](#)” on page 7798 for details.

You can request “Fit Statistics” tables for the models at each step of the selection process with the **DETAILS=** option in the **MODEL** statement. The ODS name of the “Fit Statistics” table is FitStatistics.

Parameter Estimates

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

- the parameter label that 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 standard parameter estimate, which is computed on a standardized design matrix. Let $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2)$ denote the original design matrix, where \mathbf{X}_1 is the submatrix for all the forced-in effects, and \mathbf{X}_2 is the submatrix for the rest of the effects that are subject to selection. Let

$$\mathbf{X}_2^* = [\mathbf{I} - \mathbf{X}_1(\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1']\mathbf{X}_2 \text{ and } \mathbf{X}_2^{**} = s_Y \mathbf{X}_2^* \left[\frac{\text{diag}(\mathbf{X}_2^{*'}\mathbf{X}_2^*)}{n - p_1} \right]^{-\frac{1}{2}}$$

where p_1 is the rank of \mathbf{X}_1 and $s_Y = \sqrt{\frac{\mathbf{Y}^{*'}\mathbf{Y}^*}{n - p_1}}$ with $\mathbf{Y}^* = [\mathbf{I} - \mathbf{X}_1(\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1']\mathbf{Y}$.

Then standard parameter estimates are defined as $(\mathbf{0}, \boldsymbol{\beta}_2^{**})$, where $(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2^{**})$ are the parameter estimates computed on the standardized design matrix $(\mathbf{X}_1, \mathbf{X}_2^{**})$.

You can also use the **DETAILS=** option in the **MODEL** statement to request “Parameter Estimates” tables for the models at each step of the selection process. The ODS name of the “Parameter Estimates” table is **ParameterEstimates**.

Parameter Estimates for Quantile Process

The “Parameter Estimates for Quantile Process” table contains the parameter estimates for the quantile process of the final selected model. The following statements show how you can request the output data set of this table by using the ODS OUTPUT statement:

```
proc quantselect data=Data;
  ods output ProcessEst=outProcessEst;
  model y=x1-x10 / selection=forward quantile=process;
run;
proc print data=outProcessEst;
run;
```

The output data set contains the following variables:

- **QuantileLabel**, the label of quantile levels
- **QuantileLevel**, the quantile levels
- variables for parameter estimates

Given the quantile-level grid for the quantile process,

$$\{0 = \tau_{(0)} \leq \tau_{(1)} \leq \cdots \leq \tau_{(s)} \leq \tau_{(s+1)} = 1\}$$

The i th observation in the “Parameter Estimates for Quantile Process” table corresponds to the optimal solution of the i th quantile level in the quantile-level grid. The i th **QuantileLabel** value is in the form of t_i , and the i th **QuantileLevel** value is equal to $\tau_{(i)}$. For more information about the quantile-level grid, see the section “[Quantile Process Regression](#)” on page 7791.

ODS Table Names

PROC QUANTSELECT assigns a name to each table it creates. You can use these names to refer to the table when you use the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in [Table 96.16](#).

For more information about ODS, see Chapter 20, “Using the Output Delivery System.”

Table 96.16 ODS Tables Produced by PROC QUANTSELECT

ODS Table Name	Description	Statement	Option
BSplineDetails	B-spline basis details	EFFECT	DETAILS
Dimensions	Number of effects and parameters	MODEL	Default
EntryCandidates	Entry effect ranking	MODEL	DETAILS=
FitStatistics	Selected model fit statistics	MODEL	Default
RemovalCandidates	Removal effect ranking	MODEL	DETAILS=
ClassLevelCoding	Classification variable coding	CLASS	SHOWCODING
ClassLevelInfo	Classification variable levels	CLASS	Default
CollectionLevelInfo	Levels of collection effects	EFFECT	DETAILS
MMLevelInfo	Levels of multimember effects	EFFECT	DETAILS
ModelInfo	Model information	MODEL	Default
NObs	Number of observations	MODEL	Default
ParameterNames	Labels for column names in the design matrix	PROC	OUTDESIGN(names)
ParameterEstimates	Selected model parameter estimates	MODEL	Default
PolynomialDetails	Polynomial details	EFFECT	DETAILS
PolynomialScaling	Polynomial scaling	EFFECT	DETAILS
ProcessEst	Parameter estimates for quantile process	MODEL	QUANTILE
SelectedEffects	List of selected effects	MODEL	Default
SelectionSummary	Selection summary	MODEL	Default
StopReason	Reason why selection stopped	MODEL	Default
TPFSplineDetails	Thin-plate spline basis details	EFFECT	DETAILS

ODS Graphics

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

Before you create graphs, ODS Graphics must be enabled (for example, by specifying the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section “Enabling and Disabling ODS Graphics” on page 609 in Chapter 21, “Statistical Graphics Using ODS.”

The overall appearance of graphs is controlled by ODS styles. Styles and other aspects of using ODS Graphics are discussed in the section “A Primer on ODS Statistical Graphics” on page 608 in Chapter 21, “Statistical Graphics Using ODS.”

PROC QUANTSELECT assigns a name to each graph it creates using ODS. You can use these names to refer to the graphs when using ODS. The names are listed in Table 96.17.

Table 96.17 ODS Graphics Produced by PROC QUANTSELECT

ODS Graph Name	Plot Description	PLOTS Option
ACLPlot	Average check loss by step	ACL
AICCPLOT	Corrected Akaike’s information criterion by step	CRITERIA(UNPACK)
AICPlot	Akaike’s information criterion by step	CRITERIA(UNPACK)
AdjR1Plot	Adjusted quantile regression R by step	CRITERIA(UNPACK)
ChooseCriterionPlot	CHOOSE= criterion by step	COEFFICIENTS(UNPACK)
CoefficientPanel	Coefficients and CHOOSE= criterion by step	COEFFICIENTS
CoefficientPlot	Coefficients by step	COEFFICIENTS(UNPACK)
CriterionPanel	Fit criteria by step	CRITERIA
SBCPlot	Schwarz Bayesian information criterion by step	CRITERIA(UNPACK)
ValidateACLPlot	Average square error on validation data by step	CRITERIA(UNPACK)

Example: QUANTSELECT Procedure

Example 96.1: Simulation Study

This simulation study exemplifies the unity of motive and effect for the PROC QUANTSELECT procedure. The following statements generate a data set that is based on 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 of the data set indicates the true quantile level of the response y conditional on $\mathbf{x} = (x_1, \dots, x_p)$.

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

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

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 x_2 and x_3 and drop all the other effects in this data set at $\tau = 0.1$. By the same rationale, x_1 and x_3 should be selected at $\tau = 0.5$ with $\beta_1(0.5) = 0.4$ and $\beta_3(0.5) \approx -0.810882$, and x_1 and x_2 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 QUANTSELECT with the adaptive LASSO method:

```

proc quantselect data=analysisData;
    model y= x1-x&p / quantile=0.1 0.5 0.9
        selection=lasso(adaptive);
    output out=out p=pred;
run;

```

Output 96.1.1 shows that, by default, the CHOOSE= and STOP= options are both set to SBC.

Output 96.1.1 Model Information

The QUANTSELECT Procedure

Model Information	
Data Set	WORK.ANALYSISDATA
Dependent Variable	y
Selection Method	Adaptive LASSO
Quantile Type	Single Level
Stop Criterion	SBC
Choose Criterion	SBC

The selected effects and the relevant estimates are shown in Output 96.1.2 for $\tau = 0.1$, Output 96.1.3 for $\tau = 0.5$, and Output 96.1.4 for $\tau = 0.9$. You can see that the adaptive LASSO method correctly selects active effects for all three quantile levels.

Output 96.1.2 Parameter Estimates at $\tau = 0.1$ **Selected Effects:** Intercept x2 x3

Parameter Estimates			
Parameter	DF	Estimate	Standardized Estimate
Intercept	1	0.011793	0
x2	1	-0.228709	-0.218287
x3	1	-1.379907	-0.784520

Output 96.1.3 Parameter Estimates at $\tau = 0.5$ **Selected Effects:** Intercept x1 x3

Parameter Estimates			
Parameter	DF	Estimate	Standardized Estimate
Intercept	1	0.011778	0
x1	1	0.425843	0.118792
x3	1	-0.863316	-0.490822

Output 96.1.4 Parameter Estimates at $\tau = 0.9$ **Selected Effects:** Intercept x1 x2

Parameter Estimates			
Parameter	DF	Estimate	Standardized Estimate
Intercept	1	-0.007738	0
x1	1	0.782942	0.218407
x2	1	0.576445	0.550177

The QUANTSELECT procedure can perform effect selection not only at a single quantile level but also for the entire quantile process. You can specify the QUANTILE=PROCESS option to do effect selection for the entire quantile process. With the QUANTILE=PROCESS option specified, the ParameterEstimates table produced by the QUANTSELECT procedure actually shows the mean prediction model of y conditional on \mathbf{x} . In this simulation study, the true mean model is

$$E(y|\mathbf{x}) = \mathbf{x}\boldsymbol{\beta}$$

where

$$\beta_1 = E(U) - 0.1 = 0.4$$

$$\beta_2 = E(U^2) - 0.25 \approx 0.083333$$

$$\beta_3 = E(\exp(U)) - \exp(0.9) \approx -0.741321$$

$$\beta_4 = \dots = \beta_p = 0$$

The following statements perform effect selection for the quantile process with the forward selection method.

```
proc quantselect data=analysisData;
  model y= x1-x&p / quantile=process(ntau=all)
    selection=forward;
run;
```

Output 96.1.5 shows that, by default, the SELECT= and STOP= options are both set to SBC. The selected effects and the relevant estimates for the conditional mean model are shown in Output 96.1.6.

