THE POWER TO KNOW.

SAS/ETS® 15.1 User's Guide
The VARMAX Procedure

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

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

SAS/ETS® 15.1 User's Guide

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

November 2018

Chapter 42
The VARMAX Procedure

Contents

Overview: VARMAX Procedure

Given a multivariate time series, the VARMAX procedure estimates the model parameters and generates forecasts that are associated with vector autoregressive moving average processes with exogenous regressors (VARMAX) models. Often, economic or financial variables are not only contemporaneously correlated with each other, but also correlated with each other's past values. You can use the VARMAX procedure to model these types of time relationships. In many economic and financial applications, the variables of interest (dependent, response, or endogenous variables) are influenced by variables external to the system under consideration (independent, input, predictor, regressor, or exogenous variables). The VARMAX procedure enables you to model the dynamic relationships both among the dependent variables and between the dependent and independent variables.

A VARMAX model is defined in terms of the orders of the autoregressive or moving average processes (or both). When you use the VARMAX procedure, these orders can be specified by options or they can be automatically determined according to the information criteria. The VARMAX procedure supports the following information criteria: Akaike's information criterion (AIC), the corrected AIC (AICC), the Hannan-Quinn criterion (HQC), the final prediction error (FPE), and the Schwarz Bayesian criterion (SBC), which is

also known as the Bayesian information criterion (BIC). For the definitions and usages of the information criteria, see the section "The Minimum Information Criterion (MINIC) Method" on page 3066.

If you do not want to use automatic order selection, the VARMAX procedure provides the following autoregressive order identification aids: partial cross-correlations, partial autoregressive coefficients, partial canonical correlations, and Yule-Walker estimates.

For situations where the stationarity of the time series is in question, the VARMAX procedure provides the following tests to aid in determining the presence of unit roots and cointegration: Dickey-Fuller tests, the Stock-Watson common trends test for the possibility of cointegration among nonstationary vector processes of integrated order one, and Johansen cointegration tests for nonstationary vector processes of integrated order one and order two.

For stationary vector times series or nonstationary series that are made stationary by appropriate differencing or cointegration, the VARMAX procedure provides the vector autoregressive and moving average (VARMA) model and the vector error correction model (VECM). The vector error correction model can be in both autoregressive (AR) and autoregressive and moving average (ARMA) forms.

To cope with the problem of high dimensionality in the parameters of the VAR model and the VECM, the VARMAX procedure provides both the Bayesian vector autoregressive (BVAR) model and the Bayesian vector error correction model (BVECM). Bayesian models are used when prior information about the model parameters is available.

The VARMAX procedure also allows independent (exogenous) variables and their distributed lags to influence dependent (endogenous) variables in various models. These models are identified by an X suffix added to the original model name; for example, VARMAX, VECMX, BVARX, and BVECMX.

Correlations in the second moments of the vector time series might exist; this is called conditional heteroscedasticity. The VARMAX procedure supports three forms of multivariate generalized autoregressive conditional heteroscedasticity (GARCH) models to model the conditional heteroscedasticity: the Baba-Engle-Kroner-Kraft (BEKK) GARCH model, the constant conditional correlation (CCC) GARCH model, and the dynamic conditional correlation (DCC) GARCH model. For CCC and DCC GARCH models, five subforms of univariate GARCH models are supported: the GARCH model, the exponential GARCH (EGARCH) model, the quadratic GARCH (QGARCH) model, the threshold GARCH (TGARCH) model, and the power GARCH (PGARCH) model.

You can use the VARMAX-GARCH model or the VEC-ARMAX-GARCH model to simultaneously model both the first and second moments of the time series.

Finally, for stationary time series exhibiting long-range dependence (also known as long memory or persistence), that is series with a slowly decaying sample autocorrelation function, the VARMAX procedure supports the VARFIMA (vector autoregressive fractionally integrated moving average) and VARFIMAX models.

Forecasting is one of the main objectives of multivariate time series analysis. After successfully fitting the VARMAX, BVARX, VECMX, BVECMX, VARFIMAX and multivariate GARCH models, the VARMAX procedure computes predicted values and conditional heteroscedasticity based on the parameter estimates and the past values of the vector time series. Out-of-sample multistep-ahead forecasts are also supported. Simulation-based conditional forecasts and scenario analysis are supported for the VAR, BVAR, VECM, and BVECM models with or without the exogenous variables.

The following model parameter estimation methods are supported:

- the least squares (LS) method, which can be applied to VARX models

- the maximum likelihood (ML) method, which can be applied to all types of models and is used by default for VARFIMAX models,
- the conditional maximum likelihood (CML) method, which can be applied to VARMAX models

When you use the ML or CML method, you can start your optimization with the default or with different initial parameter values.

The VARMAX procedure supports the estimation of the restricted model when you impose linear constraints on the parameters of interest. The VARMAX procedure also supports various hypothesis tests of long-run effects and adjustment coefficients by using the likelihood ratio test that is based on Johansen cointegration analysis. It also supports the likelihood ratio test of weak exogeneity for each variable. In fact, because the VARMAX procedure outputs log-likelihood values for all models, you can always use the likelihood ratio test to check any linear hypothesis on parameters that are estimated in the models by estimating the restricted and unrestricted models separately. The VARMAX procedure also supports another alternative test, the Wald test.

After fitting the model parameters, the VARMAX procedure uses the following tests to provide model checks and residual analysis: Durbin-Watson (DW) statistics, the F test for autoregressive conditional heteroscedastic (ARCH) disturbance, the F test for AR disturbance, the Jarque-Bera normality test, and the portmanteau test.

The VARMAX procedure supports several modeling features, including seasonal deterministic terms, linear and quadratic time trends, subset models, multiple regression with distributed lags, the dead-start model (which does not have present values of the exogenous variables), and so on.

The VARMAX procedure provides a Granger causality test to determine the Granger-causal relationships between two distinct groups of variables. It also provides the following: the infinite order AR representation, the impulse response function (also called infinite order MA representation), the decomposition of the predicted error covariances, roots of the characteristic functions for both the AR and MA parts to evaluate the proximity of the roots to the unit circle, and contemporaneous relationships among the components of the vector time series.

Getting Started: VARMAX Procedure

This section provides several examples of the types of models that the VARMAX procedure supports.

Vector Autoregressive Model

Let $\mathbf{y}_t = (y_{1t}, \ldots, y_{kt})'$, $t = 1, 2, \ldots$, denote a k-dimensional time series vector of random variables of interest. The pth-order VAR process is written as

$$\mathbf{y}_t = \boldsymbol{\delta} + \Phi_1 \mathbf{y}_{t-1} + \cdots + \Phi_p \mathbf{y}_{t-p} + \boldsymbol{\epsilon}_t$$

where $\boldsymbol{\epsilon}_t = (\epsilon_{1t}, \ldots, \epsilon_{kt})'$ is a vector white noise process such that $\mathrm{E}(\boldsymbol{\epsilon}_t) = 0$, $\mathrm{E}(\boldsymbol{\epsilon}_t \boldsymbol{\epsilon}_t') = \Sigma$, and $\mathrm{E}(\boldsymbol{\epsilon}_t \boldsymbol{\epsilon}_s') = 0$ for $t \neq s$; $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_k)'$ is a constant vector; and Φ_i is a $k \times k$ matrix.

Analyzing and modeling the series jointly enables you to understand the dynamic relationships over time among the series and to improve the accuracy of forecasts for individual series by using the additional information available from the related series and their forecasts.

Consider the first-order stationary bivariate vector autoregressive model:

$$\mathbf{y}_t = \begin{pmatrix} 1.2 & -0.5 \\ 0.6 & 0.3 \end{pmatrix} \mathbf{y}_{t-1} + \boldsymbol{\epsilon}_t, \quad \text{with } \Sigma = \begin{pmatrix} 1.0 & 0.5 \\ 0.5 & 1.25 \end{pmatrix}$$

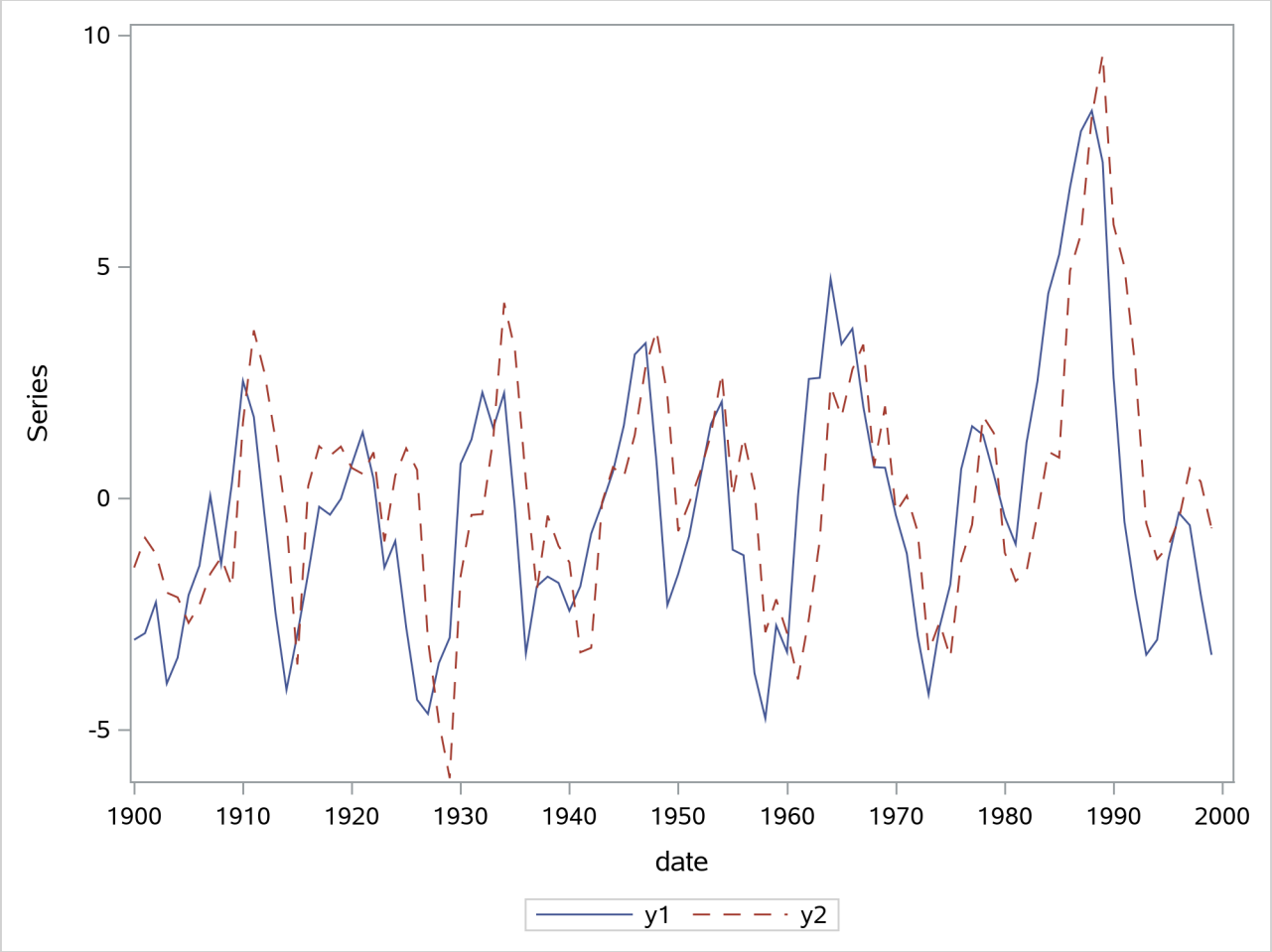
The following IML procedure statements simulate a bivariate vector time series from this model to provide test data for the VARMAX procedure:

```
proc iml;
   sig = {1.0  0.5, 0.5 1.25};
   phi = {1.2 -0.5, 0.6 0.3};
   /* simulate the vector time series */
   call varmasim(y,phi) sigma = sig n = 100 seed = 34657;
   cn = {'y1' 'y2'};
   create simul1 from y[colname=cn];
   append from y;
quit;
```

