SAS/ETS® 14.1 User's Guide

The HPQLIM Procedure

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

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

SAS/ETS® 14.1 User's Guide

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

July 2015

Chapter 22
The HPQLIM Procedure

Contents

Overview: HPQLIM Procedure

The HPQLIM (high-performance qualitative and limited dependent variable model) procedure is a high-performance version of the QLIM procedure in SAS/ETS software, which analyzes univariate limited dependent variable models in which dependent variables are observed only in a limited range of values. Unlike the QLIM procedure, which can be run only on an individual workstation, the HPQLIM procedure takes advantage of a computing environment that enables it to distribute the optimization task to one or more nodes. In addition, each node can use one or more threads to perform the optimization on its subset of the data. When several nodes are used and each node uses several threads to carry out its part of the work, the result is a highly parallel computation that provides a dramatic gain in performance.

With the HPQLIM procedure you can read and write data in distributed form and perform analyses in distributed mode and single-machine mode. For more information about how to affect the execution mode of SAS high-performance analytical procedures, see the section "Processing Modes" on page 62 in Chapter 3, "Shared Concepts and Topics."

The HPQLIM procedure is specifically designed to operate in the high-performance distributed environment. It can use maximum likelihood or Bayesian methods. In both cases, the likelihood evaluation is performed in a distributed environment. By default, PROC HPQLIM uses multiple threads to perform computations.

The HPQLIM procedure is similar in use to the other SAS procedures that support regression or simultaneous equations models. For example, the standard model with censoring or truncation is estimated by specifying the endogenous variable to be truncated or censored. When the data are limited by specific values or variables, the limits of the dependent variable can be specified with the CENSORED or TRUNCATED option in the ENDOGENOUS or MODEL statement. For example, the two-limit censored model requires two variables: one that contains the lower (bottom) bound and one that contains the upper (top) bound. The following statements execute the model in the distributed computing environment with two threads and four nodes:

```
proc hpqlim data=a;
   model y = x1 x2 x3;
   endogenous y ~ censored(lb=bottom ub=top);
   performance nthreads=2 nodes=4 details;
run;
```

The bounds can be numbers if they are fixed for all observations in the data set. For example, the standard Tobit model can be specified as follows:

```
proc hpqlim data=a;
   model y = x1 x2 x3;
   endogenous y ~ censored(lb=0);
   performance nthreads=2 nodes=4 details;
run;
```

PROC HPQLIM Features

The HPQLIM procedure supports the following models:

- linear regression models with heteroscedasticity
- Tobit models (censored and truncated) with heteroscedasticity
- stochastic frontier production and cost models

In linear regression models with heteroscedasticity, the assumption that error variance is constant across observations is relaxed. The HPQLIM procedure allows for a number of different linear and nonlinear variance specifications.

The HPQLIM procedure also offers a class of models in which the dependent variable is censored or truncated from below or above or both. When a continuous dependent variable is observed only within a certain range, and values outside this range are not available, the HPQLIM procedure offers a class of models that adjust for truncation. In some cases, the dependent variable is continuous only in a certain range, and all values outside this range are reported as being on its boundary. For example, if it is not possible to observe negative values, the value of the dependent variable is reported as equal to 0. Because the data are censored, ordinary least squares (OLS) results are inconsistent, and it cannot be guaranteed that the predicted values from the model will fall in the appropriate region.

Stochastic frontier production and cost models allow for random shocks of the production or cost. They include a systematic positive component in the error term that adjusts for technical or cost inefficiency.

The HPQLIM procedure can use maximum likelihood or Bayesian methods. Initial starting values for the nonlinear optimizations are typically calculated by OLS. Initial values for the Bayesian sampling are typically calculated by maximum likelihood.

Getting Started: HPQLIM Procedure

This example illustrates the use of the HPQLIM procedure. The data were originally published by Mroz (1987), and the following statements show a subset of the Mroz (1987) data set:

```
title1 'Estimating a Tobit model';

data subset;
   input Hours Yrs_Ed Yrs_Exp @@;
   if Hours eq 0 then Lower=.;
      else               Lower=Hours;
datalines;
0 8 9 0 8 12 0 9 10 0 10 15 0 11 4 0 11 6
1000 12 1 1960 12 29 0 13 3 2100 13 36
3686 14 11 1920 14 38 0 15 14 1728 16 3
1568 16 19 1316 17 7 0 17 15
;
```

In these data, Hours is the number of hours that the wife worked outside the household in a given year, Yrs_Ed is the years of education, and Yrs_Exp is the years of work experience.

By the nature of the data it is clear that there are a number of women who committed some positive number of hours to outside work ($y_i > 0$ is observed). There are also a number of women who did not work outside the home at all ($y_i = 0$ is observed). This yields the following model:

$$y_i^* = \mathbf{x}_i' \boldsymbol{\beta} + \epsilon_i$$

$$y_i = \begin{cases} y_i^* & \text{if } y_i^* > 0 \\ 0 & \text{if } y_i^* \leq 0 \end{cases}$$

where $\epsilon_i \sim iidN(0, \sigma^2)$ and the set of explanatory variables is denoted by $\mathbf{x}_i$. The following statements fit a Tobit model to the hours worked with years of education and years of work experience as covariates:

```
/*-- Tobit Model --*/
proc hpqlim data=subset;
   model hours = yrs_ed yrs_exp;
   endogenous hours ~ censored(lb=0);
   performance nthreads=2 nodes=4 details;
run;
```

The output of the HPQLIM procedure is shown in Output 22.1.

Figure 22.1 Tobit Analysis Results

Estimating a Tobit model

The HPQLIM Procedure

Model Fit Summary	
Number of Endogenous Variables	1
Endogenous Variable	Hours
Number of Observations	17
Log Likelihood	-74.93700
Maximum Absolute Gradient	1.18953E-6
Number of Iterations	23
Optimization Method	Quasi-Newton
AIC	157.87400
Schwarz Criterion	161.20685

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
Intercept	1	-5598.295129	27.692220	-202.16	<.0001
Yrs_Ed	1	373.123254	53.988877	6.91	<.0001
Yrs_Exp	1	63.336247	36.551299	1.73	0.0831
_Sigma	1	1582.859635	390.076480	4.06	<.0001

The "Parameter Estimates" table contains four rows. The first three rows correspond to the vector estimate of the regression coefficients $\boldsymbol{\beta}$. The last row is called _Sigma, which corresponds to the estimate of the error variance σ.

Syntax: HPQLIM Procedure

The following statements are available in the HPQLIM procedure:

PROC HPQLIM *options* ;
  BAYES < *options* > ;
  BOUNDS *bound1* < , *bound2* . . . > ;
  BY *variables* ;
  FREQ *variable* ;
  ENDOGENOUS *variables* ∼ *options* ;
  HETERO *dependent variables* ∼ *exogenous variables* / *options* ;
  INIT *initvalue1* < , *initvalue2* . . . > ;
  MODEL *dependent variables* = *regressors* / *options* ;
  OUTPUT *options* ;
  PRIOR *variables* ∼ *distributions* ;
  RESTRICT *restriction1* < , *restriction2* . . . > ;
  TEST *options* ;
  WEIGHT *variable* ;

One MODEL statement is required. If a FREQ or WEIGHT statement is specified more than once, the variable that is specified in the first instance is used.

Functional Summary

Table 22.1 summarizes the statements and options used with the HPQLIM procedure.

Table 22.1 PROC HPQLIM Functional Summary

Description	Statement	Option
Data Set Options		
Specifies the input data set	PROC HPQLIM	DATA=
Writes parameter estimates to an output data set	PROC HPQLIM	OUTEST=
Writes predictions to an output data set	OUTPUT	OUT=
Declaring the Role of Variables		
Specifies BY-group processing	BY	
Specifies a frequency variable	FREQ	
Specifies a weight variable	WEIGHT	NONORMALIZE
Printing Control Options		
Requests all printing options	PROC HPQLIM	PRINTALL
Prints the correlation matrix of the estimates	PROC HPQLIM	CORRB
Prints the covariance matrix of the estimates	PROC HPQLIM	COVB
Suppresses the normal printed output	PROC HPQLIM	NOPRINT

Table 22.1 *continued*

Description	Statement	Option
Plotting Options		
Displays plots	PROC HPQLIM	PLOTS=
Optimization Process Control Options		
Selects the iterative minimization method to use	PROC HPQLIM	METHOD=
Specifies the maximum number of iterations allowed	PROC HPQLIM	MAXITER=
Specifies the maximum number of function calls	PROC HPQLIM	MAXFUNC=
Specifies the upper limit of CPU time in seconds	PROC HPQLIM	MAXTIME=
Specifies an absolute convergence criterion	PROC HPQLIM	ABSCONV=
Specifies an absolute function convergence criterion	PROC HPQLIM	ABSFCONV=
Specifies an absolute gradient convergence criterion	PROC HPQLIM	ABSGCONV=
Specifies a relative function convergence criterion	PROC HPQLIM	FCONV=
Specifies a relative gradient convergence criterion	PROC HPQLIM	GCONV=
Specifies an absolute parameter convergence criterion	PROC HPQLIM	ABSXCONV=
Specifies a matrix singularity criterion	PROC HPQLIM	SINGULAR=
Sets boundary restrictions on parameters	BOUNDS	
Sets initial values for parameters	INIT	
Sets linear restrictions on parameters	RESTRICT	
Model Estimation Options		
Suppresses the intercept parameter	MODEL	NOINT
Specifies the method to calculate parameter covariance	PROC HPQLIM	COVEST=
Bayesian MCMC Options		
Specifies the initial values of the MCMC	INIT	
Specifies the maximum number of tuning phases	BAYES	MAXTUNE=
Specifies the minimum number of tuning phases	BAYES	MINTUNE=
Specifies the number of burn-in iterations	BAYES	NBI=
Specifies the number of iterations during the sampling phase	BAYES	NMC=
Specifies the number of iterations during the tuning phase	BAYES	NTU=
Controls options for constructing the initial proposal covariance matrix	BAYES	PROPCOV
Specifies the sampling scheme	BAYES	SAMPLING=
Specifies the random number generator seed	BAYES	SEED=
Controls the thinning of the Markov chain	BAYES	THIN=
Bayesian Summary Statistics and Convergence Diagnostic Options		
Displays convergence diagnostics	BAYES	DIAGNOSTICS=
Displays summary statistics of the posterior samples	BAYES	STATISTICS=

Table 22.1 *continued*

Description	Statement	Option
Bayesian Prior and Posterior Sample Options		
Specifies a SAS data set for the posterior samples	BAYES	OUTPOST=
Bayesian Analysis Options		
Specifies the normal prior distribution	PRIOR	NORMAL(MEAN=, VAR=)
Specifies the gamma prior distribution	PRIOR	GAMMA(SHAPE=, SCALE=)
Specifies the inverse gamma prior distribution	PRIOR	IGAMMA(SHAPE=, SCALE=)
Specifies the uniform prior distribution	PRIOR	UNIFORM(MIN=, MAX=)
Specifies the beta prior distribution	PRIOR	BETA(SHAPE1=, SHAPE2=, MIN=, MAX=)
Specifies the *t* prior distribution	PRIOR	T(LOCATION=, DF=)
Endogenous Variable Options		
Specifies a discrete variable	ENDOGENOUS	DISCRETE()
Specifies a censored variable	ENDOGENOUS	CENSORED()
Specifies a truncated variable	ENDOGENOUS	TRUNCATED()
Specifies a stochastic frontier variable	ENDOGENOUS	FRONTIER()
Heteroscedasticity Model Options		
Specifies the function for heteroscedasticity models	HETERO	LINK=
Squares the function for heteroscedasticity models	HETERO	SQUARE
Specifies no constant for heteroscedasticity models	HETERO	NOCONST
Output Control Options		
Outputs predicted values	OUTPUT	PREDICTED
Outputs the structured part	OUTPUT	XBETA
Outputs residuals	OUTPUT	RESIDUAL
Outputs the error standard deviation	OUTPUT	ERRSTD
Outputs marginal effects	OUTPUT	MARGINAL
Outputs probability for the current response	OUTPUT	PROB
Outputs probability for all responses	OUTPUT	PROBALL
Outputs the expected value	OUTPUT	EXPECTED
Outputs the conditional expected value	OUTPUT	CONDITIONAL
Outputs inverse Mills ratio	OUTPUT	MILLS
Outputs technical efficiency measures	OUTPUT	TE1
	OUTPUT	TE2
Includes covariances in the OUTEST= data set	PROC HPQLIM	COVOUT
Includes correlations in the OUTEST= data set	PROC HPQLIM	CORROUT

Table 22.1 *continued*

Description	Statement	Option
Test Request Options		
Requests Wald, Lagrange multiplier, and likelihood ratio tests	TEST	ALL
Requests the Wald test	TEST	WALD
Requests the Lagrange multiplier test	TEST	LM
Requests the likelihood ratio test	TEST	LR

PROC HPQLIM Statement

PROC HPQLIM *options* **;**

The PROC HPQLIM statement invokes the HPQLIM procedure. You can specify the following *options*.