Output 96.1.5 Model Information
The QUANTSELECT Procedure

Model Information	
Data Set	WORK.ANALYSISDATA
Dependent Variable	y
Selection Method	Forward
Quantile Type	Process
Select Criterion	SBC
Stop Criterion	SBC
Choose Criterion	SBC

Output 96.1.6 Parameter Estimates

Parameter Estimates			
Parameter	DF	Estimate	Standardized Estimate
Intercept	1	0.007833	0
x1	1	0.418825	0.116834
x2	1	0.094791	0.090472
x3	1	-0.785686	-0.446687

Linear regression is the most popular method for estimating conditional means. The following statements show how to select effects with the GLMSELECT procedure, and Output 96.1.7 shows the resulting selected effects and their estimates. You can see that the mean estimates from the QUANTSELECT procedure are similar to those from the GLMSELECT procedure. However, quantile regression can provide detailed distribution information, which is not available from linear regression.

```
proc glmselect data=analysisData;
  model y= x1-x3 / selection=forward(select=sbc stop=sbc choose=sbc);
run;
```

Output 96.1.7 Parameter Estimates**The GLMSELECT Procedure
Selected Model**

Parameter Estimates				
Parameter	DF	Estimate	Standard	
			Error	t Value
Intercept	1	-0.010143	0.043129	-0.24
x1	1	0.434553	0.057385	7.57
x2	1	0.114183	0.016771	6.81
x3	1	-0.797194	0.028156	-28.31

Example 96.2: Econometric Growth Data

This example shows how you can use the QUANTREG procedure to further analyze the final selected models from the QUANTSELECT procedure, and how you can find the set of observations for a specified range of quantile levels. The data under investigation contain economic growth rate records for countries during two time periods: 1965–1975 and 1975–1985. This data set comes from a study by Barro and Lee (1994) and is also analyzed in the section “[Example 95.2: Quantile Regression for Econometric Growth Data](#)” on page 7729 of Chapter 95, “[The QUANTREG Procedure](#).”

The data set contains 161 observations and 16 variables. The variables, which are listed in [Table 96.18](#), include the national GDP growth rates (GDPR), 14 covariates, and a name variable (Country) that identifies the countries in one of the two periods.

Table 96.18 Variables for Econometric Growth Data

Variable	Description
Country	Country's name and time period
GDPR	Annual change of per capita GDP
lgdp2	Initial per capita GDP
mse2	Male secondary education
fse2	Female secondary education
fhe2	Female higher education
mhe2	Male higher education
lexp2	Life expectancy
lintr2	Human capital
gedy2	Education/GDP
ly2	Investment/GDP
gcony2	Public consumption/GDP
lblackp2	Black market premium
pol2	Political instability
ttrad2	Growth rate terms trade
period	Time period

The goal is to compare the effect of the covariates on GDPR at different quantile levels. The following statements perform effect selection at three quantile levels (τ): 0.1, 0.5, and 0.9.

```
data growth;
  length Country$ 22;
  input Country GDPR lgdp2 mse2 fse2 fhe2 mhe2 lexp2 lintr2 gedy2
        Iy2 gcony2 lblakp2 pol2 ttrad2 @@;
  if(index(country,'75')) then period='65-75';
  if(index(country,'85')) then period='75-85';
  datalines;
Algeria75          .0415 7.330 .1320 .0670 .0050 .0220 3.880 .1138 .0382
                   .1898 .0601 .3823 .0833 .1001
Algeria85          .0244 7.745 .2760 .0740 .0070 .0370 3.978 -.107 .0437
                   .3057 .0850 .9386 .0000 .0657
Argentina75        .0187 8.220 .7850 .6200 .0740 .1660 4.181 .4060 .0221
                   .1505 .0596 .1924 .3575 -.011
Argentina85        -.014 8.407 .9360 .9020 .1320 .2030 4.211 .1914 .0243

... more lines ...

Zambia75           .0654 .1224 .9393 .7022 -.007
                   .0120 6.989 .3760 .1190 .0130 .0420 3.757 .4388 .0339
                   .3688 .2513 .3945 .0000 -.032
Zambia85           -.046 7.109 .4200 .2740 .0110 .0270 3.854 .8812 .0477
                   .1632 .2637 .6467 .0000 -.033
Zimbabwe75         .0320 6.860 .1450 .0170 .0080 .0450 3.833 .7156 .0337
                   .2276 .0246 .1997 .0000 -.040
Zimbabwe85         -.011 7.180 .2200 .0650 .0060 .0400 3.944 .9296 .0520
                   .1559 .0518 .7862 .7161 -.024
;

proc quantselect data=growth;
  class period;
  model GDPR = period lgdp2 mse2 fse2 fhe2 mhe2 lexp2
              lintr2 gedy2 Iy2 gcony2 lblakp2 pol2 ttrad2
              / quantile=0.1 0.5 0.9 selection=backward(choose=sbc sh=5);
run;
```

The SELECTION=BACKWARD option specifies the BACKWARD method as the effect selection method, and the CHOOSE=SBC option specifies the Schwarz Bayesian information criterion for choosing the final selected effects. The estimates for the final selected effects are shown in [Output 96.2.1](#) for $\tau = 0.1$, [Output 96.2.2](#) for $\tau = 0.5$, and [Output 96.2.3](#) for $\tau = 0.9$.

Output 96.2.1 Parameter Estimates at $\tau = 0.1$

The QUANTSELECT Procedure Quantile Level = 0.1

Selected Effects: Intercept period lgdp2 mse2 lexp2 lintr2 Iy2 gcony2 lblakp2 pol2 ttrad2

Output 96.2.1 *continued*

Parameter Estimates			
Parameter	DF	Estimate	Standardized Estimate
Intercept	1	0.048847	0
period 65-75	1	0.011861	0.238272
lgdp2	1	-0.024613	-0.947421
mse2	1	0.016031	0.554367
lexp2	1	0.033898	0.277298
lintr2	1	-0.001877	-0.192986
ly2	1	0.067877	0.240002
gcony2	1	-0.176072	-0.438350
lblakp2	1	-0.026364	-0.326506
pol2	1	-0.022975	-0.223264
ttrad2	1	0.096604	0.146071

Output 96.2.2 Parameter Estimates at $\tau = 0.5$

Selected Effects: Intercept period lgdp2 mse2 lexp2 lintr2 ly2 gcony2 lblakp2 pol2 ttrad2

Parameter Estimates			
Parameter	DF	Estimate	Standardized Estimate
Intercept	1	-0.040264	0
period 65-75	1	0.008913	0.179063
lgdp2	1	-0.025823	-0.993996
mse2	1	0.014161	0.489697
lexp2	1	0.062163	0.508527
lintr2	1	-0.002688	-0.276345
ly2	1	0.068294	0.241476
gcony2	1	-0.096543	-0.240354
lblakp2	1	-0.025265	-0.312892
pol2	1	-0.019387	-0.188396
ttrad2	1	0.150668	0.227819

Output 96.2.3 Parameter Estimates at $\tau = 0.9$

Selected Effects: Intercept lgdp2 mse2 lexp2 lintr2 ly2 gcony2 lblakp2 ttrad2

Output 96.2.3 continued

Parameter Estimates			
Parameter	DF	Estimate	Standardized Estimate
Intercept	1	-0.011162	0
lgdp2	1	-0.032753	-1.260735
mse2	1	0.016583	0.573447
lexp2	1	0.073326	0.599845
lintr2	1	-0.003334	-0.342843
ly2	1	0.063929	0.226041
gcony2	1	-0.089998	-0.224060
lblakp2	1	-0.032253	-0.399439
ttrad2	1	0.213457	0.322760

Comparing the three quantile models, you can see that the final selected models for $\tau = 0.1$ and $\tau = 0.5$ have the same set of selected effects, but the final selected model for $\tau = 0.9$ excludes the effects for time period and political instability. In other words, if a country's annual change in per capita GDP represents the 90% quantile conditional on the explanatory effects, then its GDP growth rate seems consistent for both the 1965–1975 and 1975–1985 periods and resistant to political instability. In addition, if a country's GDP growth rate represents the 50% or less quantile conditional on the explanatory effects, then the country's 1975–1985 GDP growth rate seems lower than its 1965–1975 GDP growth rate, and the effect for political instability has a negative impact on its GDP growth rate.

To further investigate the impact of time period and political instability on GDP growth rate, you can use the QUANTREG procedure to test the final selected effects. In the previous statements, PROC QUANTSELECT creates a macro variable for the final selected model at each of the three quantile levels. For the current example, the macro variable `_QRSINDT1` contains the final model at $\tau = 0.1$; `_QRSINDT2` contains the final model at $\tau = 0.5$; and `_QRSINDT3` contains the final model at $\tau = 0.9$. The following statements show how to use `_QRSINDT2` to specify the model for the QUANTREG procedure at $\tau = 0.5$:

```
proc quantreg data=growth;
  class period;
  model GDPR = &_qrsindt2 / quantile=0.5;
  Time_Period: test period;
  Political_Instability: test pol2;
run;
```

Output 96.2.4 shows more information for the final selected model at $\tau = 0.5$. Output 96.2.5 and Output 96.2.6 show the test results for the effects of time period and political instability on GDP growth rate. You can see that both time period and political instability are significant for the $\tau = 0.5$ model.

Output 96.2.4 Parameter Estimates at $\tau = 0.5$ **The QUANTREG Procedure**

Parameter Estimates				
Parameter	DF	Estimate	95% Confidence Limits	
Intercept	1	-0.0403	-0.1529	0.0453
period	65-75	1	0.0089	0.0060 0.0139
period	75-85	0	0.0000	0.0000 0.0000
lgdp2	1	-0.0258	-0.0324	-0.0212
mse2	1	0.0142	0.0068	0.0182
lexp2	1	0.0622	0.0400	0.1199
lintr2	1	-0.0027	-0.0045	-0.0011
ly2	1	0.0683	0.0143	0.1077
gcony2	1	-0.0965	-0.1526	-0.0576
lblakp2	1	-0.0253	-0.0537	-0.0174
pol2	1	-0.0194	-0.0377	-0.0116
ttrad2	1	0.1507	0.0190	0.2436

Output 96.2.5 Test Results at $\tau = 0.5$

Test Time_Period Results				
Test				
Test	Statistic	DF	Chi-Square	Pr > ChiSq
Wald	13.9838	1	13.98	0.0002

Output 96.2.6 Test Results at $\tau = 0.5$

Test Political_Instability Results				
Test				
Test	Statistic	DF	Chi-Square	Pr > ChiSq
Wald	11.0589	1	11.06	0.0009

As mentioned earlier, `_QRSINDT1` and `_QRSINDT2` are identical, and `_QRSINDT3` excludes two effects from `_QRSINDT2`: time period and political instability. The following statements retest time period and political instability for the final selected model at $\tau = 0.9$:

```
proc quantreg data=growth;
  class period;
  model GDPGR = &_qrsindt2 / quantile=0.9;
  Time_Period:      test period;
  Political_Instability: test pol2;
  Period_and_Political: test period pol2;
run;
```

Output 96.2.7, Output 96.2.8, and Output 96.2.9 show the test results for the effects of time period and political instability on GDP growth rate at $\tau = 0.9$. You can see that time period, political instability, and their combination are all insignificant for the $\tau = 0.9$ model.