The following statements plot the simulated vector time series $\mathbf{y}_t$, which is shown in Figure 42.1:

```
data simul1;
   set simul1;
   date = intnx( 'year', '01jan1900'd, _n_-1 );
   format date year4.;
run;

proc sgplot data=simul1;
   series x=date y=y1 / lineattrs=(pattern=solid);
   series x=date y=y2 / lineattrs=(pattern=dash);
   yaxis label="Series";
run;
```

Figure 42.1 Plot of the Generated Data Process



The following statements fit a VAR(1) model to the simulated data:

```
/*--- Vector Autoregressive Model ---*/

proc varmax data=simul1;
   id date interval=year;
   model y1 y2 / p=1 noint lagmax=3
                 print=(estimates diagnose);
   output out=for lead=5;
run;
```

First, you specify the input data set in the PROC VARMAX statement. Then, you use the MODEL statement to designate the dependent variables, y_1 and y_2. To estimate a zero-mean VAR model, you specify the order of the autoregressive model in the P= option and the NOINT option. The MODEL statement fits the model to the data and prints parameter estimates and their significance. The PRINT=ESTIMATES option prints the matrix form of parameter estimates, and the PRINT=DIAGNOSE option prints various diagnostic tests. The LAGMAX=3 option prints the output for the residual diagnostic checks.

To output the forecasts to a data set, you specify the OUT= option in the OUTPUT statement. If you want to forecast five steps ahead, you use the LEAD=5 option. The ID statement specifies the yearly interval between observations and provides the Time column in the forecast output.

The VARMAX procedure output is shown in Figure 42.2 through Figure 42.10. The VARMAX procedure first displays descriptive statistics, as shown in Figure 42.2. The Type column indicates that the variables are dependent variables. The N column indicates the number of nonmissing observations.

Figure 42.2 Descriptive Statistics

The VARMAX Procedure

Number of Observations	100
Number of Pairwise Missing	0

Simple Summary Statistics

Variable	Type	N	Mean	Standard Deviation	Min	Max
y1	**Dependent**	100	-0.21653	2.78210	-4.75826	8.37032
y2	**Dependent**	100	0.16905	2.58184	-6.04718	9.58487

Figure 42.3 shows the model type and the estimation method that is used to fit the model to the simulated data. It also shows the AR coefficient matrix in terms of lag 1, the schematic representation, and the parameter estimates and their significance that can indicate how well the model fits the data.

The "AR" table shows the AR coefficient matrix. The "Schematic Representation" table schematically represents the parameter estimates and enables you to easily verify their significance in matrix form.

In the "Model Parameter Estimates" table, the first column shows the variable on the left side of the equation; the second column is the parameter name ARl_i_j, which indicates the (i, j) element of the lag l autoregressive coefficient; the next four columns provide the estimate, standard error, t value, and p-value for the parameter; and the last column is the regressor that corresponds to the displayed parameter.

Figure 42.3 Model Type and Parameter Estimates

The VARMAX Procedure

Type of Model	VAR(1)
Estimation Method	Least Squares Estimation

		AR	
Lag	**Variable**	**y1**	**y2**
1	**y1**	1.15977	-0.51058
	y2	0.54634	0.38499

Schematic Representation	
Variable/Lag	**AR1**
y1	+-
y2	++

+ is > 2*std error, - is < -2*std error, . is between, * is N/A

Model Parameter Estimates

Equation	**Parameter**	**Estimate**	**Standard Error**	**t Value**	**Pr > |t|**	**Variable**
y1	**AR1_1_1**	1.15977	0.05508	21.06	0.0001	y1(t-1)
	AR1_1_2	-0.51058	0.05898	-8.66	0.0001	y2(t-1)
y2	**AR1_2_1**	0.54634	0.05779	9.45	0.0001	y1(t-1)
	AR1_2_2	0.38499	0.06188	6.22	0.0001	y2(t-1)

The fitted VAR(1) model with estimated standard errors in parentheses is given as

$$\mathbf{y}_t = \begin{pmatrix} 1.160 & -0.511 \\ (0.055) & (0.059) \\ 0.546 & 0.385 \\ (0.058) & (0.062) \end{pmatrix} \mathbf{y}_{t-1} + \boldsymbol{\epsilon}_t$$

Clearly, all parameter estimates in the coefficient matrix Φ_1 are significant.

The model can also be written as two univariate regression equations:

$$\begin{aligned} y_{1t} &= 1.160\, y_{1,t-1} - 0.511\, y_{2,t-1} + \epsilon_{1t} \\ y_{2t} &= 0.546\, y_{1,t-1} + 0.385\, y_{2,t-1} + \epsilon_{2t} \end{aligned}$$

The table in Figure 42.4 shows the innovation covariance matrix estimates, the log likelihood, and the various information criteria results. The variable names in the table for the innovation covariance matrix estimates $\hat{\Sigma}$ are printed for convenience: $y1$ means the innovation for $y1$, and $y2$ means the innovation

for $y2$. The log likelihood for a VAR model that is estimated through least squares method is defined as $-T(\log(|\hat{\Sigma}_{\mathrm{ML}}|) + k)/2$, where $T(= 100 - 1 = 99)$ is the sample size except the presample being skipped because of the AR lag order, $k(= 2)$ is the number of dependent variables, and $\hat{\Sigma}_{\mathrm{ML}}$ is the maximum likelihood estimate (MLE) of innovation covariance matrix. The matrix $\hat{\Sigma}_{\mathrm{ML}}$ is computed from the reported least squares estimate of the innovation covariance matrix, $\hat{\Sigma}$, by adjusting the degrees of freedom. $\hat{\Sigma}_{\mathrm{ML}} = \frac{T-r_b}{T}\hat{\Sigma}$, where $r_b(= 2)$ is the number of parameters in each equation. You can use the information criteria to compare the fit of competing models to a set of data. The model that has a smaller value of the information criterion is preferred when it is compared to other models. For more information about how to calculate the information criteria, see the section "Multivariate Model Diagnostic Checks" on page 3085.

Figure 42.4 Innovation Covariance Estimates, Log Likelihood, and Information Criteria

Covariances of Innovations

Variable	y1	y2
y1	1.28875	0.39751
y2	0.39751	1.41839

Log-likelihood	-122.362

Information Criteria

AICC	259.9557
HQC	266.0748
AIC	258.7249
SBC	276.8908
FPEC	1.738092

Figure 42.5 shows the cross covariances of the residuals. The values of the lag 0 are slightly different from Figure 42.4 because of the different degrees of freedom.

Figure 42.5 Multivariate Diagnostic Checks

Cross Covariances of Residuals

Lag	Variable	y1	y2
0	y1	1.26271	0.38948
	y2	0.38948	1.38974
1	y1	0.03121	0.05675
	y2	-0.04646	-0.05398
2	y1	0.08134	0.10599
	y2	0.03482	-0.01549
3	y1	0.01644	0.11734
	y2	0.00609	0.11414

Figure 42.6 and Figure 42.7 show tests for white noise residuals that are based on the cross correlations of the residuals. The output shows that you cannot reject the null hypothesis that the residuals are uncorrelated.

Figure 42.6 Multivariate Diagnostic Checks, Continued

Cross Correlations of Residuals			
Lag	Variable	y1	y2
0	y1	1.00000	0.29401
	y2	0.29401	1.00000
1	y1	0.02472	0.04284
	y2	-0.03507	-0.03884
2	y1	0.06442	0.08001
	y2	0.02628	-0.01115
3	y1	0.01302	0.08858
	y2	0.00460	0.08213

Schematic Representation of Cross Correlations of Residuals				
Variable/Lag	0	1	2	3
y1	++	. .	. .	. .
y2	++	. .	. .	. .
+ is > 2*std error, - is < -2*std error, . is between				

Figure 42.7 Multivariate Diagnostic Checks, Continued

Portmanteau Test for Cross Correlations of Residuals			
Up To Lag	DF	Chi-Square	Pr > ChiSq
2	4	1.58	0.8124
3	8	2.78	0.9473

The VARMAX procedure provides diagnostic checks for the univariate form of the equations. The table in Figure 42.8 describes how well each univariate equation fits the data. From the two univariate regression equations shown in Figure 42.3, the values of R^2 in the second column of Figure 42.8 are 0.84 and 0.79. The standard deviations in the third column are the square roots of the diagonal elements of the covariance matrix from Figure 42.4. The F statistics in the fourth column test the null hypotheses $\phi_{11} = \phi_{12} = 0$ and $\phi_{21} = \phi_{22} = 0$, where ϕ_{ij} is the (i, j) element of the matrix Φ_1. The last column shows the p-values of the F statistics. The results show that each univariate model is significant.