Data Set Options

DATA=*SAS-data-set*
specifies the input SAS data set. If this option is not specified, PROC HPQLIM uses the most recently created SAS data set.

Output Data Set Options

OUTEST=*SAS-data-set*
writes the parameter estimates to an output data set.

COVOUT
writes the covariance matrix for the parameter estimates to the OUTEST= data set. This option is valid only if the OUTEST= option is specified.

CORROUT
writes the correlation matrix for the parameter estimates to the OUTEST= data set. This option is valid only if the OUTEST= option is specified.

Printing Options

NOPRINT
suppresses the normal printed output but does not suppress error listings. If this option is specified, then any other print option is turned off.

PRINTALL
turns on all the printing options. The options that are set by PRINTALL are COVB and CORRB.

CORRB
prints the correlation matrix of the parameter estimates.

COVB
prints the covariance matrix of the parameter estimates.

Model Estimation Options

COVEST=*covariance-option*
specifies the method for calculating the covariance matrix of parameter estimates. You can specify the following *covariance-options*.

OP	specifies the covariance from the outer product matrix.
HESSIAN	specifies the covariance from the inverse Hessian matrix.
QML	specifies the covariance from the outer product and Hessian matrices (the quasi-maximum likelihood estimates).

The default is COVEST=HESSIAN.

Optimization Control Options

PROC HPQLIM uses the nonlinear optimization (NLO) subsystem to perform nonlinear optimization tasks. You can specify the following *options*:

ABSCONV=*r*
ABSTOL=*r*
specifies an absolute function value convergence criterion by which minimization stops when $f(\theta^{(k)}) \leq r$. The default value of r is the negative square root of the largest double-precision value, which serves only as a protection against overflows.

ABSFCONV=*r*
ABSFTOL=*r*
specifies an absolute function difference convergence criterion by which minimization stops when the function value has a small change in successive iterations:

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

The default value is $r = 0$.

ABSGCONV=*r*
ABSGTOL=*r*
specifies an absolute gradient convergence criterion. Optimization stops when the maximum absolute gradient element is small:

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

The default value is r=1E–5.

ABSXCONV=*r*

ABSXTOL=*r*

specifies an absolute parameter convergence criterion. Optimization stops when the Euclidean distance between successive parameter vectors is small:

$$\| \theta^{(k)} - \theta^{(k-1)} \|_2 \leq r$$

The default is 0.

FCONV=*r*

FTOL=*r*

specifies a relative function convergence criterion. Optimization stops when a relative change of the function value in successive iterations is small:

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

The default value is $r = 2\epsilon$, where ϵ denotes the machine precision constant, which is the smallest double-precision floating-point number such that $1 + \epsilon > 1$.

GCONV=*r*

GTOL=*r*

specifies a relative gradient convergence criterion. For all techniques except CONGRA, optimization stops when the normalized predicted function reduction is small:

$$\frac{g(\theta^{(k)})^T [H^{(k)}]^{-1} g(\theta^{(k)})}{|f(\theta^{(k)})|} \leq r$$

For the CONGRA technique (where a reliable Hessian estimate H is not available), the following criterion is used:

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

The default value is r =1E–8.

MAXFUNC=*i*

MAXFU=*i*

specifies the maximum number of function calls in the optimization process. The default is 1,000.

The optimization can terminate only after completing a full iteration. Therefore, the number of function calls that are actually performed can exceed the number of calls that are specified by this option.

MAXITER=*i*

MAXIT=*i*

specifies the maximum number of iterations in the optimization process. The default is 200.

MAXTIME=*r*

specifies an upper limit of *r* seconds of CPU time for the optimization process. The default value is the largest floating-point double representation of your computer. The time that is specified by this option is checked only once at the end of each iteration. Therefore, the actual running time can be much longer than *r*. The actual running time includes the remaining time needed to finish the iteration and the time needed to generate the output of the results.

METHOD=*value*

specifies the iterative minimization method to use. The default is METHOD=NEWRAP. You can specify the following *values*:

CONGRA	specifies the conjugate-gradient method.
DBLDOG	specifies the double dogleg method.
NONE	specifies that no optimization be performed beyond using the ordinary least squares method to compute the parameter estimates.
NEWRAP	specifies the Newton-Raphson method (the default).
NRRIDG	specifies the Newton-Raphson Ridge method.
QUANEW	specifies the quasi-Newton method.
TRUREG	specifies the trust region method.

SINGULAR=*r*

specifies the general singularity criterion that is applied by the HPQLIM procedure in sweeps and inversions. The default for the optimization is 1E–8.

Plotting Options

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

controls the display of plots. By default, the plots are displayed in panels unless the UNPACK *global-plot-option* is specified. When you specify only one *plot-request*, you can omit the parentheses around it.

Global Plot Options

You can specify the following *global-plot-options*:

ONLY

displays only the requested plot.

UNPACKPANEL

UNPACK

specifies that all paneled plots be unpacked, meaning that each plot in a panel is displayed separately.

Plot Requests

You can specify the following *plot-requests*:

ALL

specifies all types of available plots.

AUTOCORR< (LAGS=*n***) >**

displays the autocorrelation function plots for the parameters. The optional LAGS= suboption specifies the number (up to lag *n*) of autocorrelations to be plotted in the autocorrelation function plot. If this suboption is not specified, autocorrelations are plotted up to lag 50. This *plot-request* is available only for Bayesian analysis.

BAYESDIAG
is equivalent to specifying the TRACE, AUTOCORR, and DENSITY *plot-requests*.

DENSITY< (FRINGE) >
displays the kernel density plots for the parameters. If you specify the FRINGE suboption, a fringe plot is created on the X axis of the kernel density plot. This *plot-request* is available only for Bayesian analysis.

NONE
suppresses all diagnostic plots.

TRACE< (SMOOTH) >
displays the trace plots for the parameters. The SMOOTH suboption displays a fitted penalized B-spline curve for each plot. This *plot-request* is available only for Bayesian analysis.

BAYES Statement

BAYES < *options* > **;**

The BAYES statement controls the Metropolis sampling scheme that is used to obtain samples from the posterior distribution of the underlying model and data.

DIAGNOSTICS=ALL | NONE | *(keyword-list)*
DIAG=ALL | NONE | *(keyword-list)*
controls which diagnostics are produced. All the following diagnostics are produced when you specify DIAGNOSTICS=ALL. If you do not want any of these diagnostics, specify DIAGNOSTICS=NONE. If you want some but not all of the diagnostics, or if you want to change certain settings of these diagnostics, specify one or more of the following keywords. The default is DIAGNOSTICS=NONE.

AUTOCORR < (LAGS=*numeric-list***) >**
computes the autocorrelations at lags that are specified in the *numeric-list*. Elements in the *numeric-list* are truncated to integers, and repeated values are removed. If the LAGS= option is not specified, autocorrelations of lags 1, 5, and 10 are computed.

ESS
computes Carlin's estimate of the effective sample size, the correlation time, and the efficiency of the chain for each parameter.

GEWEKE < (*geweke-options***) >**
computes the Geweke spectral density diagnostics, which are essentially a two-sample t test between the first f_1 portion and the last f_2 portion of the chain. The defaults are $f_1 = 0.1$ and $f_2 = 0.5$, but you can choose other fractions by using the following *geweke-options*:

FRAC1=*value*
specifies the fraction f_1 for the first window.

FRAC2=*value*
specifies the fraction f_2 for the second window.

HEIDELBERGER < (*heidel-options*) >

computes for each variable the Heidelberger and Welch diagnostic, which consists of a stationarity test of the null hypothesis that the sample values form a stationary process. If the stationarity test is not rejected, a halfwidth test is then carried out. Optionally, you can specify one or more of the following *heidel-options*:

EPS=*value*

specifies a positive number ϵ such that if the halfwidth is less than ϵ times the sample mean of the retained iterates, the halfwidth test is passed.

HALPHA=*value*

specifies the α level ($0 < \alpha < 1$) for the halfwidth test.

SALPHA=*value*

specifies the α level ($0 < \alpha < 1$) for the stationarity test.

MCSE

MCERROR

computes the Monte Carlo standard error for each parameter. The Monte Carlo standard error, which measures the simulation accuracy, is the standard error of the posterior mean estimate and is calculated as the posterior standard deviation divided by the square root of the effective sample size.

RAFTERY< (*raftery-options*) >

computes the Raftery and Lewis diagnostics, which evaluate the accuracy of the estimated quantile ($\hat{\theta}_Q$ for a given $Q \in (0,1)$) of a chain. $\hat{\theta}_Q$ can achieve any degree of accuracy when the chain is allowed to run for a long time. The computation stops when the estimated probability $\hat{P}_Q = \Pr(\theta \leq \hat{\theta}_Q)$ reaches within $\pm R$ of the value Q with probability S; that is, $\Pr(Q - R \leq \hat{P}_Q \leq Q + R) = S$. The following *raftery-options* enable you to specify Q, R, S, and a precision level ϵ for the test:

QUANTILE | Q=*value*

specifies the order (a value between 0 and 1) of the quantile of interest. The default is 0.025.

ACCURACY | R=*value*

specifies a small positive number as the margin of error for measuring the accuracy of the estimation of the quantile. The default is 0.005.

PROBABILITY | S=*value*

specifies the probability of attaining the accuracy of the estimation of the quantile. The default is 0.95.

EPSILON | EPS=*value*

specifies the tolerance level (a small positive number) for the stationary test. The default is 0.001.

MINTUNE=*number*

specifies the minimum number of tuning phases. The default is 2.

MAXTUNE=*number*
specifies the maximum number of tuning phases. The default is 24.

NBI=*number*
specifies the number of burn-in iterations before the chains are saved. The default is 1,000.

NMC=*number*
specifies the number of iterations after the burn-in. The default is 1,000.

NTU=*number*
specifies the number of samples for each tuning phase. The default is 500.

OUTPOST=*SAS-data-set*
names the SAS data set to contain the posterior samples. Alternatively, you can create the output data set by specifying an ODS OUTPUT statement as follows:

ODS OUTPUT POSTERIORSAMPLE = < *SAS-data-set* > **;**

PROPCOV=*value*
specifies the method that is used in constructing the initial covariance matrix for the Metropolis-Hastings algorithm. The QUANEW and NMSIMP methods find numerically approximated covariance matrices at the optimum of the posterior density function with respect to all continuous parameters. The tuning phase starts at the optimized values; in some problems, this can greatly increase convergence performance. If the approximated covariance matrix is not positive definite, then an identity matrix is used instead. You can specify the following *values*:

CONGRA
performs a conjugate-gradient optimization.

DBLDOG
performs a version of double-dogleg optimization.

NEWRAP
performs a Newton-Raphson optimization that combines a line-search algorithm with ridging.

NMSIMP
performs a Nelder-Mead simplex optimization.

NRRIDG
performs a Newton-Raphson optimization with ridging.

QUANEW
performs a quasi-Newton optimization.

TRUREG
performs a trust-region optimization.

SAMPLING=MULTIMETROPOLIS | UNIMETROPOLIS
specifies how to sample from the posterior distribution. SAMPLING=MULTIMETROPOLIS implements a Metropolis sampling scheme on a single block that contains all the parameters of the model. SAMPLING=UNIMETROPOLIS implements a Metropolis sampling scheme on multiple blocks, one for each parameter of the model. The default is SAMPLING=MULTIMETROPOLIS.

SEED=*number*

specifies an integer seed in the range 1 to $2^{31} - 1$ for the random number generator in the simulation. Specifying a seed enables you to reproduce identical Markov chains for the same specification. If you do not specify the SEED= option, or if you specify a nonpositive seed, a random seed is derived from the time of day.