Output 96.2.7 Test Results at $\tau = 0.9$ **The QUANTREG Procedure**

Test Time_Period Results				
Test				
Test	Statistic	DF	Chi-Square	Pr > ChiSq
Wald	0.0001	1	0.00	0.9941

Output 96.2.8 Test Results at $\tau = 0.9$

Test Political_Instability Results				
Test				
Test	Statistic	DF	Chi-Square	Pr > ChiSq
Wald	0.1238	1	0.12	0.7250

Output 96.2.9 Test Results at $\tau = 0.9$

Test Period_and_Political Results				
Test				
Test	Statistic	DF	Chi-Square	Pr > ChiSq
Wald	0.1367	2	0.14	0.9339

Another interesting question for quantile regression is to find the observations for a certain range of quantile levels. For example, you might want to know which countries are winners in terms of conditional GDP growth rate at the $\tau = 0.9$ level. The following statements compute the $\tau = 0.9$ quantile predictions and then search, sort, and print the list of winner countries:

```
proc quantselect data=growth;
  class period;
  model GDP = period lgdp2 mse2 fse2 fhe2 mhe2 lexp2
              lintr2 gedy2 Iy2 gcony2 lblakp2 pol2 ttrad2
              / quantile=0.9 selection=backward(choose=sbc sh=5);
  output out=growth90Out p=Pred;
run;

data growth90;
  set growth90Out;
  drop lgdp2 mse2 fse2 fhe2 mhe2 lexp2 lintr2 gedy2 Iy2
        gcony2 lblakp2 ttrad2;
  where GDP-Pred >= -1E-4;
  GdpDiff = GDP-Pred;
run;

proc sort data=growth90;
  by GdpDiff;
run;
proc print data=growth90;
run;
```


Output 96.2.10 lists the countries whose conditional GDP growth rates are equal to or higher than their $\tau = 0.9$ quantile predictions.

Output 96.2.10 Countries with High Conditional GDP Growth Rates at $\tau = 0.9$ Level

Obs	Country	GDPR	pol2	period	Pred	GdpDiff
1	Canada75	0.0346	0.0047	65-75	0.034600	0.000000
2	Canada85	0.0240	0.0000	75-85	0.024000	0.000000
3	Congo75	0.0464	0.3385	65-75	0.046400	0.000000
4	Cyprus85	0.0709	0.6000	75-85	0.070900	0.000000
5	Finland75	0.0391	0.0000	65-75	0.039100	0.000000
6	Germany_West85	0.0214	0.0000	75-85	0.021400	0.000000
7	Ghana85	-0.0150	0.0500	75-85	-0.015000	0.000000
8	United_States75	0.0155	0.0015	65-75	0.015500	0.000000
9	Yemen85	0.0305	0.0730	75-85	0.030500	0.000000
10	Denmark85	0.0234	0.0000	75-85	0.023010	0.000390
11	Japan75	0.0636	0.0005	65-75	0.062519	0.001081
12	Jordan85	0.0593	0.5000	75-85	0.058201	0.001099
13	Sudan85	0.0007	0.7000	75-85	-0.000919	0.001619
14	Iran75	0.0538	0.0072	65-75	0.051880	0.001920
15	Spain75	0.0457	0.0014	65-75	0.043241	0.002459
16	Egypt85	0.0427	0.5500	75-85	0.038409	0.004291
17	Hong_Kong85	0.0649	0.0000	75-85	0.059040	0.005860
18	Bangladesh85	0.0133	0.6507	75-85	0.006816	0.006484
19	Rwanda75	0.0590	0.0500	65-75	0.050266	0.008734
20	Brazil75	0.0637	0.0011	65-75	0.050749	0.012951
21	Syria75	0.0601	0.2500	65-75	0.046072	0.014028
22	Botswana85	0.0512	0.0000	75-85	0.030626	0.020574

Example 96.3: Pollution and Mortality

This example shows how you can use the PARTITION statement and other options to control the effect selection process. The data for this example come from a study by McDonald and Schwing (1973). The data set contains 60 observations, 15 covariates, and one response variable. The response variable is the total age-adjusted mortality rate for Standard Metropolitan Statistical Areas in 1959–1961.

The following statements fit a median model for mortality rate conditional on a set of climate, demographic, and pollution covariates by using the forward selection method. Because linear terms alone might not be sufficient to fit this model, quadratic terms are also added in the MODEL statement. The FRACTION option of the PARTITION statement requests that 30% of the observations be used for validation and the remaining 70% of the observations for training. The HIER=SINGLE option in the MODEL statement forces the effect selection process to ignore quadratic effect candidates if their corresponding main effects are not in the model. The OUTPUT statement creates a SAS data set named OutData, which contains the variable _ROLE_. This variable shows the role of each observation that the PARTITION statement assigns.

```

data mortality;
  input index aap ajant ajult size65 npf nsch25 nfek ppsm snwp nowk nin3k
  hpi nopi sdpi datm DeathRate;
  label index="the index"
    aap="Average Annual Precipitation"
    ajant="Average January Temperature"
    ajult="Average July Temperature"
    size65="Size of Population older than 65"
    npf="Number of Members per Household"
    nsch25="Number of Years of Schooling for Persons over 25"
    nfek="Number of Households with fully Equipped Kitchens"
    ppsm="Population per Square Mile"
    snwp="Size of the Nonwhite Population"
    nowk="Number of Office Workers"
    nin3k="Number of Families with an Income less than $3000"
    hpi="Hydrocarbon Pollution Index"
    nopi="Nitric Oxide Pollution Index"
    sdpi="Sulfur Dioxide Pollution Index"
    datm="Degree of Atmospheric Moisture"
    DeathRate="Age-Adjusted Death Rate: Deaths per 100,000 Population";
  datalines;
1  36  27  71   8.1  3.34  11.4  81.5  3243   8.8  42.6  11.7  21
   15   59  59   921.870
2  35  23  72  11.1  3.14  11.0  78.8  4281   3.6  50.7  14.4   8
   10   39  57   997.875
3  44  29  74  10.4  3.21   9.8  81.6  4260   0.8  39.4  12.4   6
   6   33  54   962.354
4  47  45  79   6.5  3.41  11.1  77.5  3125  27.1  50.2  20.6  18
   8   24  56   982.291
5  43  35  77   7.6  3.44   9.6  84.6  6441  24.4  43.7  14.3  43

... more lines ...

11   42  56  1003.502
58  45  24  70  11.8  3.25  11.1  79.8  3678   1.0  44.8  14.0   7
   3   8   56   895.696
59  42  83  76   9.7  3.22   9.0  76.2  9699   4.8  42.2  14.5   8
   8   49  54   911.817
60  38  28  72   8.9  3.48  10.7  79.8  3451  11.7  37.5  13.0  14
   13   39  58   954.442
;

ods graphics on;
proc quantselect data=Mortality seed=800 plots=all;
  partition fraction(validate=0.3);
  model DeathRate = aap aap*aap ajant ajant*ajant ajult
    ajult*ajult size65 size65*size65 npf npf*npf nsch25
    nsch25*nsch25 nfek nfek*nfek ppsm ppsm*ppsm snwp snwp*snwp
    nowk nowk*nowk nin3k nin3k*nin3k hpi hpi*hpi nopi
    nopi*nopi sdpi sdpi*sdpi datm datm*datm
    / quantile=0.5 selection=forward(choose=val sh=8) hier=single;
  output out=OutData p=Pred;
run;

proc print data=OutData(obs=10); run;

```

Output 96.3.1 shows the selection summary. You can see that the best model is at step 13 for validation ACL, step 5 for the SBC, and step 14 for the AIC.

Output 96.3.1 Selection Summary

The QUANTSELECT Procedure

Selection Summary							
	Effect	Number	Validation				Adjusted
Step	Entered	Effects	AIC	AICC	SBC	ACL	R1
		In					
0	Intercept	1	276.5053	276.6005	278.2895	31.7900	0.0000
1	snwp	2	251.6460	251.9387	255.2144	23.9139	0.2455
2	sdpi	3	240.3445	240.9445	245.6971	20.3977	0.3355
3	nopi	4	238.3223	239.3480	245.4591	16.9704	0.3493
4	ppsm	5	239.3875	240.9664	248.3084	16.3677	0.3397
5	aap	6	226.5892	228.8595*	237.2943*	15.7333	0.4272
6	aap*aap	7	227.6860	230.7971	240.1754	14.8892	0.4177
7	ajult	8	228.5136	232.6279	242.7871	14.5477	0.4095
8	nin3k	9	229.4258	234.7199	245.4835	14.3532	0.4001
9	ajant	10	224.7397	231.4063	242.5816	13.5693	0.4276*
10	ppsm*ppsm	11	226.5785	234.8285	246.2046	12.4032	0.4114
11	hpi	12	228.5050	238.5696	249.9153	11.6356	0.3935
12	ajant*ajant	13	229.9796	242.1129	253.1740	11.1214	0.3776
13	nfek	14	231.9208	246.4035	256.8994	10.9947*	0.3573
14	snwp*snwp	15	221.3905*	238.5334	248.1533	13.3735	0.4234
15	ajult*ajult	16	223.3153	243.4635	251.8624	13.0557	0.4033
16	sdpi*sdpi	17	224.8099	248.3483	255.1411	14.2347	0.3847
17	size65	18	226.7756	254.1356	258.8910	14.3067	0.3613
18	nin3k*nin3k	19	223.8621	255.5288	257.7618	14.5143	0.3719
19	nfek*nfek	20	224.0342	260.5559	259.7180	14.6314	0.3591
20	datm	21	222.7062	264.7062	260.1742	15.4604	0.3561
* Optimal Value Of Criterion							

Output 96.3.2 shows the selected effects and the relevant estimates.