Figure 42.8 Univariate Diagnostic Checks

Univariate Model ANOVA Diagnostics				
Variable	R-Square	Standard Deviation	F Value	Pr > F
y1	0.8351	1.13523	491.25	<.0001
y2	0.7906	1.19096	366.29	<.0001

The check for white noise residuals in terms of the univariate equation is shown in Figure 42.9. This output contains information that indicates whether the residuals are correlated and heteroscedastic. In the first table, the second column contains the Durbin-Watson test statistics to test the null hypothesis that the residuals are

uncorrelated. The third and fourth columns show the Jarque-Bera normality test statistics and their *p*-values to test the null hypothesis that the residuals have normality. The last two columns show *F* statistics and their *p*-values for ARCH(1) disturbances to test the null hypothesis that the residuals have equal covariances. The second table includes *F* statistics and their *p*-values for AR(1), AR(1,2), AR(1,2,3) and AR(1,2,3,4) models of residuals to test the null hypothesis that the residuals are uncorrelated.

Figure 42.9 Univariate Diagnostic Checks, Continued

Univariate Model White Noise Diagnostics

		Normality		ARCH	
Variable	Durbin Watson	Chi-Square	Pr > ChiSq	F Value	Pr > F
y1	1.94534	3.56	0.1686	0.13	0.7199
y2	2.06276	5.42	0.0667	2.10	0.1503

Univariate Model AR Diagnostics

	AR1		AR2		AR3		AR4	
Variable	F Value	Pr > F	F Value	Pr > F	F Value	Pr > F	F Value	Pr > F
y1	0.02	0.8980	0.14	0.8662	0.09	0.9629	0.82	0.5164
y2	0.52	0.4709	0.41	0.6650	0.32	0.8136	0.32	0.8664

The table in Figure 42.10 shows forecasts, their prediction errors, and 95% confidence limits. For more information, see the section "Forecasting" on page 3057.

Figure 42.10 Forecasts

Forecasts

					95% Confidence Limits	
Variable	Obs	Time	Forecast	Standard Error		
y1	101	2000	-3.59212	1.13523	-5.81713	-1.36711
	102	2001	-3.09448	1.70915	-6.44435	0.25539
	103	2002	-2.17433	2.14472	-6.37792	2.02925
	104	2003	-1.11395	2.43166	-5.87992	3.65203
	105	2004	-0.14342	2.58740	-5.21463	4.92779
y2	101	2000	-2.09873	1.19096	-4.43298	0.23551
	102	2001	-2.77050	1.47666	-5.66469	0.12369
	103	2002	-2.75724	1.74212	-6.17173	0.65725
	104	2003	-2.24943	2.01925	-6.20709	1.70823
	105	2004	-1.47460	2.25169	-5.88782	2.93863

Bayesian Vector Autoregressive Model

The Bayesian vector autoregressive (BVAR) model avoids problems of collinearity and overparameterization that often occur with the use of VAR models. BVAR models avoid these problems by imposing priors on the AR parameters.

The following statements fit a BVAR(1) model to the simulated data:

```
/*--- Bayesian Vector Autoregressive Process ---*/

proc varmax data=simul1;
   model y1 y2 / p=1 noint
                 prior=(lambda=0.9 theta=0.1);
run;
```

The hyperparameters, LAMBDA=0.9 and THETA=0.1, in the PRIOR= option control the prior covariance. Part of the VARMAX procedure output is shown in Figure 42.11, whose parameter estimates are slightly different from those in Figure 42.3. By choosing the appropriate priors, you might be able to obtain more accurate forecasts by using a BVAR model instead of an unconstrained VAR model. For more information, see the section "Bayesian VAR and VARX Modeling" on page 3074.

Figure 42.11 Parameter Estimates for the BVAR(1) Model

The VARMAX Procedure

Type of Model	BVAR(1)
Estimation Method	Maximum Likelihood Estimation
Prior Lambda	0.9
Prior Theta	0.1

Model Parameter Estimates

Equation	Parameter	Estimate	Standard Error	t Value	Pr > |t|	Variable
y1	**AR1_1_1**	1.02312	0.04999	20.47	0.0001	y1(t-1)
	AR1_1_2	-0.32867	0.04807	-6.84	0.0001	y2(t-1)
y2	**AR1_2_1**	0.37863	0.04867	7.78	0.0001	y1(t-1)
	AR1_2_2	0.52911	0.05670	9.33	0.0001	y2(t-1)

Covariances of Innovations

Variable	y1	y2
y1	1.39090	0.50192
y2	0.50192	1.51456

Vector Error Correction Model

A vector error correction model (VECM) can lead to a better understanding of the nature of any nonstationarity among the different component series and can also improve longer-term forecasting compared to an unconstrained model.

The VECM(p) form with the cointegration rank, $r(\leq k)$, is written as

$$\Delta \mathbf{y}_t = \boldsymbol{\delta} + \Pi \mathbf{y}_{t-1} + \sum_{i=1}^{p-1} \Phi_i^* \Delta \mathbf{y}_{t-i} + \boldsymbol{\epsilon}_t$$

where Δ is the differencing operator, such that $\Delta \mathbf{y}_t = \mathbf{y}_t - \mathbf{y}_{t-1}$; $\Pi = \alpha\beta'$, where α and β are $k \times r$ matrices; and Φ_i^* is a $k \times k$ matrix.

The VECM(p) form has an equivalent VAR(p) representation as described in the section "Vector Autoregressive Model" on page 2954.

$$\mathbf{y}_t = \boldsymbol{\delta} + (I_k + \Pi + \Phi_1^*)\mathbf{y}_{t-1} + \sum_{i=2}^{p-1}(\Phi_i^* - \Phi_{i-1}^*)\mathbf{y}_{t-i} - \Phi_{p-1}^*\mathbf{y}_{t-p} + \boldsymbol{\epsilon}_t$$

where I_k is a $k \times k$ identity matrix.

An example of the second-order nonstationary vector autoregressive model is

$$\mathbf{y}_t = \begin{pmatrix} -0.2 & 0.1 \\ 0.5 & 0.2 \end{pmatrix}\mathbf{y}_{t-1} + \begin{pmatrix} 0.8 & 0.7 \\ -0.4 & 0.6 \end{pmatrix}\mathbf{y}_{t-2} + \boldsymbol{\epsilon}_t$$

with

$$\Sigma = \begin{pmatrix} 100 & 0 \\ 0 & 100 \end{pmatrix} \text{ and } \mathbf{y}_{-1} = \mathbf{y}_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

This process can be given the following VECM(2) representation with the cointegration rank one:

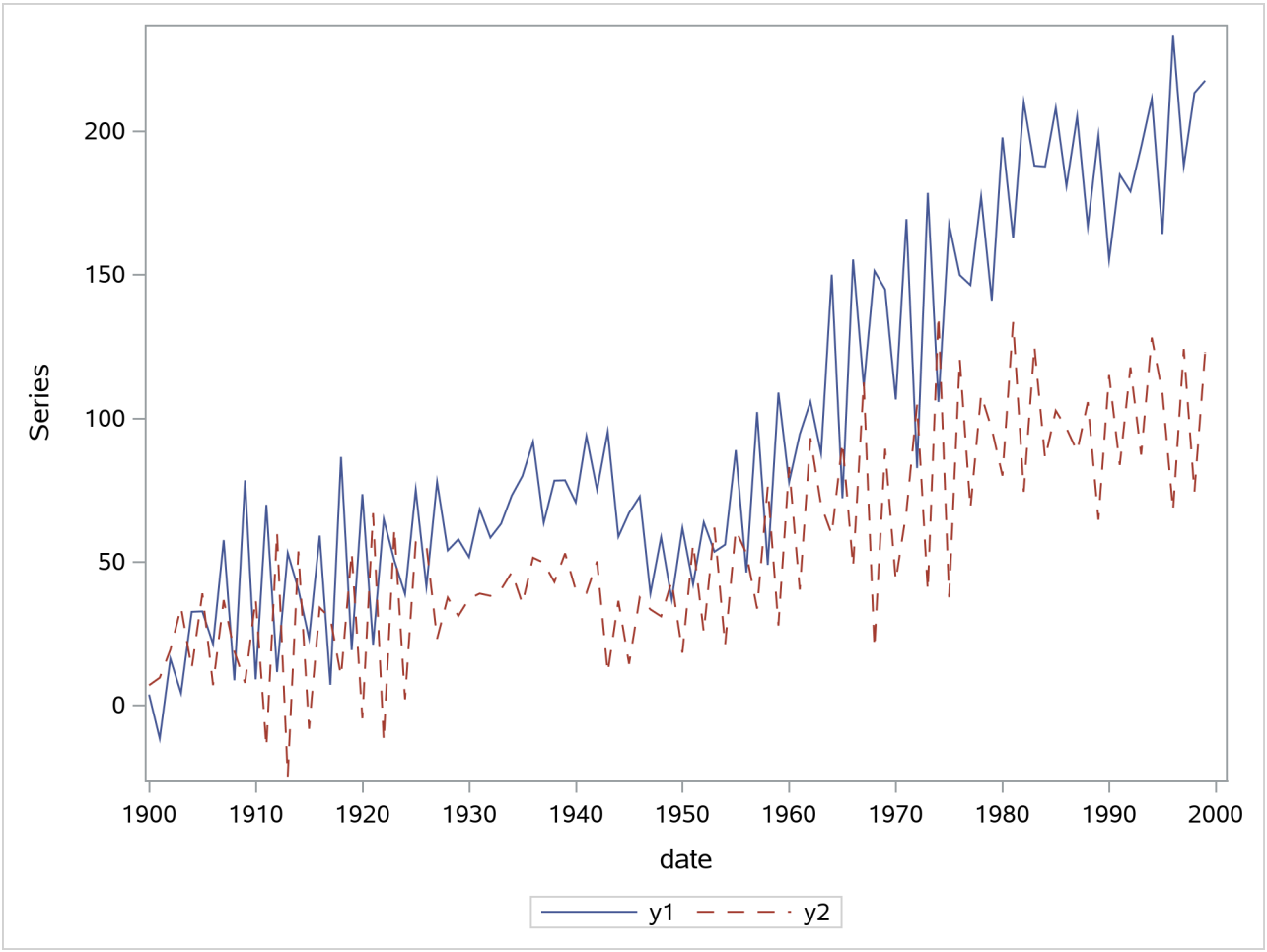
$$\Delta\mathbf{y}_t = \begin{pmatrix} -0.4 \\ 0.1 \end{pmatrix}(1, -2)\mathbf{y}_{t-1} - \begin{pmatrix} 0.8 & 0.7 \\ -0.4 & 0.6 \end{pmatrix}\Delta\mathbf{y}_{t-1} + \boldsymbol{\epsilon}_t$$