STATISTICS < (*global-options*) > = **ALL** | **NONE** | *keyword* | *(keyword-list)*

STATS < (*global-options*) > = **ALL** | **NONE** | *keyword* | *(keyword-list)*

controls the number of posterior statistics that are produced. Specifying STATISTICS=ALL is equivalent to specifying STATISTICS=(CORR COV INTERVAL PRIOR SUMMARY). If you do not want any posterior statistics, specify STATISTICS=NONE. The default is STATISTICS=(SUMMARY INTERVAL). You can specify the following *global-options*:

ALPHA=*value* < ,*value* >**...** < ,*value* >

controls the probabilities of the credible intervals. The *value*, which must be between 0 and 1, produces a pair of 100(1–*value*)% equal-tail and highest posterior density (HPD) intervals for each parameter. The default is ALPHA=0.05, which yields the 95% credible intervals for each parameter.

PERCENT=*value* < ,*value* >**...** < ,*value* >

requests the percentile points of the posterior samples. The *value* must be between 0 and 100. The default is PERCENT=25, 50, 75, which yields the 25th, 50th, and 75th percentile points, respectively, for each parameter.

You can specify the following *keywords*:

CORR

produces the posterior correlation matrix.

COV

produces the posterior covariance matrix.

INTERVAL

produces equal-tail credible intervals and HPD intervals. The default is to produce the 95% equal-tail credible intervals and 95% HPD intervals, but you can use the ALPHA= *global-option* to request intervals of any probabilities.

NONE

suppresses printing of all summary statistics.

PRIOR

produces a summary table of the prior distributions that are used in the Bayesian analysis.

SUMMARY

produces the means, standard deviations, and percentile points (25th, 50th, and 75th) for the posterior samples. You can use the PERCENT= *global-option* to request specific percentile points.

THIN=*number*

THINNING=*number*

controls the thinning of the Markov chain. Only one in every k samples is used when THIN=k. If NBI=n_0 and NMC=n, the number of samples that are retained is

$$\left[\frac{n_0 + n}{k}\right] - \left[\frac{n_0}{k}\right]$$

where $[a]$ represents the integer part of the number a. The default is THIN=1.

BOUNDS Statement

BOUNDS *bound1* < , *bound2* . . . > **;**

The BOUNDS statement imposes simple boundary constraints on the parameter estimates. BOUNDS statement constraints refer to the parameters that are estimated by the HPQLIM procedure. You can specify any number of BOUNDS statements.

Each *bound* is composed of parameters, constants, and inequality operators. Parameters that are associated with regressor variables are referred to by the names of the corresponding regressor variables. Specify each bound as follows:

item operator item < *operator item* < *operator item* . . . > >

Each *item* is a constant, the name of a parameter, or a list of parameter names. For more information about how parameters are named in the HPQLIM procedure, see the section "Naming of Parameters" on page 1182. Each *operator* is <, >, <=, or >=.

You can use both the BOUNDS statement and the RESTRICT statement to impose boundary constraints; however, the BOUNDS statement provides a simpler syntax for specifying these types of constraints. For more information, see the section "RESTRICT Statement" on page 1167.

The following BOUNDS statement constrains the estimates of the parameters that are associated with the variable ttime and the variables x1 through x10 to be between 0 and 1. The following example illustrates the use of parameter lists to specify boundary constraints.

```
bounds 0 < ttime x1-x10 < 1;
```

The following BOUNDS statement constrains the estimates of the correlation (_RHO) and sigma (_SIGMA) in the bivariate model:

```
bounds _rho >= 0, _sigma.y1 > 1, _sigma.y2 < 5;
```

BY Statement

BY *variables* **;**

A BY statement can be used with PROC HPQLIM to obtain separate analyses on observations in groups defined by the BY variables.

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

ENDOGENOUS Statement

ENDOGENOUS *variables* ∼ *options* **;**

The ENDOGENOUS statement specifies the type of dependent variables that appear on the left-hand side of the equation. The listed endogenous variables refer to the dependent variables that appear on the left-hand side of the equation. Currently, no right-hand-side endogeneity is handled in PROC HPQLIM. All variables that appear on the right-hand side of the equation are treated as exogenous.

Discrete Variable Options

DISCRETE < (*discrete-options* **) >**
specifies that the endogenous variables in this statement be discrete. You can specify the following *discrete-options*:

DISTRIBUTION=*distribution-type*
DIST=*distribution-type*
D=*distribution-type*
specifies the cumulative distribution function that is used to model the response probabilities. You can specify the following *distribution-types*:

LOGISTIC specifies the logistic distribution for the logit model.

NORMAL specifies the normal distribution for the probit model.

By default, DISTRIBUTION=NORMAL.

ORDER=DATA | FORMATTED | FREQ | INTERNAL
specifies the sort order for the levels of the discrete variables that are specified in the ENDOGENOUS statement. This ordering determines which parameters in the model correspond to each level in the data. You can specify the following sort orders:

DATA sorts levels by order of appearance in the input data set.

FORMATTED sorts levels by formatted value. The sort order is machine-dependent.

FREQ sorts levels by descending frequency count; levels that have the most observations come first in the order.

INTERNAL sorts levels by unformatted value. The sort order is machine-dependent.

By default, ORDER=FORMATTED. For more information about sort order, see the chapter on the SORT procedure in the *Base SAS Procedures Guide*.

Censored Variable Options

CENSORED (*censored-options* **)**
specifies that the endogenous variables in this statement be censored. You can specify the following *censored-options*:

LB=*value* | *variable*

LOWERBOUND=*value* | *variable*
specifies the lower bound of the censored variables. If *value* is missing or the value in *variable* is missing, no lower bound is set. By default, no lower bound is set.

UB=*value* | *variable*

UPPERBOUND=*value* | *variable*
specifies the upper bound of the censored variables. If *value* is missing or the value in *variable* is missing, no upper bound is set. By default, no upper bound is set.

Truncated Variable Options

TRUNCATED (*truncated-options* **)**
You can specify the following *truncated-options*:

LB=*value* | *variable*

LOWERBOUND=*value* | *variable*
specifies the lower bound of the truncated variables. If *value* is missing or the value in *variable* is missing, no lower bound is set. By default, no lower bound is set.

UB=*value* | *variable*

UPPERBOUND=*value* | *variable*
specifies the upper bound of the truncated variables. If *value* is missing or the value in *variable* is missing, no upper bound is set. By default, no upper bound is set.

Stochastic Frontier Variable Options

FRONTIER < (*frontier-options* **) >**
You can specify the following *frontier-options*:

TYPE=HALF | EXPONENTIAL | TRUNCATED
specifies the model type.

HALF
specifies half-normal model.

EXPONENTIAL
specifies exponential model.

TRUNCATED
specifies truncated normal model.

PRODUCTION
specifies that the estimated model be a production function.

COST
specifies that the estimated model be a cost function.

If neither PRODUCTION nor COST is specified, a production function is estimated by default.

FREQ Statement

FREQ *variable* **;**

The FREQ statement identifies a variable that contains the frequency of occurrence of each observation. PROC HPQLIM treats each observation as if it appeared *n* times, where is the value of the FREQ variable for the observation. If the frequency value is not an integer, it is truncated to an integer. If the frequency value is less than 1 or missing, the observation is not used in the model fitting. When the FREQ statement is not specified, each observation is assigned a frequency of 1. If you specify more than one FREQ statement, then the first FREQ statement is used.

HETERO Statement

HETERO *dependent variables* ∼ *exogenous variables* < / *options* > **;**

The HETERO statement specifies variables that are related to the heteroscedasticity of the residuals and the way that these variables are used to model the error variance. PROC HPQLIM supports the following heteroscedastic regression model:

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

$$\epsilon_i \sim \mathrm{N}(0, \sigma_i^2)$$

For more information about the specification of functional forms, see the section "Heteroscedasticity" on page 1172. The following *options* specify the functional forms of heteroscedasticity:

LINK=EXP | LINEAR
specifies the functional form.

EXP
specifies the exponential link function:

$$\sigma_i^2 = \sigma^2(1 + \exp(\mathbf{z}_i' \boldsymbol{\gamma}))$$

LINEAR

specifies the linear link function:

$$\sigma_i^2 = \sigma^2(1 + \mathbf{z}_i^{'}\boldsymbol{\gamma})$$

The default is LINK=EXP.

NOCONST

specifies that there be no constant in the linear or exponential heteroscedasticity model:

$$\sigma_i^2 = \sigma^2(\mathbf{z}_i^{'}\boldsymbol{\gamma})$$
$$\sigma_i^2 = \sigma^2\exp(\mathbf{z}_i^{'}\boldsymbol{\gamma})$$

This option is ignored if you do not specify the LINK= option.

SQUARE

estimates the model by using the square of the linear heteroscedasticity function. For example, you can specify the following heteroscedasticity function:

$$\sigma_i^2 = \sigma^2(1 + (\mathbf{z}_i^{'}\boldsymbol{\gamma})^2)$$

```
model y = x1 x2 / censored(lb=0);
hetero y ~ z1 / link=linear square;
```

The SQUARE option does not apply to the exponential heteroscedasticity function because the square of an exponential function of $\mathbf{z}_i^{'}\boldsymbol{\gamma}$ is the same as the exponential of $2\mathbf{z}_i^{'}\boldsymbol{\gamma}$. Hence, the only difference is that all $\boldsymbol{\gamma}$ estimates are divided by two.

This option is ignored if you do not specify the LINK= option. You cannot use the HETERO statement within a Bayesian framework.

INIT Statement

INIT *initvalue1* < , *initvalue2* . . . > **;**

The INIT statement sets initial values for parameters in the optimization. You can specify any number of INIT statements.

Each *initvalue* is written as a parameter or parameter list, followed by an optional equality operator (=), followed by a number:

parameter **<=>** *number*

MODEL Statement

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

The MODEL statement specifies the dependent variable and independent regressor variables for the regression model.

You can specify the following *option* after a slash (/).

NOINT
suppresses the intercept parameter.

You can also specify the following endogenous variable options, which are the same as the options that are specified in the ENDOGENOUS statement. If an endogenous variable option is specified in both the MODEL statement and the ENDOGENOUS statement, the option in the ENDOGENOUS statement is used.

Discrete Variable Options

DISCRETE < (*discrete-options*) >
specifies that the endogenous variables in this statement be discrete. You can specify the following *discrete-options*:

DISTRIBUTION=*distribution-type*
DIST=*distribution-type*
D=*distribution-type*
specifies the cumulative distribution function that is used to model the response probabilities. You can specify the following *distribution-types*:

LOGISTIC specifies the logistic distribution for the logit model.

NORMAL specifies the normal distribution for the probit model.

By default, DISTRIBUTION=NORMAL.

ORDER=DATA | FORMATTED | FREQ | INTERNAL
specifies the sort order for the levels of the discrete variables that are specified in the ENDOGENOUS statement. This ordering determines which parameters in the model correspond to each level in the data. You can specify the following sort orders:

DATA sorts levels by order of appearance in the input data set.

FORMATTED sorts levels by formatted value. The sort order is machine-dependent.

FREQ sorts levels by descending frequency count; levels that have the most observations come first in the order.

INTERNAL sorts levels by unformatted value. The sort order is machine-dependent.

By default, ORDER=FORMATTED. For more information about sort order, see the chapter on the SORT procedure in the *Base SAS Procedures Guide*.

Censored Variable Options

CENSORED < (*censored-options*) >
specifies that the endogenous variables in this statement be censored. You can specify the following *censored-options*:

LB=*value* | *variable*

LOWERBOUND=*value* | *variable*
specifies the lower bound of the censored variables. If *value* is missing or the value in *variable* is missing, no lower bound is set. By default, no lower bound is set.

UB=*value* | *variable*

UPPERBOUND=*value* | *variable*
specifies the upper bound of the censored variables. If *value* is missing or the value in *variable* is missing, no upper bound is set. By default, no upper bound is set.

Truncated Variable Options

TRUNCATED < (*truncated-options* **) >**
You can specify the following *truncated-options*:

LB=*value* | *variable*

LOWERBOUND=*value* | *variable*
specifies the lower bound of the truncated variables. If *value* is missing or the value in *variable* is missing, no lower bound is set. By default, no lower bound is set.