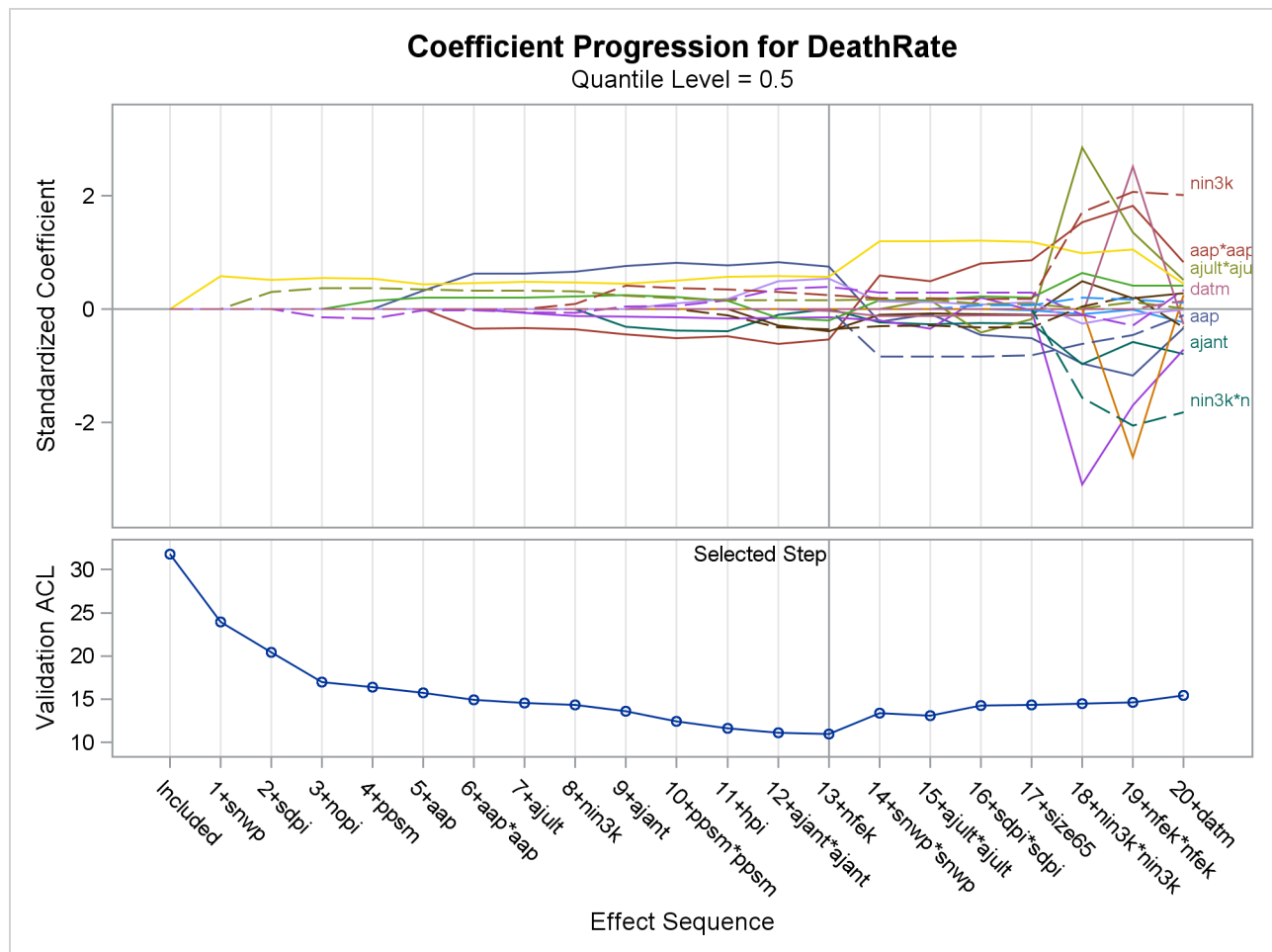
Output 96.3.2 Parameter Estimates

Selected Effects: Intercept aap aap*aap ajant ajant*ajant ajult nfek ppsm ppsm*ppsm snwp nin3k hpi nopi sdpi

Parameter Estimates			
Parameter	DF	Estimate	Standardized Estimate
Intercept	1	909.689797	0
aap	1	4.634741	0.750747
aap*aap	1	-0.047789	-0.533679
ajant	1	0.009723	0.001962
ajant*ajant	1	-0.020447	-0.389447
ajult	1	-1.672607	-0.146182
nfek	1	-0.323920	-0.030436
ppsm	1	-0.007194	-0.194141
ppsm*ppsm	1	0.000001906	0.534144
snwp	1	3.483423	0.574703
nin3k	1	3.228388	0.252681
hpi	1	-0.401693	-0.351016
nopi	1	0.795823	0.389110
sdpi	1	0.151049	0.152444

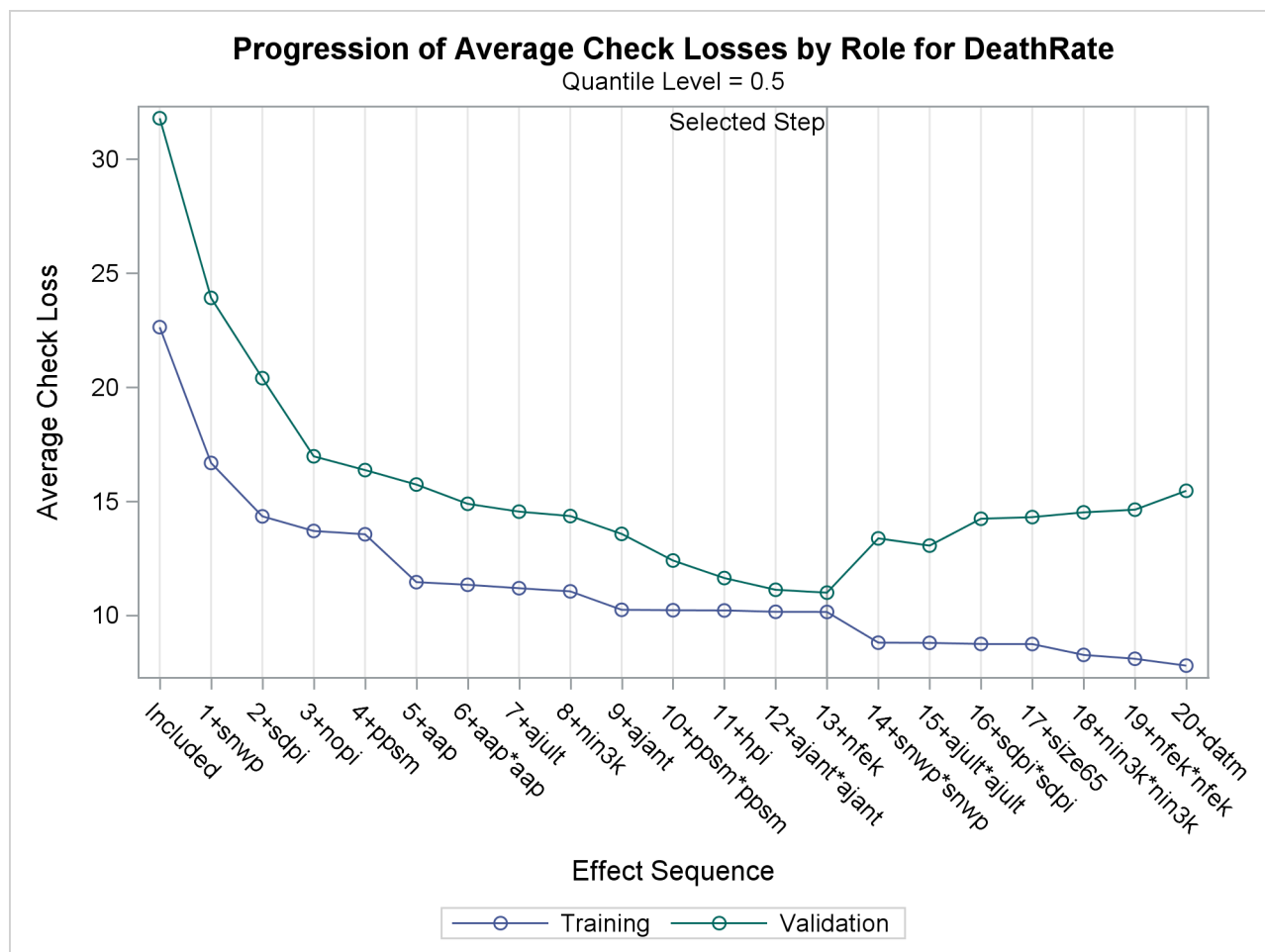
Output 96.3.3 shows the progression of the standardized parameter estimates as the selection process proceeds.

Output 96.3.3 Coefficient Panel



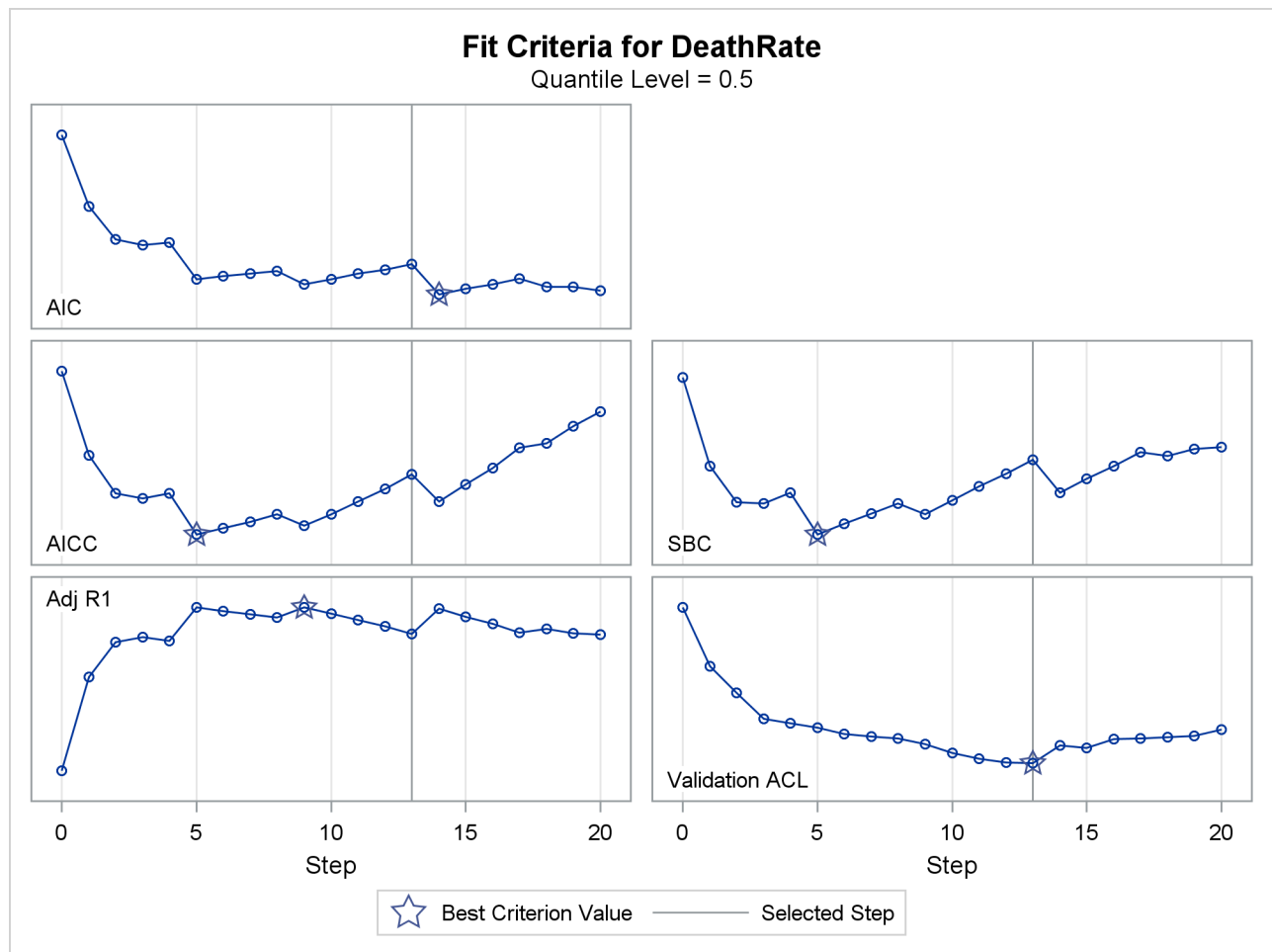
Output 96.3.4 shows the progression of the average check losses for training data and validation data as the selection process proceeds.