The following PROC IML statements generate simulated data for this VECM(2) form and the PROC SGPLOT statements plot the data, as shown in Figure 42.12:

```
proc iml;
   sig = 100*i(2);
   phi = {-0.2 0.1, 0.5 0.2, 0.8 0.7, -0.4 0.6};
   call varmasim(y,phi) sigma=sig n=100 initial=0
                        seed=45876;
   cn = {'y1' 'y2'};
   create simul2 from y[colname=cn];
   append from y;
quit;

data simul2;
```

```
   set simul2;
   date = intnx( 'year', '01jan1900'd, _n_-1 );
   format date year4. ;
run;

proc sgplot data=simul2;
   series x=date y=y1 / lineattrs=(pattern=solid);
   series x=date y=y2 / lineattrs=(pattern=dash);
   yaxis label="Series";
run;
```

Figure 42.12 Plot of Generated Data Process



Cointegration Testing

The following statements use the Johansen cointegration rank test. The COINTTEST=(JOHANSEN) option performs the Johansen trace test and is equivalent to specifying the COINTTEST option with no additional suboptions or specifying the COINTTEST=(JOHANSEN=(TYPE=TRACE)) option.

```
/*--- Cointegration Test ---*/

proc varmax data=simul2;
   model y1 y2 / p=2 noint dftest cointtest=(johansen);
run;
```

Figure 42.13 shows the output for Dickey-Fuller tests for the nonstationarity of each series and the Johansen cointegration rank test between series.

Figure 42.13 Dickey-Fuller Tests and Cointegration Rank Test

The VARMAX Procedure

Unit Root Test

Variable	Type	Rho	Pr < Rho	Tau	Pr < Tau
y1	Zero Mean	1.47	0.9628	1.65	0.9755
	Single Mean	-0.80	0.9016	-0.47	0.8916
	Trend	-10.88	0.3573	-2.20	0.4815
y2	Zero Mean	-0.05	0.6692	-0.03	0.6707
	Single Mean	-6.03	0.3358	-1.72	0.4204
	Trend	-50.49	0.0003	-4.92	0.0006

Cointegration Rank Test Using Trace

H0: Rank=r	H1: Rank>r	Eigenvalue	Trace	Pr > Trace	Drift in ECM	Drift in Process
0	0	0.5086	70.7279	<.0001	NOINT	Constant
1	1	0.0111	1.0921	0.3441		

In Dickey-Fuller tests, the second column specifies three types of models, which are zero mean, single mean, or trend. The third column (Rho) and the fifth column (Tau) are the test statistics that are used to test the null hypothesis that the series has a unit root. Other columns are their p-values. You can see that both series have unit roots. For a description of Dickey-Fuller tests, see the section "PROBDF Function for Dickey-Fuller Tests" on page 157 in Chapter 5, "SAS Macros and Functions."

In the "Cointegration Rank Test Using Trace" table, the last two columns explain the drift in the model or process. Because the NOINT option is specified, the model is

$$\Delta \mathbf{y}_t = \Pi \mathbf{y}_{t-1} + \Phi_1^* \Delta \mathbf{y}_{t-1} + \boldsymbol{\epsilon}_t$$

The column Drift in ECM indicates that there is no separate drift in the error correction model, and the column Drift in Process indicates that the process has a constant drift before differencing.

H0 is the null hypothesis, and H1 is the alternative hypothesis. The first row tests the cointegration rank $r = 0$ against $r > 0$, and the second row tests $r = 1$ against $r > 1$. The trace test statistics in the fourth column are computed by $-T \sum_{i=r+1}^{k} \log(1 - \lambda_i)$, where T is the available number of observations and λ_i is the eigenvalue in the third column. The p-values for these statistics are output in the fifth column. If you compare the p-value in each row to the significance level of interest (such as 5%), the null hypothesis that there is no cointegrated process (H0: $r = 0$) is rejected, whereas the null hypothesis that there is at most one cointegrated process (H0: $r = 1$) cannot be rejected.

The following statements fit a VECM(2) form to the simulated data:

```
/*--- Vector Error Correction Model ---*/

proc varmax data=simul2;
   model y1 y2 / p=2 noint lagmax=3
                 print=(iarr estimates);
   cointeg rank=1 normalize=y1;
run;
```

The results in Figure 42.13 indicate that the time series are cointegrated with rank = 1. So you might want to specify the RANK=1 option in the COINTEG statement. For normalizing the value of the cointegrated vector, you specify the normalized variable by using the NORMALIZE= option in the COINTEG statement. The COINTEG statement produces the estimates of the long-run parameter, $\boldsymbol{\beta}$, and the adjustment coefficient, $\boldsymbol{\alpha}$. The PRINT=(IARR) option provides the VAR(2) representation.

The VARMAX procedure output is shown in Figure 42.14 through Figure 42.17. In Figure 42.14, "1" indicates the first column of the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ matrices. Because the cointegration rank is 1 in the bivariate system, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are two-dimensional vectors. The estimated cointegrating vector is $\hat{\boldsymbol{\beta}} = (1, -1.96)'$. Therefore, the long-run relationship between y_{1t} and y_{2t} is $y_{1t} = 1.96y_{2t}$. The first element of $\hat{\boldsymbol{\beta}}$ is 1 because y_1 is specified as the normalized variable. Asymptotically, $\boldsymbol{\alpha}$ follows a normal distribution, and the t values and p-values of its elements are shown in the "Alpha and Beta Parameter Estimates" table; however, because $\boldsymbol{\beta}$ follows a nonnormal distribution, the corresponding standard errors, t values, and p-values are missing. The Variable column shows the variables that correspond to the coefficients. For example, for the coefficient $\boldsymbol{\alpha}_{ij}$ (the ith element in the jth column of $\boldsymbol{\alpha}$), ALPHAi_j, the variable is the inner product of the transpose of the jth column of $\boldsymbol{\beta}$ (Beta[,j]$'$) and the vector of lag 1 dependent variables $\mathbf{y}_{t-1}$ (_DEP_(t–1)).

Figure 42.14 Parameter Estimates for the VECM(2) Form

The VARMAX Procedure

Type of Model	VECM(2)
Estimation Method	Maximum Likelihood Estimation
Cointegrated Rank	1

Beta	
Variable	**1**
y1	1.00000
y2	-1.95575

Alpha	
Variable	**1**
y1	-0.46680
y2	0.10667

Alpha and Beta Parameter Estimates

Equation	Parameter	Estimate	Standard Error	t Value	Pr > |t|	Variable
D_y1	**ALPHA1_1**	-0.46680	0.04786	-9.75	<.0001	Beta[,1]'*_DEP_(t-1)
	BETA1_1	1.00000				y1(t-1)
D_y2	**ALPHA2_1**	0.10667	0.05146	2.07	0.0409	Beta[,1]'*_DEP_(t-1)
	BETA2_1	-1.95575				y2(t-1)

Figure 42.15 shows the parameter estimates in terms of lag 1 coefficients, $\mathbf{y}_{t-1}$, and lag 1 first-differenced coefficients, $\Delta\mathbf{y}_{t-1}$, and their significance. "Alpha * Beta′" indicates the coefficients of $\mathbf{y}_{t-1}$ and is obtained by multiplying the Alpha and Beta estimates in Figure 42.14. The parameter AR1_i_j (which is shown in the "Model Parameter Estimates" table) corresponds to the elements in the "Alpha * Beta′" matrix. The parameter AR2_i_j corresponds to the elements in the differenced lagged AR coefficient matrix. The "D_" prefixed to a variable name in Figure 42.15 implies differencing.

Figure 42.15 Parameter Estimates for the VECM(2) Form, Continued

Parameter Alpha * Beta' Estimates

Variable	y1	y2
y1	-0.46680	0.91295
y2	0.10667	-0.20862

AR Coefficients of Differenced Lag

DIF Lag	Variable	y1	y2
1	y1	-0.74332	-0.74621
	y2	0.40493	-0.57157

Model Parameter Estimates

Equation	Parameter	Estimate	Standard Error	t Value	Pr > |t|	Variable
D_y1	AR1_1_1	-0.46680	0.04786	-9.75	<.0001	y1(t-1)
	AR1_1_2	0.91295	0.09359	9.75	<.0001	y2(t-1)
	AR2_1_1	-0.74332	0.04526	-16.42	<.0001	D_y1(t-1)
	AR2_1_2	-0.74621	0.04769	-15.65	<.0001	D_y2(t-1)
D_y2	AR1_2_1	0.10667	0.05146	2.07	0.0409	y1(t-1)
	AR1_2_2	-0.20862	0.10064	-2.07	0.0409	y2(t-1)
	AR2_2_1	0.40493	0.04867	8.32	<.0001	D_y1(t-1)
	AR2_2_2	-0.57157	0.05128	-11.15	<.0001	D_y2(t-1)

Figure 42.16 shows the parameter estimates of the innovations covariance matrix and their significance.