UB=*value* | *variable*

UPPERBOUND=*value* | *variable*
specifies the upper bound of the truncated variables. If *value* is missing or the value in *variable* is missing, no upper bound is set. By default, no upper bound is set.

Stochastic Frontier Variable Options

FRONTIER < (*frontier-options* **) >**
You can specify the following *frontier-options*:

TYPE=HALF | EXPONENTIAL | TRUNCATED
specifies the model type.

HALF
specifies a half-normal model.

EXPONENTIAL
specifies an exponential model.

TRUNCATED
specifies a truncated normal model.

PRODUCTION
specifies that the estimated model be a production function.

COST
specifies that the estimated model be a cost function.

If neither PRODUCTION nor COST is specified, a production function is estimated by default.

OUTPUT Statement

OUTPUT OUT=*SAS-data-set* < *output-options* > **;**

The OUTPUT statement creates a new SAS data set to contain variables that are specified with the COPYVAR option and the following data if they are specified by *output-options*: estimates of $\mathbf{x}'\boldsymbol{\beta}$, predicted value, residual, marginal effects, probability, standard deviation of the error, expected value, conditional expected value, technical efficiency measures, and inverse Mills ratio. When the response values are missing for the observation, all output estimates except the residual are still computed as long as none of the explanatory variables are missing. This enables you to compute these statistics for prediction. You can specify only one OUTPUT statement.

You must specify the OUT= option:

OUT=*SAS-data-set*
names the output data set.

COPYVAR=*SAS-variable-names*

COPYVARS=(*SAS-variable-names***)**
adds SAS variables to the output data set

You can specify one or more of the following *output-options*:

CONDITIONAL
outputs estimates of conditional expected values of continuous endogenous variables.

ERRSTD
outputs estimates of σ_j, the standard deviation of the error term.

EXPECTED
outputs estimates of expected values of continuous endogenous variables.

MARGINAL
outputs marginal effects.

MILLS
outputs estimates of inverse Mills ratios of censored or truncated continuous, binary discrete, and selection endogenous variables.

PREDICTED
outputs estimates of predicted endogenous variables.

PROB
outputs estimates of probability of discrete endogenous variables taking the current observed responses.

PROBALL
outputs estimates of probability of discrete endogenous variables for all possible responses.

RESIDUAL
outputs estimates of residuals of continuous endogenous variables.

XBETA
outputs estimates of $\mathbf{x}'\boldsymbol{\beta}$.

TE1
outputs estimates of technical efficiency for each producer in the stochastic frontier model that is suggested by Battese and Coelli (1988).

TE2
outputs estimates of technical efficiency for each producer in the stochastic frontier model that is suggested by Jondrow et al. (1982).

PERFORMANCE Statement

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

The PERFORMANCE statement specifies *performance-options* to control the multithreaded and distributed computing environment and requests detailed performance results of the HPQLIM procedure. You can also use the PERFORMANCE statement to control whether the HPQLIM procedure executes in single-machine or distributed mode. You can specify the following *performance-options*:

DETAILS
requests a table that shows a timing breakdown of the procedure steps.

NODES=*n*
specifies the number of nodes in the distributed computing environment, provided that the data are not processed alongside the database.

NTHREADS=*n*
specifies the number of threads for analytic computations and overrides the SAS system option THREADS | NOTHREADS. If you do not specify the NTHREADS= option, PROC HPQLIM creates one thread per CPU for the analytic computations.

The PERFORMANCE statement is documented further in the section "PERFORMANCE Statement" on page 87 in Chapter 3, "Shared Concepts and Topics."

PRIOR Statement

PRIOR _REGRESSORS | *parameter-list* ~ *distribution* **;**

The PRIOR statement specifies the prior distribution of the model parameters. You must specify one parameter or a list of parameters, a tilde ~, and then a distribution with its parameters. Multiple PRIOR statements are allowed.

You can specify the following *distributions*:

NORMAL(MEAN=μ**, VAR=**σ^2**)**
specifies a normal distribution with the parameters MEAN and VAR.

GAMMA(SHAPE=*a*, SCALE=*b*)
specifies a gamma distribution with the parameters SHAPE and SCALE.

IGAMMA(SHAPE=*a*, SCALE=*b*)
specifies an inverse gamma distribution with the parameters SHAPE and SCALE.

UNIFORM(MIN=*m*, MAX=*M*)
specifies a uniform distribution that is defined between MIN and MAX.

BETA(SHAPE1=*a*, SHAPE2=*b*, MIN=*m*, MAX=*M*)
specifies a beta distribution with the parameters SHAPE1 and SHAPE2 and defined between MIN and MAX.

T(LOCATION=μ, DF=ν)
specifies a noncentral t distribution with DF degrees of freedom and a location parameter equal to LOCATION.

For more information about how to specify *distributions*, see the section "Standard Distributions" on page 1175.

You can specify the special keyword REGRESSORS to select all the parameters that are used in the linear regression component of the model.

RESTRICT Statement

RESTRICT *restriction1* < , *restriction2* . . . > **;**

The RESTRICT statement imposes linear restrictions on the parameter estimates. You can specify any number of RESTRICT statements, but the number of restrictions that are imposed is limited by the number of regressors.

Each *restriction* is written as an expression, followed by an equality operator (=) or an inequality operator (<, >, <=, >=), followed by a second expression:

expression operator expression

The *operator* can be =, <, >, <= , or >=. The *operator* and second *expression* are optional.

Restriction expressions can be composed of parameter names; multiplication ($*$), addition ($+$), and substitution ($-$) operators; and constants. Parameters that are named in restriction expressions must be among the parameters that are estimated by the model. Parameters that are associated with a regressor variable are referred to by the name of the corresponding regressor variable. The restriction expressions must be a linear function of the parameters.

The following statements illustrate the use of the RESTRICT statement:

```
proc hpqlim data=one;
   model y = x1-x10 / censored(lb=0);
   restrict x1*2 <= x2 + x3;
run;
```

TEST Statement

<'label':> **TEST** *<'string':> equation < ,equation. . . > / options* **;**

The TEST statement performs Wald, Lagrange multiplier, and likelihood ratio tests of linear hypotheses about the regression parameters in the preceding MODEL statement. Each equation specifies a linear hypothesis to be tested. All hypotheses in one TEST statement are tested jointly. Variable names in the equations must correspond to regressors in the preceding MODEL statement, and each name represents the coefficient of the corresponding regressor. Use the keyword INTERCEPT for a test that includes a constant.

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

ALL
requests Wald, Lagrange multiplier, and likelihood ratio tests.

LM
requests the Lagrange multiplier test.

LR
requests the likelihood ratio test.

WALD
requests the Wald test.

The following statements illustrate the use of the TEST statement (note the use of the INTERCEPT keyword in the second TEST statement):

```
proc hpqlim;
   model y = x1 x2 x3;
   test x1 = 0, x2 * .5 + 2 * x3 = 0;
   test _int: test intercept = 0, x3 = 0;
run;
```

The first TEST statement investigates the joint hypothesis that

$$\beta_1 = 0$$

and

$$0.5\beta_2 + 2\beta_3 = 0$$

Only linear equality restrictions and tests are permitted in PROC HPQLIM. Test expressions can be composed only of algebraic operations that involve the addition symbol (+), subtraction symbol (–), and multiplication symbol (*).

The TEST statement accepts labels that are reproduced in the printed output. You can label a TEST statement in two ways: you can specify a label followed by a colon before the TEST keyword, or you can specify a quoted string after the TEST keyword. If you specify both a label before the TEST keyword and a quoted string after the keyword, PROC HPQLIM uses the label that precedes the colon. If no label or quoted string is specified, PROC HPQLIM labels the test automatically.

WEIGHT Statement

WEIGHT *variable* < / *option* > **;**

The WEIGHT statement specifies a variable that supplies weighting values to use for each observation in estimating parameters. The log likelihood for each observation is multiplied by the corresponding weight variable value.

If the weight of an observation is nonpositive, that observation is not used in the estimation.

You can add the following *option* after a slash (/):

NONORMALIZE
specifies that the weights must be used as is. When this option is not specified, the weights are normalized so that they add up to the actual sample size. Weights w_i are normalized by multiplying them by $\frac{n}{\sum_{i=1}^{n} w_i}$, where n is the sample size.

Details: HPQLIM Procedure

Ordinal Discrete Choice Modeling

Binary Probit and Logit Model

The binary choice model is

$$y_i^* = \mathbf{x}_i'\boldsymbol{\beta} + \epsilon_i$$

where the value of the latent dependent variable, y_i^*, is observed only as follows:

$$\begin{aligned} y_i &= 1 \quad \text{if } y_i^* > 0 \\ &= 0 \quad \text{otherwise} \end{aligned}$$

The disturbance, ϵ_i, of the probit model has a standard normal distribution with the distribution function (CDF)

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \exp(-t^2/2)dt$$

The disturbance of the logit model has a standard logistic distribution with the distribution function (CDF)

$$\Lambda(x) = \frac{\exp(x)}{1+\exp(x)} = \frac{1}{1+\exp(-x)}$$

The binary discrete choice model has the following probability that the event $\{y_i = 1\}$ occurs:

$$P(y_i = 1) = F(\mathbf{x}_i'\boldsymbol{\beta}) = \begin{cases} \Phi(\mathbf{x}_i'\boldsymbol{\beta}) & \text{(probit)} \\ \Lambda(\mathbf{x}_i'\boldsymbol{\beta}) & \text{(logit)} \end{cases}$$

For more information, see the section "Ordinal Discrete Choice Modeling" on page 1985.

Ordinal Probit/Logit

When the dependent variable is observed in sequence with M categories, binary discrete choice modeling is not appropriate for data analysis. McKelvey and Zavoina (1975) propose the ordinal (or ordered) probit model.

Consider the regression equation

$$y_i^* = \mathbf{x}_i'\boldsymbol{\beta} + \epsilon_i$$

where error disturbances, ϵ_i, have the distribution function F. The unobserved continuous random variable, y_i^*, is identified as M categories. Suppose there are $M+1$ real numbers, $\mu_0, \ldots, \mu_M$, where $\mu_0 = -\infty$, $\mu_1 = 0$, $\mu_M = \infty$, and $\mu_0 \le \mu_1 \le \cdots \le \mu_M$. Define

$$R_{i,j} = \mu_j - \mathbf{x}_i'\boldsymbol{\beta}$$

The probability that the unobserved dependent variable is contained in the jth category can be written as

$$P[\mu_{j-1} < y_i^* \le \mu_j] = F(R_{i,j}) - F(R_{i,j-1})$$

For more information, see the section "Ordinal Discrete Choice Modeling" on page 1985.

Limited Dependent Variable Models

Censored Regression Models

When the dependent variable is censored, values in a certain range are all transformed to a single value. For example, the standard Tobit model can be defined as

$$y_i^* = \mathbf{x}_i'\boldsymbol{\beta} + \epsilon_i$$

$$y_i = \begin{cases} y_i^* & \text{if} y_i^* > 0 \\ 0 & \text{if} y_i^* \le 0 \end{cases}$$

where $\epsilon_i \sim iidN(0, \sigma^2)$.

The Tobit model can be generalized to handle observation-by-observation censoring. The censored model on both the lower and upper limits can be defined as

$$y_i = \begin{cases} R_i & \text{if } y_i^* \ge R_i \\ y_i^* & \text{if } L_i < y_i^* < R_i \\ L_i & \text{if } y_i^* \le L_i \end{cases}$$

You can see Chapter 29.7, "Censored Regression Models," for more details.

Truncated Regression Models

In a truncated model, the observed sample is a subset of the population where the dependent variable falls within a certain range. For example, when neither a dependent variable nor exogenous variables are observed for $y_i^* \le 0$, the truncated regression model can be specified as

$$\ell = \sum_{i \in \{y_i > 0\}} \left\{ -\ln \Phi(\mathbf{x}_i'\boldsymbol{\beta}/\sigma) + \ln\left[\frac{\phi((y_i - \mathbf{x}_i'\boldsymbol{\beta})/\sigma)}{\sigma}\right]\right\}$$

For more information, see the section "Truncated Regression Models" on page 1990.