Output 96.3.4 Average Check Loss Plot



Output 96.3.5 shows the progression of five effect selection criteria as the selection process proceeds.

Output 96.3.5 Criterion Panel



Output 96.3.6 shows the first 10 observations of the OUTPUT data set.

Output 96.3.6 OUTPUT Data Set

Obs	index	aap	ajant	ajult	size65	nph	nsch25	nfek	ppsm	snwp	nowk	nin3k	hpi	nopi	sdpi	datm	DeathRate	Pred	_ROLE_
1	1	36	27	71	8.1	3.34	11.4	81.5	3243	8.8	42.6	11.7	21	15	59	59	921.87	932.36	TRAIN
2	2	35	23	72	11.1	3.14	11.0	78.8	4281	3.6	50.7	14.4	8	10	39	57	997.88	930.62	VALIDATE
3	3	44	29	74	10.4	3.21	9.8	81.6	4260	0.8	39.4	12.4	6	6	33	54	962.35	908.09	TRAIN
4	4	47	45	79	6.5	3.41	11.1	77.5	3125	27.1	50.2	20.6	18	8	24	56	982.29	983.55	TRAIN
5	5	43	35	77	7.6	3.44	9.6	84.6	6441	24.4	43.7	14.3	43	38	206	55	1071.29	1047.71	VALIDATE
6	6	53	45	80	7.7	3.45	10.2	66.8	3325	38.5	43.1	25.5	30	32	72	54	1030.38	1062.56	TRAIN
7	7	43	30	74	10.9	3.23	12.1	83.9	4679	3.5	49.2	11.3	21	32	62	56	934.70	934.70	TRAIN
8	8	45	30	73	9.3	3.29	10.6	86.0	2140	5.3	40.4	10.5	6	4	4	56	899.53	900.48	TRAIN
9	9	36	24	70	9.0	3.31	10.5	83.2	6582	8.1	42.5	12.6	18	12	37	61	1001.90	971.06	TRAIN
10	10	36	27	72	9.5	3.36	10.7	79.3	4213	6.7	41.0	13.2	12	7	20	59	912.35	927.10	VALIDATE

Example 96.4: Surface Fitting with Many Noisy Variables

This example is based on “[Example 25.1: Surface Fitting with Many Noisy Variables](#)” on page 949 in Chapter 25, “[The ADAPTIVEREG Procedure](#).” This example shows how you can use the EFFECT statement to select a nonlinear surface model for a data set that contains many nuisance variables.

Consider a simulated data set that contains a response variable and 10 continuous predictors. Each continuous predictor is sampled independently from the uniform distribution $U(0, 1)$. The true model of the artificial data set depends nonlinearly on two variables x_1 and x_2 :

$$y = \frac{40 \exp(8((x_1 - 0.5)^2 + (x_2 - 0.5)^2))}{\exp(8((x_1 - 0.2)^2 + (x_2 - 0.7)^2)) + \exp(8((x_1 - 0.7)^2 + (x_2 - 0.2)^2))}$$

The values of the response variable are generated by adding errors from the standard normal distribution $N(0, 1)$ to the true model. The generating mechanism is adapted from Gu et al. (1990). The following statements create an artificial data set that contains 400 observations for the purpose of effect selection and 10,201 observations of missing response values for the purpose of prediction:

```
%let p=10;
data artificial;
  drop i;
  array x{&p};
  do i=1 to 400;
    do j=1 to &p;
      x{j} = ranuni(1);
    end;
    yTrain = 40*exp(8*((x1-0.5)**2+(x2-0.5)**2))/
      (exp(8*((x1-0.2)**2+(x2-0.7)**2)) +
      exp(8*((x1-0.7)**2+(x2-0.2)**2)))+rannor(1);
    output;
  end;

  yTrain = .;
  do x1=0 to 1 by 0.01;
    do x2 = 0 to 1 by 0.01;
```



```

y = 40*exp(8*((x1-0.5)**2+(x2-0.5)**2))/
    (exp(8*((x1-0.2)**2+(x2-0.7)**2))+
    exp(8*((x1-0.7)**2+(x2-0.2)**2)));
output;
end;
end;
run;

```

The variables x3 through x10 are nuisance variables that can cause overfitting in your analysis. The following statements invoke the QUANTSELECT procedure to select effects, fit a model on the selected effects, and output the model predictions to an output data set Out:

```

%macro art;
  proc quantselect data=artificial algorithm=smooth;
    %do i=1 %to &p;
      effect sp&i = spline(x&i);
    %end;
    model yTrain =
      sp1 %do i=2 %to &p; |sp&i %end; @2/details=all;
    output out=Out p=pred;
  run;
%mend;

%art;

```

You can use the [EFFECT](#) statement to generate nonlinear effects and model a nonlinear surface. This example uses spline effects on variables and includes all the two-way interactions among these spline effects.

The ALGORITHM=SMOOTH option specifies the smoothing algorithm for model fitting. It takes approximately 2.8 seconds to select the model on a PC that has an Intel i7-2600 quad-core CPU and 64-bit Windows 7 Enterprise operation system. If you use the ALGORITHM=SIMPLEX option, which is default, it takes approximately 8.7 seconds for the same computation settings.

[Output 96.4.1](#) shows the model information. By default, the effect selection method is the stepwise method, and the selection criterion is SBC for the SELECT=, CHOOSE=, and STOP= options. The default quantile level is 0.5 for median regression.

Output 96.4.1 Model Information
The QUANTSELECT Procedure

Model Information	
Data Set	WORK.ARTIFICIAL
Dependent Variable	yTrain
Selection Method	Stepwise
Quantile Type	Single Level
Select Criterion	SBC
Stop Criterion	SBC
Choose Criterion	SBC

[Output 96.4.2](#) shows the best 10 entry candidates at the selection step. You can see that sp1*sp2 is the most important effect, followed by sp1 and sp2.

Output 96.4.2 Best 10 Entry Candidates at Step 1

Best 10 Entry Candidates		
Rank	Effect	SBC
1	sp1*sp2	-496.6752
2	sp1	165.9104
3	sp2	178.2126
4	sp3	213.4593
5	sp6	220.8471
6	sp7	222.0916
7	sp9	224.3185
8	sp4	224.7100
9	sp8	226.8373
10	sp5	227.2176

Output 96.4.3 shows the selection summary.

Output 96.4.3 Selection Summary
The QUANTSELECT Procedure
Quantile Level = 0.5

Selection Summary				
		Number	Number	SBC
Step	Effect Entered	Effects In	Parms In	
0	Intercept	1	1	195.8108
1	sp1*sp2	2	49	-496.6752*
* Optimal Value Of Criterion				

The following statements produce a graph that shows both the true model and the fitted model:

```
ods graphics on;
data pred;
  set out;
  where yTrain=.;
run;

%let off0 = offsetmin=0 offsetmax=0;
%let off0 = xaxisopts=(&off0) yaxisopts=(&off0);
%let eopt = location=outside valign=top textattrs=graphlabeltext;
proc template;
  define statgraph surfaces;
    begingraph / designheight=360px;
      layout lattice/columns=2;
      layout overlay / &off0;
        entry "True Model" / &eopt;
        contourplotparm z=y y=x2 x=x1;
      endlayout;
      layout overlay / &off0;
        entry "Fitted Model" / &eopt;
        contourplotparm z=pred y=x2 x=x1;
```

```

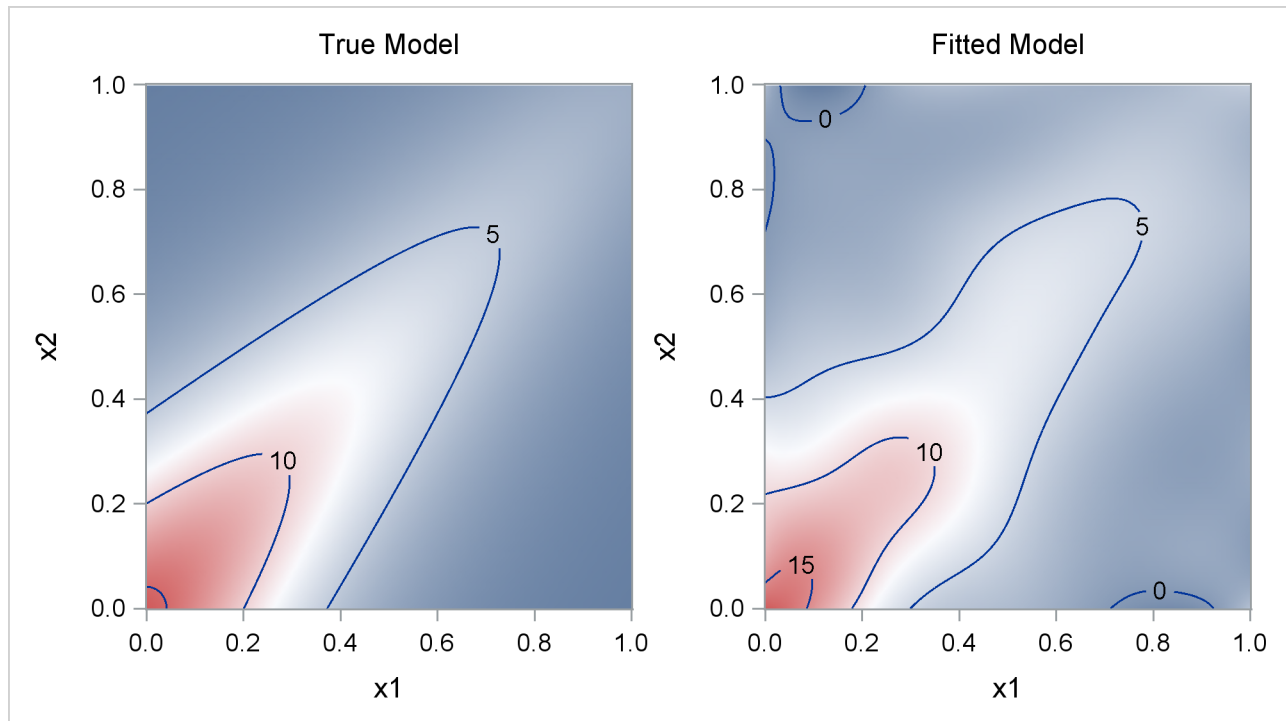
        endlayout;
    endlayout;
endgraph;
end;
run;

proc sgrender data=pred template=surfaces;
run;

```

Output 96.4.4 displays surfaces for both the true model and the fitted model. You can see that the fitted model nicely approximates the underlying true model.