Figure 42.16 Parameter Estimates for the VECM(2) Form, Continued

Covariance Parameter Estimates

Parameter	Estimate	Standard Error	t Value	Pr > |t|
COV1_1	94.75575	13.53654	7.00	<.0001
COV1_2	4.52684	10.30302	0.44	0.6614
COV2_2	109.57038	15.65291	7.00	<.0001

The fitted model is represented as

$$\Delta \mathbf{y}_t = \begin{pmatrix} -0.467 & 0.913 \\ (0.048) & (0.094) \\ 0.107 & -0.209 \\ (0.051) & (0.100) \end{pmatrix} \mathbf{y}_{t-1} + \begin{pmatrix} -0.743 & -0.746 \\ (0.045) & (0.048) \\ 0.405 & -0.572 \\ (0.049) & (0.051) \end{pmatrix} \Delta \mathbf{y}_{t-1} + \boldsymbol{\epsilon}_t$$

Figure 42.17 Change the VECM(2) Form to the VAR(2) Model

Infinite Order AR Representation			
Lag	Variable	y1	y2
1	y1	-0.21013	0.16674
	y2	0.51160	0.21980
2	y1	0.74332	0.74621
	y2	-0.40493	0.57157
3	y1	0.00000	0.00000
	y2	0.00000	0.00000

The PRINT=(IARR) option in the previous SAS statements prints the reparameterized coefficient estimates. Because LAGMAX=3 in those statements, the coefficient matrix of lag 3 is zero.

The VECM(2) form in Figure 42.17 can be rewritten as the following second-order vector autoregressive model:

$$\mathbf{y}_t = \begin{pmatrix} -0.210 & 0.167 \\ 0.512 & 0.220 \end{pmatrix} \mathbf{y}_{t-1} + \begin{pmatrix} 0.743 & 0.746 \\ -0.405 & 0.572 \end{pmatrix} \mathbf{y}_{t-2} + \boldsymbol{\epsilon}_t$$

Bayesian Vector Error Correction Model

Bayesian inference on a cointegrated system begins by using the priors of $\boldsymbol{\beta}$, which are obtained from the VECM(p) form. Bayesian vector error correction models can improve forecast accuracy for cointegrated processes.

To use a Bayesian vector error correction model, you specify both the PRIOR= option in the MODEL statement and the COINTEG statement. The following statements fit a BVECM(2) form to the simulated data:

```
/*--- Bayesian Vector Error Correction Model ---*/

proc varmax data=simul2;
   model y1 y2 / p=2 noint
                 prior=( lambda=0.5 theta=0.2 )
                 print=(estimates);
   cointeg rank=1 normalize=y1;
run;
```

The VARMAX procedure output in Figure 42.18 shows the model type fitted to the data, the estimates of the adjustment coefficient ($\boldsymbol{\alpha}$), the parameter estimates in terms of lag 1 coefficients ($\mathbf{y}_{t-1}$), and lag 1 first-differenced coefficients ($\Delta\mathbf{y}_{t-1}$).

Figure 42.18 Parameter Estimates for the BVECM(2) Form

The VARMAX Procedure

Type of Model	BVECM(2)
Estimation Method	Maximum Likelihood Estimation
Cointegrated Rank	1
Prior Lambda	0.5
Prior Theta	0.2

Alpha	
Variable	**1**
y1	-0.34173
y2	0.17202

Parameter Alpha * Beta' Estimates		
Variable	**y1**	**y2**
y1	-0.34173	0.66835
y2	0.17202	-0.33643

AR Coefficients of Differenced Lag			
DIF Lag	**Variable**	**y1**	**y2**
1	**y1**	-0.80345	-0.59201
	y2	0.33192	-0.52779

Vector Autoregressive Fractionally Integrated Moving Average Model

Fractionally integrated models can be used to model stationary time series whose sample autocorrelation function decays slowly at large positive and negative lags. This behavior is often referred to as long-range dependence (LRD), long memory, or persistence; series that exhibit such behavior are called long-range dependent (LRD).

A typical parametric model for a k-dimensional series $\mathbf{y}_t = (y_{1t}, \ldots, y_{kt})'$, $t = 1, \ldots, T$, whose individual components are LRD is the VARFIMA (vector autoregressive fractionally integrated moving average) model. It is obtained as a natural extension of the well-known class of ARFIMA models by fractionally integrating the individual components of a k-dimensional white noise series. For example, a bivariate VARFIMA$(0, D, 0)$ series with no intercept term is given by

$$\mathbf{y}_t = \begin{pmatrix} y_{1t} \\ y_{2t} \end{pmatrix} = \begin{pmatrix} (I-B)^{-d_1} & 0 \\ 0 & (I-B)^{-d_2} \end{pmatrix} \begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \end{pmatrix} = (I-B)^{-D}\boldsymbol{\epsilon}_t$$

where B is the backshift operator; $I = B^0$ is the identity operator; $d_1, d_2 \in (-1/2, 1/2)$ are the LRD parameters of the component series $\{y_{1t}\}_{t\in\mathrm{Z}}$ and $\{y_{2t}\}_{t\in\mathrm{Z}}$, respectively; $D = \mathrm{diag}(d_1, d_2)$; and $\{\boldsymbol{\epsilon}_t\}_{t\in\mathrm{Z}}=$

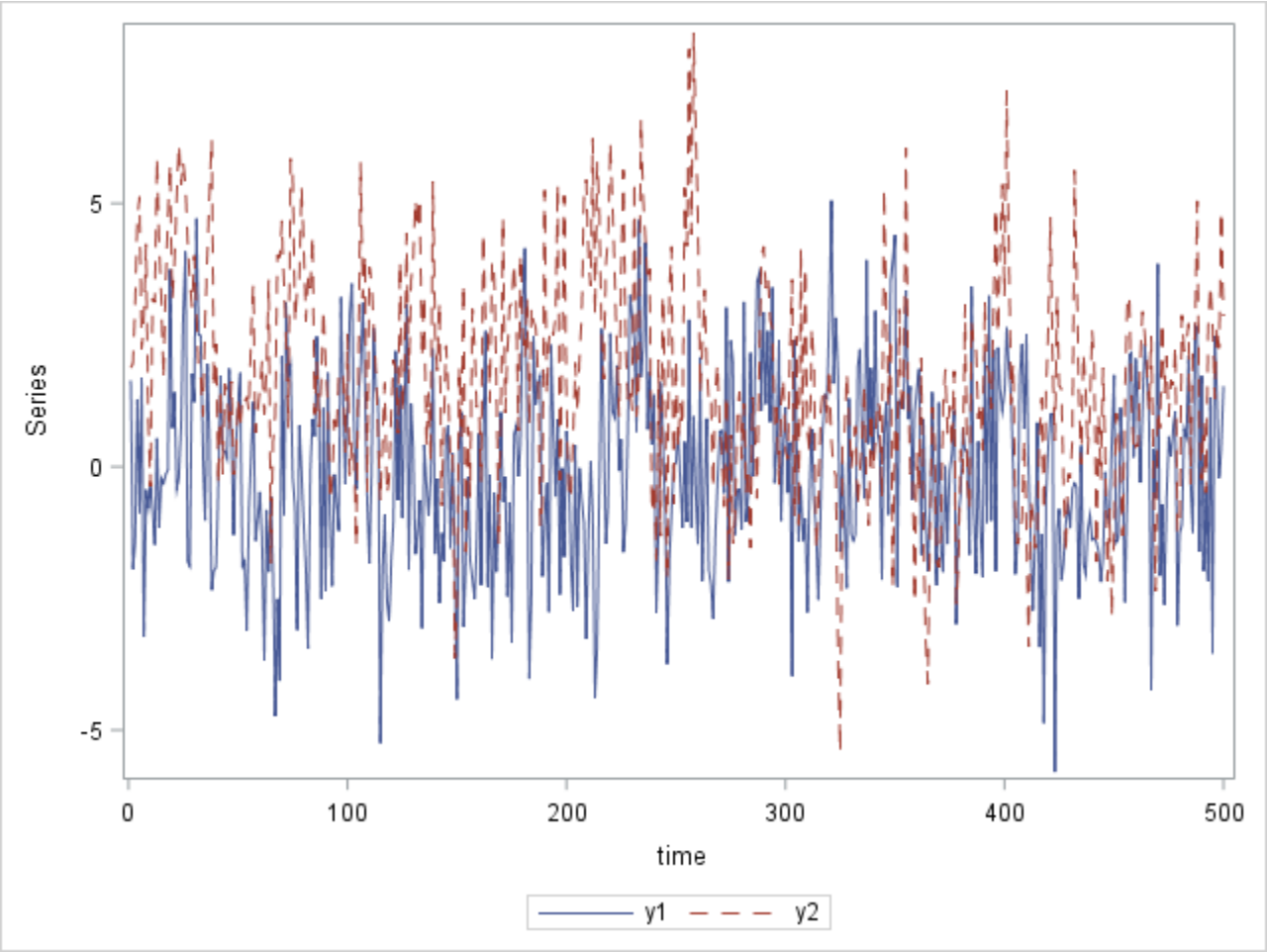
$\{(\epsilon_{1t}, \epsilon_{2t})'\}_{t \in \mathrm{Z}}$ is a bivariate white noise series indexed by the set of integers Z with zero mean $\mathrm{E}\boldsymbol{\epsilon}_t = 0$ and covariance $\mathrm{E}\boldsymbol{\epsilon}_t\boldsymbol{\epsilon}_t' = \Sigma$.

The multivariate VARFIMA model is defined analogously. The matrix Σ is in general nondiagonal, which enables the VARFIMA model to capture dependence between the individual series.

The following statements plot a simulated bivariate VARFIMA$(0, D, 0)$ series with $d_1 = 0.2$, $d_2 = 0.4$, and Gaussian errors with $\Sigma_{11} = \Sigma_{22} = 3$ and $\Sigma_{12} = 0.5$:

```
data VARFIMA0D0;
   time = _N_;
   input y1 y2;
datalines;
1.6380971 1.877144

   ... more lines ...

0.3482938 4.8601886
1.5320803 2.8687495
;

proc sgplot data = VARFIMA0D0;
   series x = time y=y1 / lineattrs=(pattern=solid);
   series x = time y=y2 / lineattrs=(pattern=dash);
   yaxis label="Series";
run;
```

Figure 42.19 Plot of the Data



Before fitting a VARFIMA model to a data set, you should plot the series' sample autocorrelation function to confirm its slow decay. It is also instructive to plot the periodogram of the series. In the presence of long memory, the periodogram explodes at frequencies near 0.