Stochastic Frontier Production and Cost Models

Stochastic frontier production models were first developed by Aigner, Lovell, and Schmidt (1977); Meeusen and van den Broeck (1977). Specification of these models allow for random shocks of the production or cost but also include a term for technical or cost inefficiency. Assuming that the production function takes a log-linear Cobb-Douglas form, the stochastic frontier production model can be written as

$$ln(y_i) = \beta_0 + \sum_n \boldsymbol{\beta}_n \ln(x_{ni}) + \epsilon_i$$

where $\epsilon_i = v_i - u_i$. The v_i term represents the stochastic error component, and the u_i term represents the nonnegative, technical inefficiency error component. The v_i error component is assumed to be distributed iid normal and independent from u_i. If $u_i > 0$, the error term ϵ_i is negatively skewed and represents technical inefficiency. If $u_i < 0$, the error term ϵ_i is positively skewed and represents cost inefficiency. PROC HPQLIM models the u_i error component as a half-normal, exponential, or truncated normal distribution.

The Normal-Half-Normal Model

When v_i is iid $N(0, \sigma_v^2)$ in a normal-half-normal model, u_i is iid $N^+(0, \sigma_u^2)$, with v_i and u_i independent of each other. Given the independence of error terms, the joint density of v and u can be written as

$$f(u, v) = \frac{2}{2\pi\sigma_u\sigma_v} \exp\left\{-\frac{u^2}{2\sigma_u^2} - \frac{v^2}{2\sigma_v^2}\right\}$$

Substituting $v = \epsilon + u$ into the preceding equation and integrating u out gives

$$f(\epsilon) \quad = \quad \frac{2}{\sigma}\phi\left(\frac{\epsilon}{\sigma}\right)\Phi\left(-\frac{\epsilon\lambda}{\sigma}\right)$$

where $\lambda = \sigma_u/\sigma_v$ and $\sigma = \sqrt{\sigma_u^2 + \sigma_v^2}$.

In the case of a stochastic frontier cost model, $v = \epsilon - u$ and

$$f(\epsilon) = \frac{2}{\sigma}\phi\left(\frac{\epsilon}{\sigma}\right)\Phi\left(\frac{\epsilon\lambda}{\sigma}\right)$$

For more information, see the section "Stochastic Frontier Production and Cost Models" on page 1991.

The Normal-Exponential Model

Under the normal-exponential model, v_i is iid $N(0, \sigma_v^2)$ and u_i is iid exponential. Given the independence of error term components u_i and v_i, the joint density of v and u can be written as

$$f(u, v) = \frac{1}{\sqrt{2\pi}\sigma_u\sigma_v} \exp\left\{-\frac{u}{\sigma_u} - \frac{v^2}{2\sigma_v^2}\right\}$$

The marginal density function of ϵ for the production function is

$$f(\epsilon) \quad = \quad \left(\frac{1}{\sigma_u}\right)\Phi\left(-\frac{\epsilon}{\sigma_v} - \frac{\sigma_v}{\sigma_u}\right)\exp\left\{\frac{\epsilon}{\sigma_u} + \frac{\sigma_v^2}{2\sigma_u^2}\right\}$$

The marginal density function for the cost function is equal to

$$f(\epsilon) = \left(\frac{1}{\sigma_u}\right)\Phi\left(\frac{\epsilon}{\sigma_v} - \frac{\sigma_v}{\sigma_u}\right)\exp\left\{-\frac{\epsilon}{\sigma_u} + \frac{\sigma_v^2}{2\sigma_u^2}\right\}$$

For more information, see the section "Stochastic Frontier Production and Cost Models" on page 1991.

The Normal–Truncated Normal Model

The normal–truncated normal model is a generalization of the normal-half-normal model that allows the mean of u_i to differ from zero. Under the normal–truncated normal model, the error term component v_i is iid $N^+(0, \sigma_v^2)$ and u_i is iid $N(\mu, \sigma_u^2)$. The joint density of v_i and u_i can be written as

$$f(u, v) = \frac{1}{\sqrt{2\pi}\sigma_u\sigma_v\Phi(\mu/\sigma_u)}\exp\left\{-\frac{(u-\mu)^2}{2\sigma_u^2} - \frac{v^2}{2\sigma_v^2}\right\}$$

The marginal density function of ϵ for the production function is

$$f(\epsilon) = \frac{1}{\sigma}\phi\left(\frac{\epsilon+\mu}{\sigma}\right)\Phi\left(\frac{\mu}{\sigma\lambda} - \frac{\epsilon\lambda}{\sigma}\right)\left[\Phi\left(\frac{\mu}{\sigma_u}\right)\right]^{-1}$$

The marginal density function for the cost function is

$$f(\epsilon) = \frac{1}{\sigma}\phi\left(\frac{\epsilon-\mu}{\sigma}\right)\Phi\left(\frac{\mu}{\sigma\lambda} + \frac{\epsilon\lambda}{\sigma}\right)\left[\Phi\left(\frac{\mu}{\sigma_u}\right)\right]^{-1}$$

For more information, see the section "Stochastic Frontier Production and Cost Models" on page 1991.

For more information about normal-half-normal, normal-exponential, and normal–truncated normal models, see Kumbhakar and Lovell (2000); Coelli, Prasada Rao, and Battese (1998).

Heteroscedasticity

If the variance of regression disturbance, (ϵ_i), is heteroscedastic, the variance can be specified as a function of variables

$$E(\epsilon_i^2) = \sigma_i^2 = f(\mathbf{z}_i'\boldsymbol{\gamma})$$

Table 22.2 shows various functional forms of heteroscedasticity and the corresponding options to request each model.

Table 22.2 Specification Summary for Modeling Heteroscedasticity

Number	Model	Options
1	$f(\mathbf{z}_i'\boldsymbol{\gamma}) = \sigma^2(1 + \exp(\mathbf{z}_i'\gamma))$	LINK=EXP (default)
2	$f(\mathbf{z}_i'\boldsymbol{\gamma}) = \sigma^2\exp(\mathbf{z}_i'\gamma)$	LINK=EXP NOCONST
3	$f(\mathbf{z}_i'\boldsymbol{\gamma}) = \sigma^2(1 + \sum_{l=1}^{L}\gamma_l z_{li})$	LINK=LINEAR
4	$f(\mathbf{z}_i'\boldsymbol{\gamma}) = \sigma^2(1 + (\sum_{l=1}^{L}\gamma_l z_{li})^2)$	LINK=LINEAR SQUARE
5	$f(\mathbf{z}_i'\boldsymbol{\gamma}) = \sigma^2(\sum_{l=1}^{L}\gamma_l z_{li})$	LINK=LINEAR NOCONST
6	$f(\mathbf{z}_i'\boldsymbol{\gamma}) = \sigma^2((\sum_{l=1}^{L}\gamma_l z_{li})^2)$	LINK=LINEAR SQUARE NOCONST

In models 3 and 5, variances of some observations might be negative. Although the HPQLIM procedure assigns a large penalty to move the optimization away from such a region, the optimization might not be able to improve the objective function value and might become locked in the region. Signs of such an outcome include extremely small likelihood values or missing standard errors in the estimates. In models 2 and 6, variances are guaranteed to be greater than or equal to zero, but variances of some observations might be very close to 0. In these scenarios, standard errors might be missing. Models 1 and 4 do not have such problems. Variances in these models are always positive and never close to 0.

For more information, see the section "Heteroscedasticity and Box-Cox Transformation" on page 1993.

Tests on Parameters

In general, the tested hypothesis can be written as

$$H_0 : \mathbf{h}(\theta) = 0$$

where $\mathbf{h}(\theta)$ is an $r \times 1$ vector-valued function of the parameters θ given by the r expressions that are specified in the TEST statement.

Let $\hat{V}$ be the estimate of the covariance matrix of $\hat{\theta}$. Let $\hat{\theta}$ be the unconstrained estimate of θ and $\tilde{\theta}$ be the constrained estimate of θ such that $h(\tilde{\theta}) = 0$. Let

$$A(\theta) = \partial h(\theta)/\partial\theta \mid_{\hat{\theta}}$$

Using this notation, the test statistics for the three types of tests are computed as follows.

- The Wald test statistic is defined as

  $$W = h'(\hat{\theta}) \left(A(\hat{\theta})\hat{V}A'(\hat{\theta}) \right)^{-1} h(\hat{\theta})$$

- The Lagrange multiplier test statistic is

  $$LM = \lambda' A(\tilde{\theta})\tilde{V}A'(\tilde{\theta})\lambda$$

  where λ is the vector of Lagrange multipliers from the computation of the restricted estimate $\tilde{\theta}$.

- The likelihood ratio test statistic is

  $$LR = 2\left(L(\hat{\theta}) - L(\tilde{\theta}) \right)$$

  where $\tilde{\theta}$ represents the constrained estimate of θ and L is the concentrated log-likelihood value.

The following statements use the TEST statement to perform a likelihood ratio test:

```
proc hpqlim;
   model y = x1 x2 x3;
   test x1 = 0, x2 * .5 + 2 * x3 = 0 /lr;
run;
```

For more information, see the section "Tests on Parameters" on page 2000.

Bayesian Analysis

To perform Bayesian analysis, you must specify a BAYES statement. Unless otherwise stated, all options that are described in this section are options in the BAYES statement.

By default, PROC HPQLIM uses the random walk Metropolis algorithm to obtain posterior samples. For the implementation details of the Metropolis algorithm in PROC HPQLIM, such as the blocking of the parameters and tuning of the covariance matrices, see the sections "Blocking of Parameters" on page 1174 and "Tuning the Proposal Distribution" on page 1174.

The Bayes theorem states that

$$p(\theta|\mathbf{y}) \propto \pi(\theta)L(y|\theta)$$

where θ is a parameter or a vector of parameters and $\pi(\theta)$ is the product of the prior densities that are specified in the PRIOR statement. The term $L(y|\theta)$ is the likelihood that is associated with the MODEL statement.

Blocking of Parameters

In a multivariate parameter model, all the parameters are updated in one single block (by default or when you specify the SAMPLING=MULTIMETROPOLIS option). This can be inefficient, especially when parameters have vastly different scales. As an alternative, you can update the parameters one at a time (by specifying SAMPLING=UNIMETROPOLIS).

Tuning the Proposal Distribution

One key factor in achieving high efficiency of a Metropolis-based Markov chain is finding a good proposal distribution for each block of parameters. This process is called tuning. The tuning phase consists of a number of loops that are controlled by the options MINTUNE and MAXTUNE. The MINTUNE= option controls the minimum number of tuning loops and has a default value of 2. The MAXTUNE= option controls the maximum number of tuning loops and has a default value of 24. Each loop repeats the number of times specified by the NTU= option, which has a default of 500. At the end of every loop, PROC HPQLIM examines the acceptance probability for each block. The acceptance probability is the percentage of NTU proposed values that have been accepted. If this probability does not fall within the acceptance tolerance range (see the following section), the proposal distribution is modified before the next tuning loop.

A good proposal distribution should resemble the actual posterior distribution of the parameters. Large sample theory states that the posterior distribution of the parameters approaches a multivariate normal distribution (Gelman et al. 2004, Appendix B; Schervish 1995, Section 7.4). That is why a normal proposal distribution often works well in practice. The default proposal distribution in PROC HPQLIM is the normal distribution.

You can see Chapter 29.7, "Bayesian Analysis," for more details.

Initial Values of the Markov Chains

You can assign initial values to any parameters. For more information, see the INIT statement. If you use the optimization PROPCOV= option, PROC HPQLIM starts the tuning at the optimized values. This option overwrites the provided initial values.

Prior Distributions

The PRIOR statement specifies the prior distribution of the model parameters. You must specify one parameter or a list of parameters, a tilde ~, and then a distribution with its parameters. You can specify multiple PRIOR statements to define independent priors. Parameters that are associated with a regressor variable are referred to by the name of the corresponding regressor variable.

You can specify the special keyword _REGRESSORS to consider all the regressors of a model. If multiple PRIOR statements affect the same parameter, the last PRIOR statement prevails. For example, in a regression with two regressors (X1, X2), the following statements imply that the prior on X1 is NORMAL(MEAN=0, VAR=1), the prior on X2 is GAMMA(SHAPE=3, SCALE=4).

```
...
prior _Regressors ~ uniform(min=0, max=1);
prior X1 X2 ~ gamma(shape=3, scale=4);
prior X1 ~ normal(mean=0, var=1);
...
```

If a parameter is not associated with a PRIOR statement or if some of the prior hyperparameters are missing, then the default choices in Table Table 22.3 are considered.