Output 96.4.4 True Model and Fitted Model



Example 96.5: Quantile Process Regression

Quantile process regression fits quantile regression models for the entire range of quantile levels from 0 to 1, which can estimate the entire probability distribution of a response variable conditional on its covariates. This example demonstrates how you can conduct quantile process regression analysis by using the QUANTSELECT procedure.

Parameter Estimates for Quantile Process Regression

The following statements simulate the data set `analysisData`, solve a quantile process regression problem on the data set, and store the quantile process parameter estimates in the data set `quantProcessEst`. The data set `analysisData` contains one response `y` and two covariates, `x1` and `x2`. The distribution of `y` conditional on `x1` and `x2` gradually changes from a normal distribution to an exponential distribution.

```

%let seed=123;
%let n=6001;
%let model=x1 x2;
data analysisData;
  do i=1 to &n;
    x1=(i-1)/(&n-1);
    x2=(1-x1)*(1-x1);
    y =x1*ranexp(&seed)+x2*(rannor(&seed)-3);
    output;
  end;
run;

proc quantselect data=analysisData;
  ods output ProcessEst=quantProcessEst;
  model y = &model / quantile=process(ntau=all) selection=none;
run;

proc print data=quantProcessEst (obs=10);
run;

```

Output 96.5.1 shows the first 10 parameter estimates of the quantile process. For more information about parameter estimates for the quantile process, see the section “Parameter Estimates for Quantile Process” on page 7803.

Output 96.5.1 Quantile Process Parameter Estimates

Obs	QuantileLabel	QuantileLevel	Intercept	x1	x2
1	t0	0.000000	-1.3799	1.3982	-4.3156
2	t1	0.000121	-1.3799	1.3982	-4.3156
3	t2	0.000248	-0.8210	0.8386	-5.0420
4	t3	0.000283	0.2030	-0.2006	-6.3677
5	t4	0.000322	0.3065	-0.3098	-6.5001
6	t5	0.000436	0.4713	-0.4854	-6.7102
7	t6	0.000541	0.2788	-0.2812	-6.2666
8	t7	0.000592	0.1957	-0.1939	-6.0724
9	t8	0.000760	0.1940	-0.1915	-6.0704
10	t9	0.000880	0.1987	-0.1963	-6.0765

To reduce the computation complexity, you can also approximate the quantile process regression by using the NTAU= n suboption of the QUANTILE=PROCESS option. The following statements approximate the quantile process by using NTAU=10:

```

proc quantselect data=analysisData;
  ods output ProcessEst=quantApproxProcessEst;
  model y = &model / quantile=process(ntau=10) selection=none;
run;

proc print data=quantApproxProcessEst;
run;

```

Output 96.5.2 shows the approximate quantile process parameter estimates. If you specify the NTAU= n option, the approximate quantile process is computed at n equally spaced quantile levels: $\{\frac{1}{n+1}, \dots, \frac{n}{n+1}\}$ besides three control quantile levels $\{0, 0.5, 1\}$.

Output 96.5.2 Approximate Quantile Process Parameter Estimates

Obs	QuantileLabel	QuantileLevel	Intercept	x1	x2
1	t0	0.000000	-1.3799	1.3982	-4.3156
2	t1	0.090909	0.6051	-0.5449	-4.8257
3	t2	0.181818	0.5273	-0.3595	-4.3719
4	t3	0.272727	0.3871	-0.0764	-3.9561
5	t4	0.363636	0.3381	0.0980	-3.6935
6	t5	0.454545	0.1747	0.4300	-3.2961
7	t6	0.500000	0.0197	0.6928	-3.0160
8	t7	0.545455	-0.1102	0.9365	-2.7518
9	t8	0.636364	-0.3543	1.4418	-2.2749
10	t9	0.727273	-0.6169	2.0165	-1.7714
11	t10	0.818182	-1.0307	2.8540	-1.0614
12	t11	0.909091	-1.4056	3.8941	-0.3141
13	t12	1.000000	-1.9936	9.1022	1.2334

Observation Quantile Levels

Quantile process regression can estimate observation quantile levels for any valid observations. For more information, see the section “[Observation Quantile Level](#)” on page 7791. You can convert an observation quantile level to the percentage of the response value conditional on its covariate values. In the following statements, the **OUTPUT** statement outputs observation quantile levels:

```
proc quantselect data=analysisData;
  model y = &model / quantile=process selection=none;
  output out=outQuantLev p ql;
run;

proc print data=outQuantLev (obs=10);
run;
```

Output 96.5.3 shows the first 10 observations of the OUT=outQuantLev data set, in which the variable ql_y contains the observation quantile levels.

Output 96.5.3 Observation Quantile Levels

Obs	i	x1	x2	y	p_y	ql_y
1	1	0	1.00000	-2.34428	-2.98693	0.74850
2	2	.000166667	0.99967	-2.74004	-2.98580	0.59681
3	3	.000333333	0.99933	-2.03681	-2.98467	0.83420
4	4	.000500000	0.99900	-3.65071	-2.98354	0.24282
5	5	.000666667	0.99867	-3.36232	-2.98240	0.35729
6	6	.000833333	0.99833	-2.27148	-2.98127	0.77046
7	7	.001000000	0.99800	-3.30503	-2.98014	0.38186
8	8	.001166667	0.99767	-3.39110	-2.97901	0.34331
9	9	.001333333	0.99734	-2.63652	-2.97788	0.63473
10	10	.001500000	0.99700	-3.62622	-2.97675	0.24533

Observationwise Distribution Estimation

Quantile process regression can estimate the entire distribution of a response variable conditional on its covariates. The following statements use the IML procedure to create macro variables for observation indices, observation quantile levels, and observation mean predictions and create a data set, `distData`, that contains all quantile levels and quantile predictions for the specified observations:

```
proc iml;
  Obs  = {1 3001 6001};
  nObs = ncol(Obs);
  call symputx("nObs",nObs);

  use analysisData;
  read all var {&model} into x;
  read all var {"y"} into y;
  close analysisData;

  use outQuantLev;
  read all var {"ql_y"} into ql;
  read all var {"p_y"} into qm;
  close outQuantLev;

  use quantProcessEst;
  read all var {"Intercept"} into beta0;
  read all var {&model} into beta;
  read all var {"QuantileLevel"} into qLev;
  close quantProcessEst;

  nTau = nrow(qLev);
  qPrCs = j((nTau*nObs),3);
  obsInfo = j(nObs,6);
  obsIndex = "_Obs1": "_Obs&nObs";
  levNames = "_qLev1": "_qLev&nObs";
  qtNames = "_qt1": "_qt&nObs";
  qmNames = "_qMean1": "_qMean&nObs";

  do j=1 to nObs;
    iObs = Obs[j];
    call symputx(obsIndex[j],iObs);
    call symputx(qtNames[j], (y[iObs]));
    call symputx(levNames[j], (ql[iObs]));
    call symputx(qmNames[j], (qm[iObs]));
    obsInfo[j,1]=iObs;
    obsInfo[j,2]=y[iObs];
    obsInfo[j,3]=x[iObs,1];
    obsInfo[j,4]=x[iObs,2];
    obsInfo[j,5]=ql[iObs];
    obsInfo[j,6]=qm[iObs];
    Quantiles = beta0 + beta*t(x[iObs,]);
    call sort(Quantiles,1);
    qPrCs[((j-1)*nTau+1):(j*nTau),1]=iObs;
    qPrCs[((j-1)*nTau+1):(j*nTau),2]=qLev;
    qPrCs[((j-1)*nTau+1):(j*nTau),3]=Quantiles;
  end;
```

```

obsInfoColName = {"Index" "Response Value" "x1" "x2"
                  "Quantile Level" "Mean Prediction"};
obsInfoLabel = {"Information for Specified Observations"};
print obsInfo[colname=obsInfoColName label=obsInfoLabel];

create distData from qPrCs[colname={"iObs" "qLev" "Quantiles"}];
append from qPrCs;
close distData;
quit;

```

Output 96.5.4 shows the observation information table for the specified observations.