The following statements produce the periodogram and the sample autocorrelation function for the specified data:

```
ods graphics on;
proc timeseries data= VARFIMA0D0 plots = (periodogram acf);
   var y1 y2;
   spectra freq  / adjmean;
   corr / NLAG = 30;
run;
```

Figure 42.20 Sample Autocorrelation Functions of the Two Series

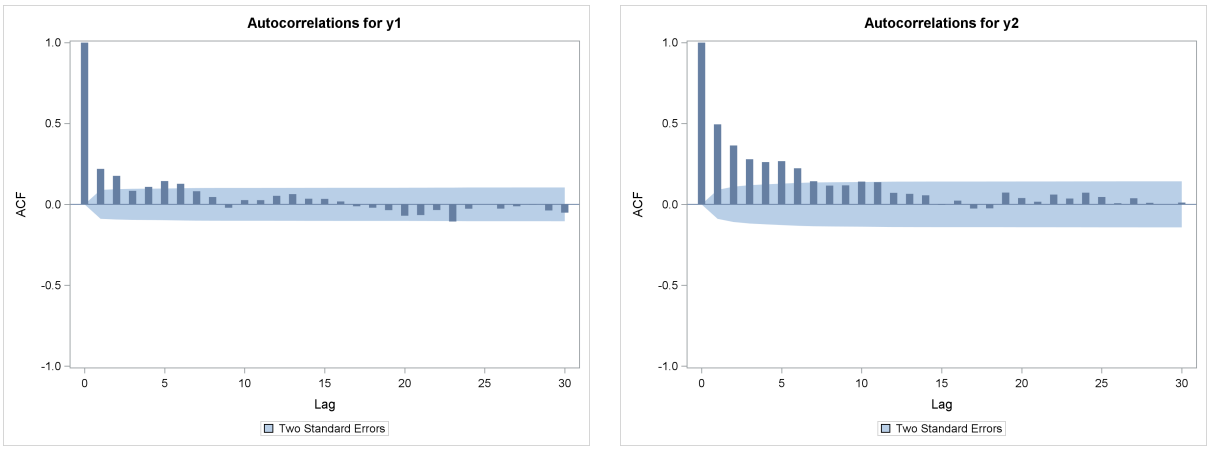
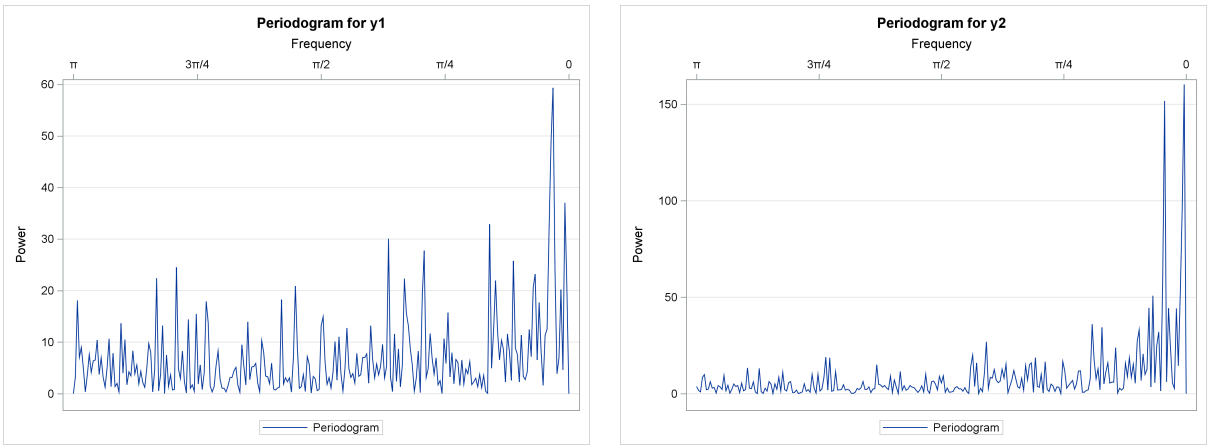


Figure 42.21 Periodograms of the Two Series



The magnitude of the LRD parameters d_1 and d_2 controls the memory of the two series. Series $y2$ has a larger LRD parameter than series $y1$ and hence is expected to exhibit longer memory. In the time domain, this effect is illustrated in Figure 42.20, where the autocorrelation function of series $y2$ (right plot in Figure 42.20) decays more slowly than the autocorrelation function of series $y1$ (left plot in Figure 42.20) with the increasing lag.

Figure 42.21 is the frequency domain analogue of Figure 42.20. In this case, the longer memory of series $y2$ is reflected by its periodogram (right plot in Figure 42.21), which blows up higher than the periodogram of series $y1$ (left plot in Figure 42.21) at frequencies near 0. Note the different scales used in the two plots.

The following statements fit the VARFIMA$(0, D, 0)$ model with no intercept term to the data. The FI option in the MODEL statement specifies fractional integration.

```
proc varmax data = VARFIMA0D0;
   model y1 y2 / fi noint method = ML;
run;
```

Figure 42.22 Parameter Estimates for the VARFIMA$(0, D, 0)$ Model

The VARMAX Procedure

Type of Model	VARFIMA(0,D,0)
Estimation Method	Maximum Likelihood Estimation

Model Parameter Estimates

Equation	Parameter	Estimate	Standard Error	t Value	Pr > |t|	Variable
y1	**D1**	0.20250	0.03555	5.70	0.0001	
y2	**D2**	0.38839	0.03053	12.72	0.0001	

Covariances of Innovations

Variable	y1	y2
y1	3.20607	0.48068
y2	0.48068	3.15651

The estimation method that PROC VARMAX uses by default for the VARFIMA series is maximum likelihood (for more information, see the section "VARFIMA and VARFIMAX Modeling" on page 3121). All five parameter are estimated close to their true value and are significant.

Vector Autoregressive Model with Exogenous Variables

A VAR process can be affected by other observable variables that are determined outside the system of interest. Such variables are called exogenous (independent) variables. Exogenous variables can be stochastic or nonstochastic. The process can also be affected by the lags of exogenous variables. A model used to describe this process is called a VARX(p,s) model.

The VARX(p,s) model is written as

$$\mathbf{y}_t = \boldsymbol{\delta} + \sum_{i=1}^{p} \Phi_i \mathbf{y}_{t-i} + \sum_{i=0}^{s} \Theta_i^* \mathbf{x}_{t-i} + \boldsymbol{\epsilon}_t$$

where $\mathbf{x}_t = (x_{1t}, \ldots, x_{rt})'$ is an r-dimensional time series vector and Θ_i^* is a $k \times r$ matrix.

For example, a VARX(1,0) model is

$$\mathbf{y}_t = \boldsymbol{\delta} + \Phi_1 \mathbf{y}_{t-1} + \Theta_0^* \mathbf{x}_t + \boldsymbol{\epsilon}_t$$

where $\mathbf{y}_t = (y_{1t}, y_{2t}, y_{3t})'$ and $\mathbf{x}_t = (x_{1t}, x_{2t})'$.

The following statements fit the VARX(1,0) model to the given data:

```
data grunfeld;
   input year y1 y2 y3 x1 x2 x3;
   label y1='Gross Investment GE'
         y2='Capital Stock Lagged GE'
         y3='Value of Outstanding Shares GE Lagged'
         x1='Gross Investment W'
         x2='Capital Stock Lagged W'
         x3='Value of Outstanding Shares Lagged W';
datalines;
1935  33.1 1170.6  97.8 12.93  191.5   1.8
1936  45.0 2015.8 104.4 25.90  516.0    .8
1937  77.2 2803.3 118.0 35.05  729.0   7.4
1938  44.6 2039.7 156.2 22.89  560.4  18.1

   ... more lines ...


/*--- Vector Autoregressive Process with Exogenous Variables ---*/

proc varmax data=grunfeld;
   model y1-y3 = x1 x2 / p=1 lagmax=5
                         printform=univariate
                         print=(impulsx=(all) estimates);
run;
```

The VARMAX procedure output is shown in Figure 42.23 through Figure 42.25.

Figure 42.23 shows the descriptive statistics for the dependent (endogenous) and independent (exogenous) variables with labels.

Figure 42.23 Descriptive Statistics for the VARX(1, 0) Model

The VARMAX Procedure

Number of Observations	20
Number of Pairwise Missing	0

Simple Summary Statistics

Variable	Type	N	Mean	Standard Deviation	Min	Max	Label
y1	**Dependent**	20	102.29000	48.58450	33.10000	189.60000	Gross Investment GE
y2	**Dependent**	20	1941.32500	413.84329	1170.60000	2803.30000	Capital Stock Lagged GE
y3	**Dependent**	20	400.16000	250.61885	97.80000	888.90000	Value of Outstanding Shares GE Lagged
x1	**Independent**	20	42.89150	19.11019	12.93000	90.08000	Gross Investment W
x2	**Independent**	20	670.91000	222.39193	191.50000	1193.50000	Capital Stock Lagged W

Figure 42.24 shows the parameter estimates for the constant, the lag zero coefficients of exogenous variables, and the lag one AR coefficients. From the schematic representation of parameter estimates, the significance of the parameter estimates can be easily verified. The symbol "C" means the constant and "XL0" means the lag zero coefficients of exogenous variables.

Figure 42.24 Parameter Estimates for the VARX(1, 0) Model

The VARMAX Procedure

Type of Model	VARX(1,0)
Estimation Method	Least Squares Estimation

Constant	
Variable	**Constant**
y1	-12.01279
y2	702.08673
y3	-22.42110

XLag			
Lag	**Variable**	**x1**	**x2**
0	**y1**	1.69281	-0.00859
	y2	-6.09850	2.57980
	y3	-0.02317	-0.01274

AR				
Lag	**Variable**	**y1**	**y2**	**y3**
1	**y1**	0.23699	0.00763	0.02941
	y2	-2.46656	0.16379	-0.84090
	y3	0.95116	0.00224	0.93801

Schematic Representation			
Variable/Lag	**C**	**XL0**	**AR1**
y1	.	+.	...
y2	+	.+	...
y3	-	..	+.+

+ is > 2*std error, - is < -2*std error, . is between, * is N/A

Figure 42.25 shows the parameter estimates and their significance.