Table 22.3 Default Values for Prior Distributions

PRIOR Distribution	Hyperparameter$_1$	Hyperparameter$_2$	Min	Max	Parameters Default Choice
NORMAL	MEAN=0	VAR=1E6	$-\infty$	∞	Regression-Location-Threshold
IGAMMA	SHAPE=2.000001	SCALE=1	> 0	∞	Scale
GAMMA	SHAPE=1	SCALE=1	0	∞	
UNIFORM			$-\infty$	∞	
BETA	SHAPE1=1	SHAPE2=1	$-\infty$	∞	
T	LOCATION=0	DF=3	$-\infty$	∞	

For density specification, see the section "Standard Distributions" on page 1175.

Standard Distributions

Table 22.4 through Table 22.9 show all the distribution density functions that PROC HPQLIM recognizes. You specify these distribution densities in the PRIOR statement.

Table 22.4 Beta Distribution

PRIOR statement	BETA(SHAPE1=a, SHAPE2=b, MIN=m, MAX=M)
	Note: Commonly $m = 0$ and $M = 1$.
Density	$\frac{(\theta-m)^{a-1}(M-\theta)^{b-1}}{B(a,b)(M-m)^{a+b-1}}$
Parameter restriction	$a > 0, \quad b > 0, \quad -\infty < m < M < \infty$

Range	$\begin{cases} [m, M] & \text{when } a = 1, b = 1 \\ [m, M) & \text{when } a = 1, b \neq 1 \\ (m, M] & \text{when } a \neq 1, b = 1 \\ (m, M) & \text{otherwise} \end{cases}$
Mean	$\frac{a}{a+b} \times (M - m) + m$
Variance	$\frac{ab}{(a+b)^2(a+b+1)} \times (M - m)^2$
Mode	$\begin{cases} \frac{a-1}{a+b-2} \times M + \frac{b-1}{a+b-2} \times m & a > 1, b > 1 \\ m \text{ and } M & a < 1, b < 1 \\ m & \begin{cases} a < 1, b \geq 1 \\ a = 1, b > 1 \end{cases} \\ M & \begin{cases} a \geq 1, b < 1 \\ a > 1, b = 1 \end{cases} \\ \text{not unique} & a = b = 1 \end{cases}$
Defaults	SHAPE1=SHAPE2=1, MIN $\rightarrow -\infty$, MAX $\rightarrow \infty$

Table 22.5 Gamma Distribution

PRIOR statement	GAMMA(SHAPE=a, SCALE=b)
Density	$\frac{1}{b^a \Gamma(a)} \theta^{a-1} e^{-\theta/b}$
Parameter restriction	$a > 0, b > 0$
Range	$[0, \infty)$
Mean	ab
Variance	ab^2
Mode	$(a - 1)b$
Defaults	SHAPE=SCALE=1

Table 22.6 Inverse Gamma Distribution

PRIOR statement	IGAMMA(SHAPE=a, SCALE=b)
Density	$\frac{b^a}{\Gamma(a)}\theta^{-(a+1)}e^{-b/\theta}$
Parameter restriction	$a > 0, b > 0$
Range	$0 < \theta < \infty$
Mean	$\frac{b}{a-1}, \qquad a > 1$
Variance	$\frac{b^2}{(a-1)^2(a-2)}, \qquad a > 2$
Mode	$\frac{b}{a+1}$
Defaults	SHAPE=2.000001, SCALE=1

Table 22.7 Normal Distribution

PRIOR statement	NORMAL(MEAN=μ, VAR=σ^2)
Density	$\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(\theta-\mu)^2}{2\sigma^2}\right)$
Parameter restriction	$\sigma^2 > 0$
Range	$-\infty < \theta < \infty$
Mean	μ
Variance	σ^2
Mode	μ
Defaults	MEAN=0, VAR=1000000

Table 22.8 *t* Distribution

PRIOR statement	T(LOCATION=μ, DF=ν)
Density	$\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\pi\nu}}\left[1+\frac{(\theta-\mu)^2}{\nu}\right]^{-\frac{\nu+1}{2}}$
Parameter restriction	$\nu > 0$
Range	$-\infty < \theta < \infty$
Mean	μ, for $\nu > 1$
Variance	$\frac{\nu}{\nu-2}$, for $\nu > 2$
Mode	μ
Defaults	LOCATION=0, DF=3

Table 22.9 Uniform Distribution

PRIOR statement	UNIFORM(MIN=m, MAX=M)
Density	$\frac{1}{M-m}$
Parameter restriction	$-\infty < m < M < \infty$
Range	$\theta \in [m, M]$
Mean	$\frac{m+M}{2}$
Variance	$\frac{(M-m)^2}{12}$
Mode	Not unique
Defaults	MIN$\rightarrow -\infty$, MAX$\rightarrow \infty$

Output to SAS Data Set

XBeta, Predicted, and Residual

Xbeta is the structural part on the right-hand side of the model. The predicted value is the predicted dependent variable value. For censored variables, if the predicted value is outside the boundaries, it is reported as the closest boundary. The residual is defined only for continuous variables and is defined as

$$\text{Residual} = \text{Observed} - \text{Predicted}$$

Error Standard Deviation

The error standard deviation is σ_i in the model. It varies only when the HETERO statement is used.

Marginal Effects

A marginal effect is defined as a contribution of one control variable to the response variable. For a binary choice model with two response categories, $\mu_0 = -\infty$ and $\mu_1 = 0$, $\mu_2 = \infty$. For an ordinal response model with M response categories $(\mu_0, \cdots, \mu_M)$, define

$$R_{i,j} = \mu_j - \mathbf{x}_i'\boldsymbol{\beta}$$

The probability that the unobserved dependent variable is contained in the jth category can be written as

$$P[\mu_{j-1} < y_i^* \leq \mu_j] = F(R_{i,j}) - F(R_{i,j-1})$$

The marginal effect of changes in the regressors on the probability of $y_i = j$ is then

$$\frac{\partial \text{Prob}[y_i = j]}{\partial \mathbf{x}} = [f(\mu_{j-1} - \mathbf{x}_i'\boldsymbol{\beta}) - f(\mu_j - \mathbf{x}_i'\boldsymbol{\beta})]\boldsymbol{\beta}$$

where $f(x) = \frac{dF(x)}{dx}$. In particular,

$$f(x) = \frac{dF(x)}{dx} = \begin{cases} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} & \text{(probit)} \\ \frac{e^{-x}}{[1+e^{(-x)}]^2} & \text{(logit)} \end{cases}$$

The marginal effects in the truncated regression model are

$$\frac{\partial E[y_i|L_i < y_i^* < R_i]}{\partial \mathbf{x}} = \boldsymbol{\beta}\left[1 - \frac{(\phi(a_i) - \phi(b_i))^2}{(\Phi(b_i) - \Phi(a_i))^2} + \frac{a_i\phi(a_i) - b_i\phi(b_i)}{\Phi(b_i) - \Phi(a_i)}\right]$$

where $a_i = \frac{L_i - \mathbf{x}_i'\boldsymbol{\beta}}{\sigma_i}$ and $b_i = \frac{R_i - \mathbf{x}_i'\boldsymbol{\beta}}{\sigma_i}$.

The marginal effects in the censored regression model are

$$\frac{\partial E[y|\mathbf{x}_i]}{\partial \mathbf{x}} = \boldsymbol{\beta} \times \text{Prob}[L_i < y_i^* < R_i]$$

Expected and Conditionally Expected Values

The expected value is the unconditional expectation of the dependent variable. For a censored variable, it is

$$E[y_i] = \Phi(a_i)L_i + (\mathbf{x}_i'\boldsymbol{\beta} + \lambda\sigma_i)(\Phi(b_i) - \Phi(a_i)) + (1 - \Phi(b_i))R_i$$

For a left-censored variable ($R_i = \infty$), this formula is

$$E[y_i] = \Phi(a_i)L_i + (\mathbf{x}_i'\boldsymbol{\beta} + \lambda\sigma_i)(1 - \Phi(a_i))$$

where $\lambda = \frac{\phi(a_i)}{1-\Phi(a_i)}$.

For a right-censored variable ($L_i = -\infty$), this formula is

$$E[y_i] = (\mathbf{x}_i'\boldsymbol{\beta} + \lambda\sigma_i)\Phi(b_i) + (1 - \Phi(b_i))R_i$$

where $\lambda = -\frac{\phi(b_i)}{\Phi(b_i)}$.

For a noncensored variable, this formula is

$$E[y_i] = \mathbf{x}_i'\boldsymbol{\beta}$$

The conditional expected value is the expectation when the variable is inside the boundaries:

$$E[y_i|L_i < y_i < R_i] = \mathbf{x}_i'\boldsymbol{\beta} + \lambda\sigma_i$$

Technical Efficiency

Technical efficiency for each producer is computed only for stochastic frontier models.

In general, the stochastic production frontier can be written as

$$y_i = f(x_i;\beta)\exp\{v_i\}TE_i$$

where y_i denotes producer i's actual output, $f(\cdot)$ is the deterministic part of the production frontier, $\exp\{v_i\}$ is a producer-specific error term, and TE_i is the technical efficiency coefficient, which can be written as

$$TE_i = \frac{y_i}{f(x_i;\beta)\exp\{v_i\}}$$

For a Cobb-Douglas production function, $TE_i = \exp\{-u_i\}$. For more information, see the section "Stochastic Frontier Production and Cost Models" on page 1171.

The cost frontier can be written in general as

$$E_i = c(y_i, w_i; \beta) \exp\{v_i\}/CE_i$$

where w_i denotes producer i's input prices, $c(\cdot)$ is the deterministic part of the cost frontier, $\exp\{v_i\}$ is a producer-specific error term, and CE_i is the cost efficiency coefficient, which can be written as

$$CE_i = \frac{c(x_i, w_i; \beta) \exp\{v_i\}}{E_i}$$

For a Cobb-Douglas cost function, $CE_i = \exp\{-u_i\}$. For more information, see the section "Stochastic Frontier Production and Cost Models" on page 1171. Hence, both technical and cost efficiency coefficients are the same. The estimates of technical efficiency are provided in the following subsections.

Normal-Half-Normal Model

Define $\mu_* = -\epsilon\sigma_u^2/\sigma^2$ and $\sigma_*^2 = \sigma_u^2\sigma_v^2/\sigma^2$. Then, as shown by Jondrow et al. (1982), conditional density is as follows:

$$f(u|\epsilon) = \frac{f(u,\epsilon)}{f(\epsilon)} = \frac{1}{\sqrt{2\pi}\sigma_*} \exp\left\{-\frac{(u-\mu_*)^2}{2\sigma_*^2}\right\} \Big/ \left[1 - \Phi\left(-\frac{\mu_*}{\sigma_*}\right)\right]$$

Hence, $f(u|\epsilon)$ is the density for $N^+(\mu_*, \sigma_*^2)$.

From this result, it follows that the estimate of technical efficiency (Battese and Coelli 1988) is

$$TE1_i = E(\exp\{-u_i\}|\epsilon_i) = \left[\frac{1 - \Phi(\sigma_* - \mu_{*i}/\sigma_*)}{1 - \Phi(-\mu_{*i}/\sigma_*)}\right] \exp\left\{-\mu_{*i} + \frac{1}{2}\sigma_*^2\right\}$$

The second version of the estimate (Jondrow et al. 1982) is

$$TE2_i = \exp\{-E(u_i|\epsilon_i)\}$$

where

$$E(u_i|\epsilon_i) = \mu_{*i} + \sigma_*\left[\frac{\phi(-\mu_{*i}/\sigma_*)}{1 - \Phi(-\mu_{*i}/\sigma_*)}\right] = \sigma_*\left[\frac{\phi(\epsilon_i\lambda/\sigma)}{1 - \Phi(\epsilon_i\lambda/\sigma)} - \left(\frac{\epsilon_i\lambda}{\sigma}\right)\right]$$

Normal-Exponential Model

Define $A = -\tilde{\mu}/\sigma_v$ and $\tilde{\mu} = -\epsilon - \sigma_v^2/\sigma_u$. Then, as shown by Kumbhakar and Lovell (2000), conditional density is as follows:

$$f(u|\epsilon) = \frac{1}{\sqrt{2\pi}\sigma_v\Phi(-\tilde{\mu}/\sigma_v)} \exp\left\{-\frac{(u-\tilde{\mu})^2}{2\sigma^2}\right\}$$

Hence, $f(u|\epsilon)$ is the density for $N^+(\tilde{\mu}, \sigma_v^2)$.