Output 96.5.4 Information for Observations 1, 3001, and 6001

Information for Specified Observations						
	Index	Response			Quantile	
		Value	x1	x2	Level	Mean Prediction
ROW1	1	-2.34428	0	1	0.748503	-2.986932
ROW2	3001	-0.557309	0.5	0.25	0.3433134	-0.280479
ROW3	6001	0.9299095	1	0	0.5788423	1.044977

The following statements plot the conditional cumulative distribution functions (CDFs) for the specified observations:

```

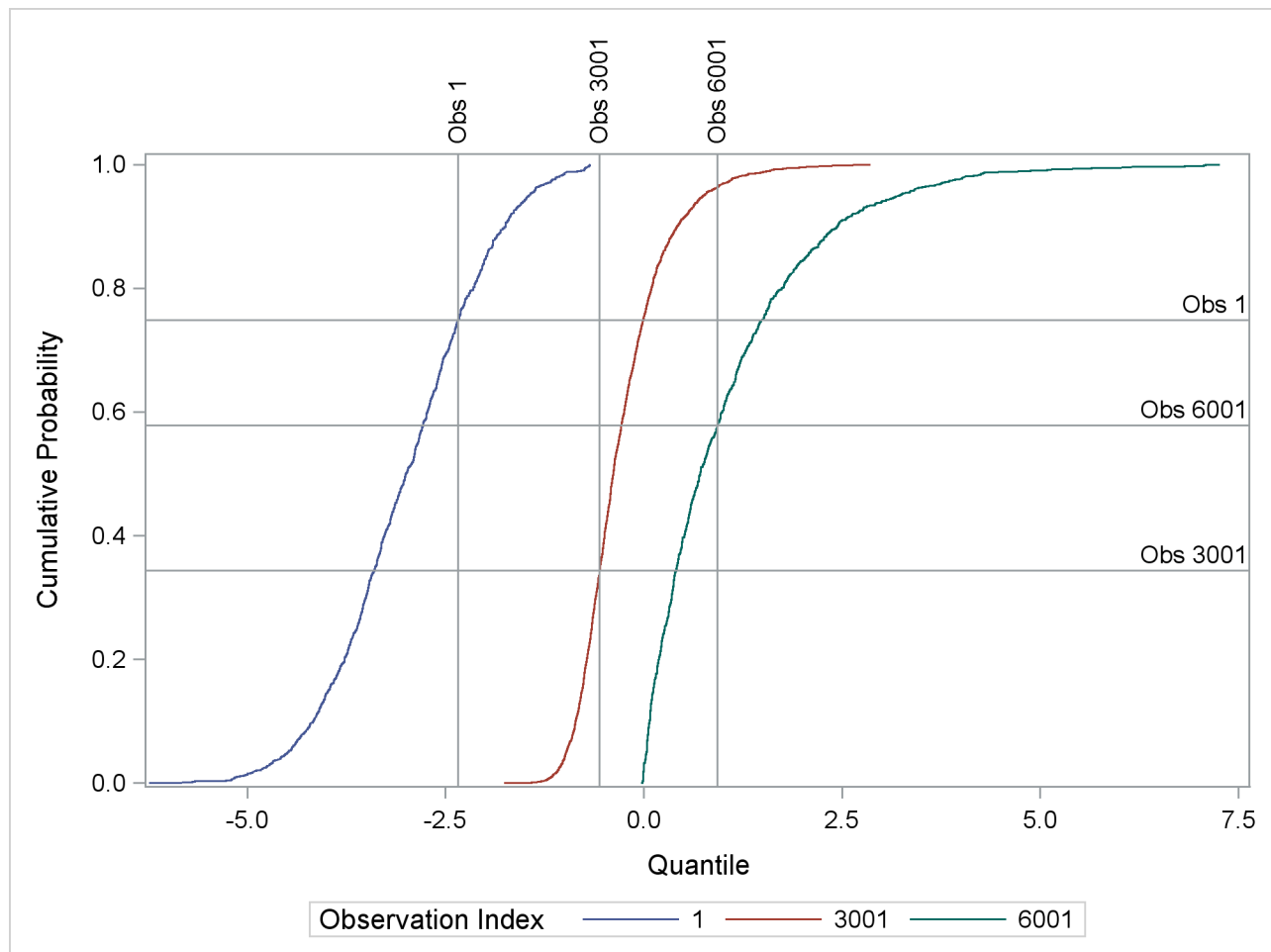
data distData;
set distData;
label iObs = "Observation Index"
      qLev = "Cumulative Probability"
      Quantiles = "Quantile";
run;

%macro plotCDF;
proc sgplot data=distData;
series y=qLev x=Quantiles/group=iObs;
%do j=1 %to &nObs;
refline &&qLev&j/label=("Obs &&_Obs&j")
axis=y labelloc=inside;
refline &&qt&j/label=("Obs &&_Obs&j")
axis=x;
%end;
run;
%mend;

%plotCDF;

```

Figure 96.5.5 displays the conditional CDFs for the three specified observations. Each observation has a vertical reference line for its observed response value and a horizontal reference line for its quantile level.

Output 96.5.5 Conditional Cumulative Distribution Function for Observation 1

You can also estimate the conditional probability density function (PDF) for the specified observations by using the KDE procedure. The following statements compute the probability estimates for each predicted quantile of the specified observations and plot their PDFs:

```
proc iml;
  use distData;
  read all var {"iObs"} into iObs;
  read all var {"qLev"} into qLev;
  read all var {"Quantiles"} into Y;
  close distData;

  nObs = nrow(iObs);
  nTau = nrow(qLev) / nObs;
  pProb = j(nTau, 3 + nObs);

  pProb[, 1] = t(1:nTau);
  pProb[, 2] = qLev[1:nTau];
  pProb[1, 3] = (qLev[1] + qLev[2]) / 2;
  pProb[nTau, 3] = 1 - (qLev[nTau - 1] + qLev[nTau]) / 2;
  pProb[2:(nTau - 1), 3] = (qLev[3:nTau] - qLev[1:(nTau - 2)]) / 2;
```



```

do j=1 to nObs;
    jump = (j-1)*nTau;
    pProb[, (3+j)] = Y[(jump+1):(jump+nTau)];
end;

create probData from pProb[colname={"obs" "quantLev" "pProb"
                                   "Obs1" "Obs3001" "Obs6001"}];

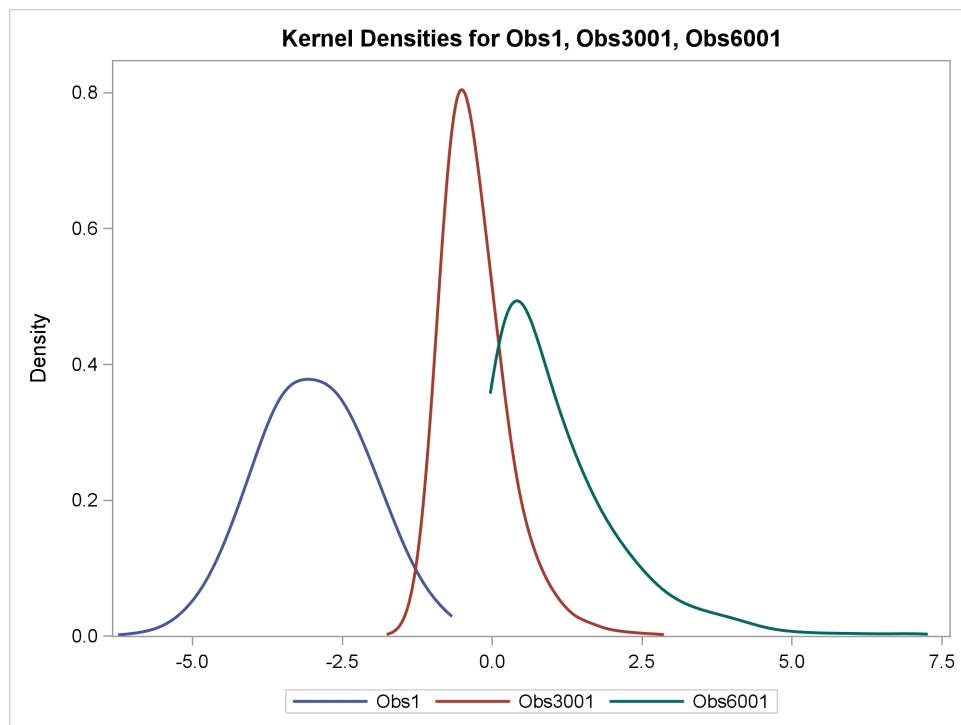
append from pProb;
close probData;
quit;

proc kde data=probData;
    weight pProb;
    univar Obs1 Obs3001 Obs6001/ plots=densityoverlay;
run;

```

Figure 96.15 displays the density estimation plots for the specified observations. You can see that the PDF for Observation 1 is a normal PDF, the PDF for Observation 3001 is slightly right-skewed, and the PDF for Observation 6001 is an exponential PDF.

Figure 96.15 Density Estimates



References

- Akaike, H. (1981). "Likelihood of a Model and Information Criteria." *Journal of Econometrics* 16:3–14.
- Barro, R., and Lee, J. W. (1994). "Data Set for a Panel of 138 Countries." Discussion paper, National Bureau of Econometric Research. <http://admin.nber.org/pub/barro.lee/readme.txt>.
- Belloni, A., and Chernozhukov, V. (2011). "L1-Penalized Quantile Regression in High-Dimensional Sparse Models." *Annals of Statistics* 39:82–130.
- Chernozhukov, V., and Hansen, C. (2008). "Instrumental Variable Quantile Regression: A Robust Inference Approach." *Journal of Econometrics* 142:379–398.
- Darlington, R. B. (1968). "Multiple Regression in Psychological Research and Practice." *Psychological Bulletin* 69:161–182.
- Gu, C., Bates, D. M., Chen, Z., and Wahba, G. (1990). "The Computation of GCV Function through Householder Tridiagonalization with Application to the Fitting of Interaction Splines Models." *SIAM Journal on Matrix Analysis and Applications* 10:457–480.
- 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.
- Judge, G. G., Griffiths, W. E., Hill, R. C., Lütkepohl, H., and Lee, T.-C. (1985). *The Theory and Practice of Econometrics*. 2nd ed. New York: John Wiley & Sons.
- Koenker, R., and Bassett, G. W. (1978). "Regression Quantiles." *Econometrica* 46:33–50.
- 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.
- McDonald, G. C., and Schwing, R. C. (1973). "Instabilities of Regression Estimates Relating Air Pollution to Mortality." *Technometrics* 15:463–481.
- 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.
- Tibshirani, R. (1996). "Regression Shrinkage and Selection via the Lasso." *Journal of the Royal Statistical Society, Series B* 58:267–288.
- Time Inc. (1987). "What They Make." *Sports Illustrated* (April 20): 54–81.
- Wu, Y., and Liu, Y. (2009). "Variable Selection in Quantile Regression." *Statistica Sinica* 19:801–817.
- Zou, H. (2006). "The Adaptive Lasso and Its Oracle Properties." *Journal of the American Statistical Association* 101:1418–1429.