Figure 42.25 Parameter Estimates for the VARX(1, 0) Model Continued

Model Parameter Estimates

Equation	Parameter	Estimate	Standard Error	t Value	Pr > |t|	Variable
y1	CONST1	-12.01279	27.47108	-0.44	0.6691	1
	XL0_1_1	1.69281	0.54395	3.11	0.0083	x1(t)
	XL0_1_2	-0.00859	0.05361	-0.16	0.8752	x2(t)
	AR1_1_1	0.23699	0.20668	1.15	0.2722	y1(t-1)
	AR1_1_2	0.00763	0.01627	0.47	0.6470	y2(t-1)
	AR1_1_3	0.02941	0.04852	0.61	0.5548	y3(t-1)
y2	CONST2	702.08673	256.48046	2.74	0.0169	1
	XL0_2_1	-6.09850	5.07849	-1.20	0.2512	x1(t)
	XL0_2_2	2.57980	0.50056	5.15	0.0002	x2(t)
	AR1_2_1	-2.46656	1.92967	-1.28	0.2235	y1(t-1)
	AR1_2_2	0.16379	0.15193	1.08	0.3006	y2(t-1)
	AR1_2_3	-0.84090	0.45304	-1.86	0.0862	y3(t-1)
y3	CONST3	-22.42110	10.31166	-2.17	0.0487	1
	XL0_3_1	-0.02317	0.20418	-0.11	0.9114	x1(t)
	XL0_3_2	-0.01274	0.02012	-0.63	0.5377	x2(t)
	AR1_3_1	0.95116	0.07758	12.26	0.0001	y1(t-1)
	AR1_3_2	0.00224	0.00611	0.37	0.7201	y2(t-1)
	AR1_3_3	0.93801	0.01821	51.50	0.0001	y3(t-1)

The fitted model is given as

$$\begin{pmatrix} y_{1t} \\ y_{2t} \\ y_{3t} \end{pmatrix} = \begin{pmatrix} -12.013 \\ (27.471) \\ 702.086 \\ (256.480) \\ -22.421 \\ (10.312) \end{pmatrix} + \begin{pmatrix} 1.693 & -0.009 \\ (0.544) & (0.054) \\ -6.099 & 2.580 \\ (5.078) & (0.501) \\ -0.023 & -0.013 \\ (0.204) & (0.020) \end{pmatrix} \begin{pmatrix} x_{1t} \\ x_{2t} \end{pmatrix}$$

$$+ \begin{pmatrix} 0.237 & 0.008 & 0.029 \\ (0.207) & (0.016) & (0.049) \\ -2.467 & 0.164 & -0.841 \\ (1.930) & (0.152) & (0.453) \\ 0.951 & 0.002 & 0.938 \\ (0.078) & (0.006) & (0.018) \end{pmatrix} \begin{pmatrix} y_{1,t-1} \\ y_{2,t-1} \\ y_{3,t-1} \end{pmatrix} + \begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \\ \epsilon_{3t} \end{pmatrix}$$

Parameter Estimation and Testing on Restrictions

In the previous example, the VARX(1,0) model is written as

$$\mathbf{y}_t = \boldsymbol{\delta} + \Theta_0^* \mathbf{x}_t + \Phi_1 \mathbf{y}_{t-1} + \boldsymbol{\epsilon}_t$$

with

$$\Theta_0^* = \begin{pmatrix} \theta_{11}^* & \theta_{12}^* \\ \theta_{21}^* & \theta_{22}^* \\ \theta_{31}^* & \theta_{32}^* \end{pmatrix} \quad \Phi_1 = \begin{pmatrix} \phi_{11} & \phi_{12} & \phi_{13} \\ \phi_{21} & \phi_{22} & \phi_{23} \\ \phi_{31} & \phi_{32} & \phi_{33} \end{pmatrix}$$

In Figure 42.25 of the preceding section, you can see several insignificant parameters. For example, the coefficients XL0_1_2, AR1_1_2, and AR1_3_2 are insignificant.

The following statements restrict the coefficients of $\theta_{12}^* = \phi_{12} = \phi_{32} = 0$ for the VARX(1,0) model:

```
/*--- Models with Restrictions and Tests ---*/

proc varmax data=grunfeld;
   model y1-y3 = x1 x2 / p=1 print=(estimates);
   restrict XL(0,1,2)=0, AR(1,1,2)=0, AR(1,3,2)=0;
run;
```

The output in Figure 42.26 shows that three parameters θ_{12}^*, ϕ_{12}, and ϕ_{32} are replaced by the restricted values, zeros, and their standard errors are also zeros to indicate that the parameters are fixed to these values.

Figure 42.26 Parameter Estimation with Restrictions

The VARMAX Procedure

Model Parameter Estimates

Equation	Parameter	Estimate	Standard Error	t Value	Pr > |t|	Variable
y1	CONST1	-2.16781	13.13755	-0.17	0.8715	1
	XL0_1_1	1.67592	0.40792	4.11	0.0012	x1(t)
	XL0_1_2	0.00000	0.00000			x2(t)
	AR1_1_1	0.27671	0.17606	1.57	0.1401	y1(t-1)
	AR1_1_2	0.00000	0.00000			y2(t-1)
	AR1_1_3	0.01747	0.03519	0.50	0.6279	y3(t-1)
y2	CONST2	768.14598	224.12735	3.43	0.0045	1
	XL0_2_1	-6.30880	4.85729	-1.30	0.2166	x1(t)
	XL0_2_2	2.65308	0.43840	6.05	0.0001	x2(t)
	AR1_2_1	-2.16968	1.83550	-1.18	0.2584	y1(t-1)
	AR1_2_2	0.10945	0.11751	0.93	0.3686	y2(t-1)
	AR1_2_3	-0.93053	0.41478	-2.24	0.0429	y3(t-1)
y3	CONST3	-19.88165	7.69575	-2.58	0.0227	1
	XL0_3_1	-0.03576	0.20079	-0.18	0.8614	x1(t)
	XL0_3_2	-0.00919	0.01747	-0.53	0.6076	x2(t)
	AR1_3_1	0.96398	0.06907	13.96	0.0001	y1(t-1)
	AR1_3_2	0.00000	0.00000			y2(t-1)
	AR1_3_3	0.93412	0.01473	63.41	0.0001	y3(t-1)

The output in Figure 42.27 shows the estimates of the Lagrangian parameters and their significance. Based on the p-values associated with the Lagrangian parameters, you cannot reject the null hypotheses $\theta_{12}^* = 0$, $\phi_{12} = 0$, and $\phi_{32} = 0$ with the 0.05 significance level.

Figure 42.27 RESTRICT Statement Results

Testing of the Restricted Parameters

Parameter	Estimate	Standard Error	t Value	Pr > |t|	Equation
Restrict0	1.74969	21.44026	0.08	0.9353	XL0_1_2 = 0
Restrict1	30.36254	70.74347	0.43	0.6700	AR1_1_2 = 0
Restrict2	55.42191	164.03075	0.34	0.7371	AR1_3_2 = 0

The TEST statement in the following example tests $\phi_{31} = 0$ and $\theta_{12}^{*} = \phi_{12} = \phi_{32} = 0$ for the VARX(1,0) model:

```
proc varmax data=grunfeld;
   model y1-y3 = x1 x2 / p=1;
   test AR(1,3,1)=0;
   test XL(0,1,2)=0, AR(1,1,2)=0, AR(1,3,2)=0;
run;
```

The output in Figure 42.28 shows that the first column in the output is the index corresponding to each TEST statement. You can reject the hypothesis test $\phi_{31} = 0$ at the 0.05 significance level, but you cannot reject the joint hypothesis test $\theta_{12}^{*} = \phi_{12} = \phi_{32} = 0$ at the 0.05 significance level.

Figure 42.28 TEST Statement Results

The VARMAX Procedure

Testing of the Parameters

Test	DF	Chi-Square	Pr > ChiSq
1	1	150.31	<.0001
2	3	0.34	0.9522

Causality Testing

The following statements use the CAUSAL statement to compute the Granger causality test for a VAR(1) model. For the Granger causality tests, the autoregressive order should be defined by the P= option in the MODEL statement. The variable groups are defined in the CAUSAL statement as well. Regardless of whether the variables specified in the GROUP1= and GROUP2= options are designated as dependent or exogenous (independent) variables in the MODEL statement, the CAUSAL statement fits the VAR(p) model by considering the variables in the two groups as dependent variables.

```
/*--- Causality Testing ---*/

proc varmax data=grunfeld;
   model y1-y3 = x1 x2 / p=1 noprint;
   causal group1=(x1) group2=(y1-y3);
   causal group1=(y3) group2=(y1 y2);
run;
```

The output in Figure 42.29 is associated with the CAUSAL statement. The first CAUSAL statement fits the VAR(1) model by using the variables $y1$, $y2$, $y3$, and $x1$. The second CAUSAL statement fits the VAR(1) model by using the variables $y1$, $y3$, and $y2$.

Figure 42.29 CAUSAL Statement Results

The VARMAX Procedure

Granger-Causality Wald Test			
Test	**DF**	**Chi-Square**	**Pr > ChiSq**
1	3	2.40	0.4946
2	2	262.88	<.0001

Test 1:	**Group 1 Variables:**	x1
	Group 2 Variables:	y1 y2 y3
Test 2:	**Group 1 Variables:**	y3
	Group 2 Variables:	y1 y2

The null hypothesis of the Granger causality test is that GROUP1 is influenced only by itself, and not by GROUP2.

The first column in the output is the index corresponding to each CAUSAL statement. The output shows that you cannot reject that $x1$ is influenced by itself and not by ($y1$, $y2$, $y3$) at the 0.05 significance level for Test 1. You can reject that $y3$ is influenced by itself and not by ($y1$, $y2$) for Test 2. For more information, see the section "VAR and VARX Modeling" on page 3067.