From this result, it follows that the estimate of technical efficiency is

$$TE1_i = E(\exp\{-u_i\}|\epsilon_i) = \left[\frac{1 - \Phi(\sigma_v - \tilde{\mu}_i/\sigma_v)}{1 - \Phi(-\tilde{\mu}_i/\sigma_v)}\right] \exp\left\{-\tilde{\mu}_i + \frac{1}{2}\sigma_v^2\right\}$$

The second version of the estimate is

$$TE2_i = \exp\{-E(u_i|\epsilon_i)\}$$

where

$$E(u_i|\epsilon_i) = \tilde{\mu}_i + \sigma_v \left[\frac{\phi(-\tilde{\mu}_i/\sigma_v)}{1-\Phi(-\tilde{\mu}_i/\sigma_v)}\right] = \sigma_v \left[\frac{\phi(A)}{\Phi(-A)} - A\right]$$

Normal–Truncated Normal Model

Define $\tilde{\mu} = (-\sigma_u^2\epsilon_i + \mu\sigma_v^2)/\sigma^2$ and $\sigma_*^2 = \sigma_u^2\sigma_v^2/\sigma^2$. Then, as shown by Kumbhakar and Lovell (2000), conditional density is as follows:

$$f(u|\epsilon) = \frac{1}{\sqrt{2\pi}\sigma_*[1-\Phi(-\tilde{\mu}/\sigma_*)]} \exp\left\{-\frac{(u-\tilde{\mu})^2}{2\sigma_*^2}\right\}$$

Hence, $f(u|\epsilon)$ is the density for $N^+(\tilde{\mu}, \sigma_*^2)$.

From this result, it follows that the estimate of technical efficiency is

$$TE1_i = E(\exp\{-u_i\}|\epsilon_i) = \frac{1-\Phi(\sigma_* - \tilde{\mu}_i/\sigma_*)}{1-\Phi(-\tilde{\mu}_i/\sigma_*)} \exp\left\{-\tilde{\mu}_i + \frac{1}{2}\sigma_*^2\right\}$$

The second version of the estimate is

$$TE2_i = \exp\{-E(u_i|\epsilon_i)\}$$

where

$$E(u_i|\epsilon_i) = \tilde{\mu}_i + \sigma_* \left[\frac{\phi(\tilde{\mu}_i/\sigma_*)}{1-\Phi(-\tilde{\mu}_i/\sigma_*)}\right]$$

OUTEST= Data Set

The OUTEST= data set contains all the parameters that are estimated by a MODEL statement. Each parameter contains the estimate for the corresponding parameter in the corresponding model. In addition, the OUTEST= data set contains the following variables:

NAME	indicates the name of the independent variable.
TYPE	indicates the type of observation. PARM indicates the row of coefficients; STD indicates the row of standard deviations of the corresponding coefficients.
STATUS	indicates the convergence status for optimization.

The rest of the columns correspond to the explanatory variables.

The OUTEST= data set contains one observation for the MODEL statement, which shows the parameter estimates for that model. If you specify the COVOUT option in the PROC HPQLIM statement, the OUTEST= data set includes additional observations for the MODEL statement, which show the rows of the covariance matrix of parameter estimates. For covariance observations, the value of the _TYPE_ variable is COV, and the _NAME_ variable identifies the parameter that is associated with that row of the covariance matrix. If you specify the CORROUT option in the PROC HPQLIM statement, the OUTEST= data set includes additional observations for the MODEL statement, which show the rows of the correlation matrix of parameter estimates. For correlation observations, the value of the _TYPE_ variable is CORR, and the _NAME_ variable identifies the parameter that is associated with that row of the correlation matrix.

Naming

Naming of Parameters

The parameters are named in the same way as in other SAS procedures such as the REG and PROBIT procedures. The constant in the regression equation is called Intercept. The coefficients of independent variables are named by the independent variables. The standard deviation of the errors is called _Sigma. If the HETERO statement is included, the coefficients of the independent variables in the HETERO statement are called _H.*x*, where *x* is the name of the independent variable.

Naming of Output Variables

Table 22.10 shows the *options* in the OUTPUT statement, with the corresponding variable names and their explanations.

Table 22.10 OUTPUT Statement Options

output-option	**Variable Name**	**Explanation**
CONDITIONAL	CEXPCT_y	Conditional expected value of y, conditioned on the truncation
ERRSTD	ERRSTD_y	Standard deviation of error term
EXPECTED	EXPCT_y	Unconditional expected value of y
MARGINAL	MEFF_x	Marginal effect of x on y ($\frac{\partial y}{\partial x}$) with single equation
PREDICTED	P_y	Predicted value of y
RESIDUAL	RESID_y	Residual of y, (y – PredictedY)
PROB	PROB_y	Probability that y is taking the observed value in this observation (discrete y only)
PROBALL	PROBi_y	Probability that y is taking the ith value (discrete y only)
MILLS	MILLS_y	Inverse Mills ratio for y
TE1	TE1	Technical efficiency estimate for each producer proposed by Battese and Coelli (1988)
TE2	TE2	Technical efficiency estimate for each producer proposed by Jondrow et al. (1982)
XBETA	XBETA_y	Structure part ($\mathbf{x}'\boldsymbol{\beta}$) of y equation

If you prefer to name the output variables differently, you can use the RENAME option in the data set. For example, the following statements rename the residual of y as *Resid*:

```
proc hpqlim data=one;
   model y = x1-x10 / censored;
   output out=outds(rename=(resid_y=resid)) residual;
run;
```

ODS Table Names

PROC HPQLIM assigns a name to each table that 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 22.11.

Table 22.11 ODS Tables Produced in PROC HPQLIM

ODS Table Name	Description	Option
ODS Tables Created by the MODEL Statement and TEST Statement		
ResponseProfile	Response profile	Default
FitSummary	Summary of nonlinear estimation	Default
ParameterEstimates	Parameter estimates	Default
SummaryContResponse	Summary of continuous response	Default
CovB	Covariance of parameter estimates	COVB
CorrB	Correlation of parameter estimates	CORRB
ODS Tables Created by the BAYES Statement		
AutoCorr	Autocorrelation statistics for each parameter	Default
Corr	Correlation matrix of the posterior samples	STATS=COR
Cov	Covariance matrix of the posterior samples	STATS=COV
ESS	Effective sample size for each parameter	Default
MCSE	Monte Carlo standard error for each parameter	Default
Geweke	Geweke diagnostics for each parameter	Default
Heidelberger	Heidelberger-Welch diagnostics for each parameter	DIAGNOSTICS=HEIDEL
PostIntervals	Equal-tail and HPD intervals for each parameter	Default
PosteriorSample	Posterior samples	(ODS output data set only)
PostSummaries	Posterior summaries	Default
PriorSummaries	Prior summaries	STATS=PRIOR
Raftery	Raftery-Lewis diagnostics for each parameter	DIAGNOSTICS=RAFTERY
ODS Tables Created by the TEST Statement		
TestResults	Test results	Default

ODS Graphics

You can use a name to reference every graph that is produced through ODS Graphics. The names of the graphs that PROC HPQLIM generates are listed in Table 22.12.

Table 22.12 Graphs Produced by PROC HPQLIM When a BAYES Statement Is Included

ODS Graph Name	Plot Description	Statement and Option
Bayesian Diagnostic Plots		
ADPanel	Autocorrelation function and density panel	PLOTS=(AUTOCORR DENSITY)
AutocorrPanel	Autocorrelation function panel	PLOTS=AUTOCORR
AutocorrPlot	Autocorrelation function plot	PLOTS(UNPACK)=AUTOCORR
DensityPanel	Density panel	PLOTS=DENSITY
DensityPlot	Density plot	PLOTS(UNPACK)=DENSITY
TAPanel	Trace and autocorrelation function panel	PLOTS=(TRACE AUTOCORR)
TADPanel	Trace, density, and autocorrelation function panel	PLOTS=(TRACE AUTOCORR DENSITY)
TDPanel	Trace and density panel	PLOTS=(TRACE DENSITY)
TracePanel	Trace panel	PLOTS=TRACE
TracePlot	Trace plot	PLOTS(UNPACK)=TRACE

Examples: The HPQLIM Procedure

Example 22.1: High-Performance Model with Censoring

This example shows the use of the HPQLIM procedure with an emphasis on processing a large data set and on the performance improvements that are achieved by executing in the high-performance distributed environment.

The following DATA step generates 5 million replicates from a censored model. The model contains seven variables.

```
data simulate;
     call streaminit(12345);
     array vars x1-x7;
     array parms{7}  (3 4 2 4 -3 -5 -3);

     intercept=2;

    do i=1 to 5000000;
        sum_xb=0;
```

```
            do j=1 to 7;
               vars[j]=rand('NORMAL',0,1);
               sum_xb=sum_xb+parms[j]*vars[j];
            end;
            y=intercept+sum_xb+400*rand('NORMAL',0,1);
            if y>400 then y=400;
            if y<0 then y=0;
            output;
         end;
      keep y x1-x7;
      run;
```

The following statements estimate a censored model. The model is executed in the distributed computing environment with two threads and only one node. These settings are used to obtain a hypothetical environment that might resemble running the HPQLIM procedure on a desktop workstation with a dual-core CPU. To run these statements successfully, you need to set the macro variables GRIDHOST and GRIDINSTALLLOC to resolve to appropriate values, or you can replace the references to the macro variables in the example with the appropriate values.

```
 option set=GRIDHOST="&GRIDHOST";
 option set=GRIDINSTALLLOC="&GRIDINSTALLLOC";


proc hpqlim data=simulate ;
   performance nthreads=2 nodes=1 details
                host="&GRIDHOST" install="&GRIDINSTALLLOC";
   model y=x1-x7 /censored(lb=0 ub=400);
run;
```

Output 22.1.1 shows that the censored model was estimated on the grid, defined in a macro variable named GRIDHOST, in a distributed environment on only one node with two threads.

Output 22.1.1 Censored Model with One Node and Two Threads: Performance Table

Estimating a Tobit model

Performance Information	
Host Node	<< your grid host >>
Install Location	<< your grid install location >>
Execution Mode	Distributed
Number of Compute Nodes	1
Number of Threads per Node	2

Output 22.1.2 shows the estimation results for the censored model. The “Model Fit Summary” table shows detailed information about the model and indicates that all 5 million observations were used to fit the model. All parameter estimates in the “Parameter Estimates” table are highly significant and correspond to their theoretical values that were set during the data generating process. The optimization of the model with 5 million observations took 45.4 seconds.

Output 22.1.2 Censored Model with One Node and Two Threads: Summary

Model Information	
Data Source	SIMULATE
Response Variable	y
Optimization Technique	Quasi-Newton

Number of Observations	
Number of Observations Read	5000000
Number of Observations Used	5000000

Summary Statistics of Continuous Responses

Variable	Mean	Standard Error	Type	Lower Bound	Upper Bound	N Obs Lower Bound	N Obs Upper Bound
y	127.0	159.491090	Censored	0	400.0	249E4	8E5

Convergence criterion (FCONV=2.220446E-16) satisfied.

Model Fit Summary	
Number of Endogenous Variables	1
Endogenous Variable	y
Number of Observations	5000000
Log Likelihood	-15268972
Maximum Absolute Gradient	0.0003291
Number of Iterations	11
Optimization Method	Quasi-Newton
AIC	30537962
Schwarz Criterion	30538083

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
Intercept	1	2.220379	0.222201	9.99	<.0001
x1	1	3.055533	0.201620	15.15	<.0001
x2	1	4.000176	0.201570	19.85	<.0001
x3	1	1.852740	0.201555	9.19	<.0001
x4	1	4.170266	0.201533	20.69	<.0001
x5	1	-3.010679	0.201458	-14.94	<.0001
x6	1	-5.176016	0.201541	-25.68	<.0001
x7	1	-2.695948	0.201671	-13.37	<.0001
_Sigma	1	399.997845	0.261930	1527.12	<.0001

Procedure Task Timing

Task	Seconds	Percent
Reading and Levelizing Data	1.46	3.12%
Communication to Client	0.09	0.19%
Optimization	45.39	96.69%
Post-optimization	0.00	0.00%

In the following statements, the PERFORMANCE statement is modified to use a grid with 10 nodes, with each node capable of spawning eight threads:

```
proc hpqlim data=simulate ;
   performance nthreads=8 nodes=10 details
                host="&GRIDHOST" install="&GRIDINSTALLLOC";
   model y=x1-x7 /censored(lb=0 ub=400);
run;
```

The second model which was run on a grid with 10 nodes and eight threads each (Output 22.1.3) took only 1.4 seconds instead of 45.4 seconds to optimize.