Subject Index

- backward elimination
 - QUANTSELECT procedure, [7793](#)
- candidates for addition or removal
 - QUANTSELECT procedure, [7801](#)
- class level coding
 - QUANTSELECT procedure, [7800](#)
- class level information
 - QUANTSELECT procedure, [7800](#)
- dimension information
 - QUANTSELECT procedure, [7800](#)
- displayed output
 - QUANTSELECT procedure, [7800](#)
- effect selection
 - QUANTSELECT procedure, [7788](#), [7792](#)
- Extremal Quantile Levels
 - QUANTSELECT procedure, [7792](#)
- fit statistics
 - QUANTSELECT procedure, [7794](#), [7802](#)
- forward selection
 - QUANTSELECT procedure, [7793](#)
- GLMSELECT procedure
 - ODS Graphics, [7770](#)
- hierarchy
 - QUANTSELECT procedure, [7780](#)
- Information criteria
 - QUANTSELECT procedure, [7788](#)
- macro variables
 - QUANTSELECT procedure, [7796](#)
- model
 - hierarchy (QUANTSELECT), [7780](#)
- model information
 - QUANTSELECT procedure, [7800](#)
- number of observations
 - QUANTSELECT procedure, [7800](#)
- Observation Quantile Level
 - QUANTSELECT procedure, [7791](#)
- ODS graph names
 - QUANTSELECT procedure, [7804](#)
- ODS Graphics
 - GLMSELECT procedure, [7770](#)
- options summary
 - EFFECT statement, [7778](#)
- parameter estimates
 - QUANTSELECT procedure, [7802](#)
- parameter estimates for quantile process
 - QUANTSELECT procedure, [7803](#)
- Quantile Process Regression
 - QUANTSELECT procedure, [7791](#)
- quantile regression
 - QUANTSELECT procedure, [7788](#)
- QUANTSELECT procedure, [7794](#)
 - backward elimination, [7793](#)
 - candidates for addition or removal, [7801](#)
 - class level coding, [7800](#)
 - class level information, [7800](#)
 - dimension information, [7800](#)
 - displayed output, [7800](#)
 - effect selection, [7788](#), [7792](#)
 - Extremal Quantile Levels, [7792](#)
 - fit statistics, [7794](#), [7802](#)
 - forward selection, [7793](#)
 - hierarchy, [7780](#)
 - Information criteria, [7788](#)
 - macro variables, [7796](#)
 - model hierarchy, [7780](#)
 - model information, [7800](#)
 - number of observations, [7800](#)
 - Observation Quantile Level, [7791](#)
 - ODS graph names, [7804](#)
 - output table names, [7804](#)
 - parameter estimates, [7802](#)
 - parameter estimates for quantile process, [7803](#)
 - Quantile Process Regression, [7791](#)
 - quantile regression, [7788](#)
 - Quasi-Likelihood Ratio Tests, [7789](#)
 - selected effects, [7802](#)
 - selection reason, [7801](#)
 - selection summary, [7801](#)
 - stepwise selection, [7793](#)
 - stop reason, [7801](#)
 - test data, [7798](#)
 - validation data, [7798](#)
- Quasi-Likelihood Ratio Tests
 - QUANTSELECT procedure, [7789](#)
- selected effects
 - QUANTSELECT procedure, [7802](#)

selection reason

QUANTSELECT procedure, [7801](#)

selection summary

QUANTSELECT procedure, [7801](#)

stepwise selection

QUANTSELECT procedure, [7793](#)

stop reason

QUANTSELECT procedure, [7801](#)

test data

QUANTSELECT procedure, [7798](#)

validation data

QUANTSELECT procedure, [7798](#)

Syntax Index

- ACL
 - STATS= option (QUANTSELECT), [7785](#)
- ADAPTIVE option
 - MODEL statement (QUANTSELECT), [7782](#)
- ADJR1
 - STATS= option (QUANTSELECT), [7785](#)
- AIC
 - STATS= option (QUANTSELECT), [7785](#)
- AICC
 - STATS= option (QUANTSELECT), [7785](#)
- ALGORITHM option
 - PROC QUANTSELECT statement, [7766](#)
- BY statement
 - QUANTSELECT procedure, [7773](#)
- CHOOSE= option
 - MODEL statement (QUANTSELECT), [7782](#)
- CLASS statement
 - QUANTSELECT procedure, [7774](#)
- CODE statement
 - QUANTSELECT procedure, [7777](#)
- CPREFIX= option
 - CLASS statement (QUANTSELECT), [7774](#)
- DATA= option
 - PROC QUANTSELECT statement, [7766](#)
- DELIMITER option
 - CLASS statement (QUANTSELECT), [7774](#)
- DESCENDING option
 - CLASS statement (QUANTSELECT), [7775](#)
- DETAILS option
 - MODEL statement (QUANTSELECT), [7780](#)
- EFFECT statement
 - QUANTSELECT procedure, [7778](#)
- FITSTATISTICS
 - DETAILS=STEPS option (QUANTSELECT), [7780](#)
- HIERARCHY= option
 - MODEL statement (QUANTSELECT), [7780](#)
- INCLUDE= option
 - MODEL statement (QUANTSELECT), [7783](#)
- keyword option
 - OUTPUT statement (QUANTSELECT), [7786](#)
- LPREFIX= option
 - CLASS statement (QUANTSELECT), [7775](#)
- LR1
 - TEST= option (QUANTSELECT), [7785](#)
- LR2
 - TEST= option (QUANTSELECT), [7786](#)
- MAXMACRO= option
 - PROC QUANTSELECT statement, [7766](#)
- MAXSTEP option
 - MODEL statement (QUANTSELECT), [7783](#)
- MISSING option
 - CLASS statement (QUANTSELECT), [7775](#)
- MODEL statement
 - QUANTSELECT procedure, [7779](#)
- NAMELEN= option
 - PROC QUANTSELECT statement, [7768](#)
- NOINT option
 - MODEL statement (QUANTSELECT), [7781](#)
- NOPRINT option
 - PROC QUANTSELECT statement, [7768](#)
- ORDER= option
 - CLASS statement (QUANTSELECT), [7775](#)
- OUT= option
 - OUTPUT statement (QUANTSELECT), [7787](#)
- OUTDESIGN= option
 - PROC QUANTSELECT statement, [7768](#)
- OUTPUT statement
 - QUANTSELECT procedure, [7786](#)
- PARAM= option
 - CLASS statement (QUANTSELECT), [7775](#)
- PARAMETERESTIMATES
 - DETAILS=STEPS option (QUANTSELECT), [7780](#)
- PARTITION statement
 - QUANTSELECT procedure, [7787](#)
- PLOT option
 - PROC QUANTSELECT statement, [7770](#)
- PLOTS option
 - PROC QUANTSELECT statement, [7770](#)
- PREDICTED keyword
 - OUTPUT statement (QUANTSELECT), [7787](#)
- PROC QUANTSELECT statement, *see*
 - QUANTSELECT procedure
- QUANTILES option

MODEL statement (QUANTSELECT), 7781
 QUANTLEVEL keyword
 OUTPUT statement (QUANTSELECT), 7787
 QUANTSELECT procedure, 7765
 syntax, 7765
 QUANTSELECT procedure, BY statement, 7773
 QUANTSELECT procedure, CLASS statement, 7774
 CPREFIX= option, 7774
 DELIMITER option, 7774
 DESCENDING option, 7775
 LPREFIX= option, 7775
 MISSING option, 7775
 ORDER= option, 7775
 PARAM= option, 7775
 REF= v-option, 7776
 SHOWCODING option, 7774
 SPLIT option, 7776
 QUANTSELECT procedure, CODE statement, 7777
 QUANTSELECT procedure,
 DETAILS=STEPS(FITSTATISTICS) option
 FITSTATISTICS, 7780
 QUANTSELECT procedure, DE-
 TAILS=STEPS(PARAMETERESTIMATES)
 option
 PARAMETERESTIMATES, 7780
 QUANTSELECT procedure, EFFECT statement, 7778
 QUANTSELECT procedure, MODEL statement, 7779
 ADAPTIVE option, 7782
 CHOOSE= option, 7782
 DETAILS option, 7780
 HIERARCHY= option, 7780
 INCLUDE= option, 7783
 MAXSTEP option, 7783
 NOINT option, 7781
 NTAU= option, 7781
 QUANTILES= option, 7781
 SELECT= option, 7783
 SELECTION= option, 7781
 SLENTY= option, 7783
 SLSTAY= option, 7783
 STATS option, 7785
 STOP= option, 7783
 STOPHORIZON= option, 7785
 TEST option, 7785
 QUANTSELECT procedure, OUTPUT statement,
 7786
 keyword option, 7786
 OUT= option, 7787
 PREDICTED keyword, 7787
 QUANTLEVEL keyword, 7787
 RESIDUAL keyword, 7787
 QUANTSELECT procedure, PARTITION statement,
 7787
 FRACTION option, 7787

ROLEVAR= option, 7787
 QUANTSELECT procedure, PROC QUANTSELECT
 statement, 7765
 ALGORITHM option, 7766
 DATA= option, 7766
 MAXMACRO= option, 7766
 NAMELEN= option, 7768
 NOPRINT option, 7768
 OUTDESIGN= option, 7768
 PLOT option, 7770
 PLOTS option, 7770
 SEED= option, 7772
 TESTDATA= option, 7773
 VALDATA= option, 7773
 QUANTSELECT procedure, STATS= option
 ACL, 7785
 ADJR1, 7785
 AIC, 7785
 AICC, 7785
 R1, 7785
 SBC, 7785
 QUANTSELECT procedure, TEST= option
 LR1, 7785
 LR2, 7786
 QUANTSELECT procedure, WEIGHT statement,
 7788

 R1
 STATS= option (QUANTSELECT), 7785
 RANDOM option
 QUANTSELECT procedure, PARTITION
 statement, 7787
 REF= v-option
 CLASS statement (QUANTSELECT), 7776
 RESIDUAL keyword
 OUTPUT statement (QUANTSELECT), 7787
 ROLEVAR= option
 QUANTSELECT procedure, PARTITION
 statement, 7787

 SBC
 STATS= option (QUANTSELECT), 7785
 SEED= option
 PROC QUANTSELECT statement, 7772
 SELECT= option
 MODEL statement (QUANTSELECT), 7783
 SELECTION= option
 MODEL statement (QUANTSELECT), 7781
 SHOWCODING option
 CLASS statement (QUANTSELECT), 7774
 SLENTY= option
 MODEL statement (QUANTSELECT), 7783
 SLSTAY= option
 MODEL statement (QUANTSELECT), 7783

SPLIT option

 CLASS statement (QUANTSELECT), [7776](#)

STATS option

 MODEL statement (QUANTSELECT), [7785](#)

STOP= option

 MODEL statement (QUANTSELECT), [7783](#)

STOPHORIZON= option

 MODEL statement (QUANTSELECT), [7785](#)

TEST option

 MODEL statement (QUANTSELECT), [7785](#)

TESTDATA= option

 PROC QUANTSELECT statement, [7773](#)

VALDATA= option

 PROC QUANTSELECT statement, [7773](#)

WEIGHT statement

 QUANTSELECT procedure, [7788](#)