Output 22.1.3 Censored Model on Ten Nodes with Eight Threads Each: Performance Table

Estimating a Tobit model

Performance Information	
Host Node	<< your grid host >>
Install Location	<< your grid install location >>
Execution Mode	Distributed
Number of Compute Nodes	10
Number of Threads per Node	8

Because the two models being estimated are identical, it is reasonable to expect that Output 22.1.2 and Output 22.1.4 would show the same results except for the performance. However, in certain circumstances, you might observe slight numerical differences in the results (depending on the number of nodes and threads) because the order in which partial results are accumulated, the limits of numerical precision, and the propagation of error in numerical computations can make a difference in the final result.

Output 22.1.4 Censored Model on Ten Nodes with Eight Threads Each: Summary

Model Information	
Data Source	SIMULATE
Response Variable	y
Optimization Technique	Quasi-Newton

Number of Observations	
Number of Observations Read	5000000
Number of Observations Used	5000000

Summary Statistics of Continuous Responses

Variable	Mean	Standard Error	Type	Lower Bound	Upper Bound	N Obs Lower Bound	N Obs Upper Bound
y	127.0	159.491090	Censored	0	400.0	249E4	8E5

Convergence criterion (FCONV=2.220446E-16) satisfied.

Output 22.1.4 *continued*

Model Fit Summary	
Number of Endogenous Variables	1
Endogenous Variable	y
Number of Observations	5000000
Log Likelihood	-15268972
Maximum Absolute Gradient	0.0008332
Number of Iterations	10
Optimization Method	Quasi-Newton
AIC	30537962
Schwarz Criterion	30538083

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
Intercept	1	2.220358	0.222201	9.99	<.0001
x1	1	3.055491	0.201620	15.15	<.0001
x2	1	4.000196	0.201570	19.85	<.0001
x3	1	1.852735	0.201555	9.19	<.0001
x4	1	4.170323	0.201533	20.69	<.0001
x5	1	-3.010670	0.201458	-14.94	<.0001
x6	1	-5.176019	0.201541	-25.68	<.0001
x7	1	-2.695886	0.201671	-13.37	<.0001
_Sigma	1	399.997846	0.261930	1527.12	<.0001

Procedure Task Timing

Task	Seconds	Percent
Reading and Levelizing Data	0.09	5.77%
Communication to Client	0.12	7.67%
Optimization	1.38	86.56%
Post-optimization	0.00	0.00%

As this example suggests, increasing the number of nodes and the number of threads per node improves performance significantly. When you use the parallelism that a high-performance distributed environment affords, you can see an even more dramatic reduction in the time required for the optimization as the number of observations in the data set increases. When the data set is extremely large, the computations might not even be possible with the typical memory resources and computational constraints of a desktop computer. Under such circumstances the high-performance distributed environment becomes a necessity.

Example 22.2: Bayesian High-Performance Model with Censoring

This example shows the use of the Bayesian analysis available in the HPQLIM procedure with an emphasis on processing a large data set and on the performance improvements that are achieved by executing in a high-performance distributed environment.

The model and the data set are the same as in Example 22.1, and the priors are set to the defaults.

The model is executed in the distributed computing environment with two threads and only one node. These settings are used to obtain a hypothetical environment that might resemble running the HPQLIM procedure on a desktop workstation with a dual-core CPU. To run the following statements successfully, you need to set the macro variables GRIDHOST and GRIDINSTALLLOC to resolve to appropriate values, or you can replace the references to the macro variables in the example with the appropriate values.

```
 option set=GRIDHOST="&GRIDHOST";
 option set=GRIDINSTALLLOC="&GRIDINSTALLLOC";


proc hpqlim data=simulate ;
bayes nbi=10000 nmc=30000;
   performance nthreads=2 nodes=1 details
               host="&GRIDHOST" install="&GRIDINSTALLLOC";
   model y=x1-x7 /censored(lb=0 ub=400);
   %*;     ods output PerformanceInfo=perfInfo;
   %*;     ods output Timing=time;
run;
```

Output 22.2.1 shows a summary of the posterior distribution that is associated with the censored model when you use diffuse prior distributions.

Output 22.2.1 Posterior Summary for Bayesian Censored Model

Estimating a Tobit model

The HPQLIM Procedure

Posterior Summaries						
				Percentiles		
Parameter	N	Mean	Standard Deviation	25%	50%	75%
Intercept	30000	2.2168	0.2166	2.0682	2.2171	2.3674
x1	30000	3.0656	0.1953	2.9356	3.0638	3.1974
x2	30000	3.9936	0.2052	3.8572	3.9951	4.1308
x3	30000	1.8496	0.2019	1.7201	1.8475	1.9813
x4	30000	4.1637	0.1986	4.0295	4.1574	4.2996
x5	30000	-3.0226	0.1981	-3.1603	-3.0249	-2.8848
x6	30000	-5.1776	0.1985	-5.3099	-5.1720	-5.0456
x7	30000	-2.6889	0.1993	-2.8235	-2.6882	-2.5523
_Sigma	30000	400.0	0.2623	399.8	400.0	400.2

Output 22.2.2 show a summary of the performance when you use a distributed computing environment with one node and two threads.

Output 22.2.2 Performance Analysis for Bayesian Censored Model on One Node with Two Threads

Estimating a Tobit model

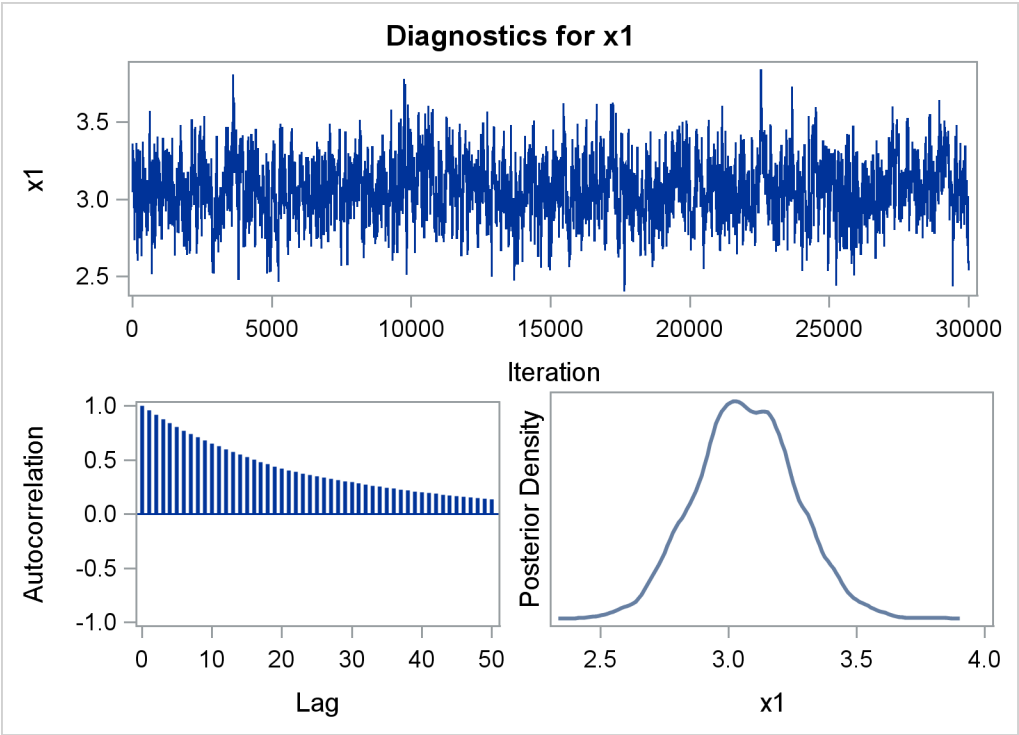
Performance Information	
Host Node	<< your grid host >>
Install Location	<< your grid install location >>
Execution Mode	Distributed
Number of Compute Nodes	1
Number of Threads per Node	2

Estimating a Tobit model

Procedure Task Timing		
Task	**Seconds**	**Percent**
Reading and Levelizing Data	1.59	0.00%
Communication to Client	0.06	0.00%
Bayesian Analysis: Likelihood for MCMC	46088.82	99.88%
Bayesian Analysis: MCMC	1.13	0.00%
Optimization	50.91	0.11%
Post-optimization	0.00	0.00%

Finally, Output 22.2.3 shows the diagnostic and summary plots that are associated with *X1*.

Output 22.2.3 Bayesian Diagnostic and Summary Plots for x1



In the following statements, the PERFORMANCE statement is modified to use a grid with 10 nodes, where each node spawns eight threads:

```
 option set=GRIDHOST="&GRIDHOST";
 option set=GRIDINSTALLLOC="&GRIDINSTALLLOC";


proc hpqlim data=simulate ;
bayes nbi=10000 nmc=30000;
   performance nthreads=8 nodes=10 details
               host="&GRIDHOST" install="&GRIDINSTALLLOC";
   model y=x1-x7 /censored(lb=0 ub=400);
   %*;     ods output PerformanceInfo=perfInfo;
   %*;     ods output Timing=time;
run;
```

The two models are identical, but the second implementation, which was run on a grid that used 10 nodes with eight threads each, took only 15.7 minutes instead of 12.8 hours to sample from the same posterior distribution.

Output 22.2.4 Performance Analysis for Bayesian Censored Model on Ten Nodes with Eight Threads Each

Estimating a Tobit model

Performance Information	
Host Node	<< your grid host >>
Install Location	<< your grid install location >>
Execution Mode	Distributed
Number of Compute Nodes	10
Number of Threads per Node	8

Estimating a Tobit model

Procedure Task Timing		
Task	**Seconds**	**Percent**
Reading and Levelizing Data	0.09	0.01%
Communication to Client	0.21	0.02%
Bayesian Analysis: Likelihood for MCMC	942.12	99.82%
Bayesian Analysis: MCMC	0.21	0.02%
Optimization	1.24	0.13%
Post-optimization	0.00	0.00%

References

Aigner, C., Lovell, C. A. K., and Schmidt, P. (1977). "Formulation and Estimation of Stochastic Frontier Production Function Models." *Journal of Econometrics* 6:21–37.

Battese, G. E., and Coelli, T. J. (1988). "Prediction of Firm-Level Technical Efficiencies with a Generalized Frontier Production Function and Panel Data." *Journal of Econometrics* 38:387–399.

Christensen, L. R., and Greene, W. H. (1976). "Economies of Scale in U.S. Electric Power Generation." *Journal of Political Economy* 84:655–676.

Coelli, T. J., Prasada Rao, D. S., and Battese, G. E. (1998). *An Introduction to Efficiency and Productivity Analysis*. London: Kluwer Academic.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004). *Bayesian Data Analysis*. 2nd ed. London: Chapman & Hall.

Jondrow, J., Lovell, C. A. K., Materov, I. S., and Schmidt, P. (1982). "On the Estimation of Technical Efficiency in the Stochastic Frontier Production Function Model." *Journal of Econometrics* 19:233–238.

Kumbhakar, S. C., and Lovell, C. A. K. (2000). *Stochastic Frontier Analysis*. New York: Cambridge University Press.

McKelvey, R. D., and Zavoina, W. (1975). "A Statistical Model for the Analysis of Ordinal Level Dependent Variables." *Journal of Mathematical Sociology* 4:103–120.

Meeusen, W., and van den Broeck, J. (1977). "Efficiency Estimation from Cobb-Douglas Production Functions with Composed Error." *International Economic Review* 18:435–444.

Mroz, T. A. (1987). "The Sensitivity of an Empirical Model of Married Women's Work to Economic and Statistical Assumptions." *Econometrica* 55:765–799.

Schervish, M. J. (1995). *Theory of Statistics*. New York: Springer-Verlag.

Wooldridge, J. M. (2002). *Econometric Analysis of Cross Section and Panel Data*. Cambridge, MA: MIT Press.

Subject Index

Syntax Index