Multivariate GARCH Models

Modeling and forecasting the volatility of time series has been the focus of many researchers and practitioners, especially in the fields of risk management, portfolio optimization, and asset pricing. One of the most powerful tools for volatility modeling is the autoregressive conditional heteroscedasticity (ARCH) model proposed by Engle (1982) and extended by Bollerslev (1986) to the generalized autoregressive conditional heteroscedasticity (GARCH) model. The VARMAX procedure supports three forms of multivariate GARCH models: BEKK, CCC, and DCC. This section shows some examples of how to specify, estimate, and compare various forms of multivariate GARCH models.

Data about two indices, the Dow Jones Industrial Average index and the Standard & Poor's 500 index, are obtained from Yahoo Finance and used in this section. The sample contains daily data from February 16, 2005, to February 13, 2015. The following statements input the daily prices and then generate the daily returns:

```
data indices;
   input date : MMDDYY10. DJIA SP500;
   logDJIA = log(DJIA); logSP500 = log(SP500);
   rDJIA = (logDJIA-lag(logDJIA))*100;
   rSP500 = (logSP500-lag(logSP500))*100;
datalines;
2/16/2005    10834.88    1210.34
2/17/2005    10754.26    1200.75
2/18/2005    10785.22    1201.59
```

```
   ... more lines ...

2/12/2015    17972.38    2088.48
2/13/2015    18019.35    2096.99
;
```

To model the volatility of bivariate returns, rDJIA and rSP500, you can start with the BEKK GARCH(1,1) model. The following equations describe the bivariate BEKK GARCH(1,1) model:

$$
\begin{aligned}
r_t &= H_t^{\frac{1}{2}} \epsilon_t \\
H_t &= C + A_1' r_{t-1} r_{t-1}' A_1 + G_1' H_{t-1} G_1 \\
&= \begin{bmatrix} c_{11} & c_{12} \\ c_{12} & c_{22} \end{bmatrix} + \begin{bmatrix} a_{11,1} & a_{12,1} \\ a_{21,1} & a_{22,1} \end{bmatrix}' \begin{bmatrix} r_{1,t-1} \\ r_{2,t-1} \end{bmatrix} \begin{bmatrix} r_{1,t-1} \\ r_{2,t-1} \end{bmatrix}' \begin{bmatrix} a_{11,1} & a_{12,1} \\ a_{21,1} & a_{22,1} \end{bmatrix} \\
&+ \begin{bmatrix} g_{11,1} & g_{12,1} \\ g_{21,1} & g_{22,1} \end{bmatrix}' \begin{bmatrix} h_{11,t-1} & h_{12,t-1} \\ h_{12,t-1} & h_{22,t-1} \end{bmatrix} \begin{bmatrix} g_{11,1} & g_{12,1} \\ g_{21,1} & g_{22,1} \end{bmatrix}
\end{aligned}
$$

In these equations, r_t is the vector of returns at time t, H_t is the conditional covariance matrix of r_t, $H_t^{\frac{1}{2}}$ denotes the square root of H_t such that the square of matrix $H_t^{\frac{1}{2}}$ is H_t, ϵ_t is the innovation at time t and follows an iid bivariate standard normal distribution, C is a 2×2 symmetric parameter matrix, A_1 is a 2×2 full parameter matrix for the first lag of the ARCH term, and G_1 is a 2×2 full parameter matrix for the first lag of the GARCH term. Hence, there are 11 parameters in total for a bivariate BEKK GARCH(1,1) model; that is, a vector $\left(c_{11}, c_{12}, c_{22}, a_{11,1}, a_{21,1}, a_{12,1}, a_{22,1}, g_{11,1}, g_{21,1}, g_{12,1}, g_{22,1}\right)'$.

You can use the FORM=BEKK option in the GARCH statement to specify the BEKK GARCH form, or you can omit this option because BEKK is the default value for the FORM= option. The Q= option in the GARCH statement specifies the lags of the ARCH terms, and the P= option in the GARCH statement specifies the lags of the GARCH terms. The forecasts of conditional covariance matrices are output to a SAS data set when you specify the OUTHT= option in the GARCH statement. The parameter estimates and their covariance matrix are output to a SAS data set when you specify the OUTEST= option together with the OUTCOV option in the PROC VARMAX statement.

The following statement specifies the BEKK GARCH(1,1) model:

```
/*--- BEKK ---*/

proc varmax data=indices outest=oebekk outcov;
   model rDJIA rSP500 / noint;
   garch p=1 q=1 form=bekk outht=ohbekk;
run;
```

Figure 42.30 shows the log likelihood and the information criteria. They are used later in the model comparison.

Figure 42.30 BEKK GARCH Log Likelihood and Information Criteria

The VARMAX Procedure

Log-likelihood	1360.976

Information Criteria	
AICC	-2699.85
HQC	-2676.68
AIC	-2699.95
SBC	-2635.82
FPEC	0.080617

Figure 42.31 shows the parameters estimates for the BEKK GARCH(1,1) model. For the constant term C, **GCHC***i_j*, $i, j = 1, 2$, correspond to parameters c_{ij}, respectively. Because C is symmetric, **GCHC2_1** is omitted. For the ARCH and GARCH terms, **ACH***l_i_j*, $l = 1, i, j = 1, 2$, correspond to $a_{ij,l}$, respectively, and **GCH***l_i_j*, $l = 1, i, j = 1, 2$, correspond to $g_{ij,l}$, respectively.

Figure 42.31 BEKK GARCH Parameter Estimates

GARCH Model Parameter Estimates

Parameter	Estimate	Standard Error	t Value	Pr > |t|
GCHC1_1	0.19101	0.00000		
GCHC1_2	0.09343	0.00000		
GCHC2_2	0.00000	0.01807	0.00	1.0000
ACH1_1_1	0.27518	0.13503	2.04	0.0417
ACH1_2_1	0.20619	0.11122	1.85	0.0639
ACH1_1_2	0.24907	0.11982	2.08	0.0377
ACH1_2_2	0.23448	0.09739	2.41	0.0161
GCH1_1_1	0.11391	0.10984	1.04	0.2998
GCH1_2_1	0.64841	0.11363	5.71	0.0001
GCH1_1_2	0.75455	0.11263	6.70	0.0001
GCH1_2_2	0.20598	0.11520	1.79	0.0739

As shown in Figure 42.31, the standard errors of **GCHC1_1** and **GCHC1_2** are both zeros. It might be a sign that the numerical optimization for the BEKK GARCH model converges to a local minimum instead of the global minimum, which often happens for nonlinear optimization of complex models that have many parameters. A possible way to solve this problem is to try different initial values. To obtain reasonable initial values, the following statements fit a diagonal BEKK GARCH model (that is, a restricted BEKK GARCH model in which the ARCH and GARCH parameter matrices are diagonal):

```
/*--- Diagonal BEKK ---*/

proc varmax data=indices outest=oebekk outcov;
   model rDJIA rSP500 / noint;
   garch p=1 q=1 form=bekk;
   restrict ach(1,1,2), ach(1,2,1), gch(1,1,2), gch(1,2,1);
run;
```

The parameter estimates of the diagonal BEKK GARCH model are shown in Figure 42.32. As expected, the standard errors of the off-diagonal elements of the ARCH and GARCH parameter matrices (namely **ACH1_1_2**, **ACH1_2_1**, **GCH1_1_2**, and **GCH1_2_1**) are all zeros because they are restricted in the RESTRICT statement. All other parameters have valid standard errors.

Figure 42.32 Diagonal BEKK GARCH Parameter Estimates

The VARMAX Procedure

GARCH Model Parameter Estimates

Parameter	Estimate	Standard Error	t Value	Pr > |t|
GCHC1_1	0.01407	0.00254	5.53	0.0001
GCHC1_2	0.01446	0.00262	5.51	0.0001
GCHC2_2	0.01598	0.00299	5.34	0.0001
ACH1_1_1	0.25702	0.01251	20.54	0.0001
ACH1_2_1	0.00000	0.00000		
ACH1_1_2	0.00000	0.00000		
ACH1_2_2	0.26061	0.01302	20.02	0.0001
GCH1_1_1	-0.95794	0.00413	-231.85	0.0001
GCH1_2_1	0.00000	0.00000		
GCH1_1_2	0.00000	0.00000		
GCH1_2_2	-0.95694	0.00443	-216.10	0.0001

Figure 42.33 shows the log likelihood and the information criteria. The log likelihood for the diagonal BEKK GARCH model is larger than that of the previous estimated BEKK GARCH model (which is shown in Figure 42.30). The larger value confirms that the previous BEKK GARCH model does not converge to the global minimum.

Figure 42.33 Diagonal BEKK GARCH Log Likelihood and Information Criteria

Log-likelihood	1520.235

Information Criteria	
AICC	-3026.43
HQC	-3011.66
AIC	-3026.47
SBC	-2985.66
FPEC	0.080617

The following statements reestimate the BEKK GARCH model whose initial values are parameter estimates of the diagonal BEKK GARCH model (which are shown in Figure 42.32):

```
/*--- BEKK with Initial Values ---*/

proc varmax data=indices outest=oebekk outcov;
   model rDJIA rSP500 / noint;
   garch p=1 q=1 form=bekk;
   initial gchc(1,1)=0.01407, gchc(1,2)=0.01446, gchc(2,2)=0.01598,
```

```
            ach(1,1,1)=0.25702, ach(1,2,2)=0.26061,
            gch(1,1,1)=-0.95794, gch(1,2,2)=-0.95694,
            ach(1,1,2), ach(1,2,1), gch(1,1,2), gch(1,2,1);
run;
```

The parameter estimates of the reestimated BEKK GARCH models are shown in Figure 42.34. The standard errors of all parameters are valid.

Figure 42.34 Reestimated BEKK GARCH Parameter Estimates

The VARMAX Procedure

GARCH Model Parameter Estimates

Parameter	Estimate	Standard Error	t Value	Pr > |t|
GCHC1_1	0.01999	0.00394	5.07	0.0001
GCHC1_2	0.02043	0.00391	5.22	0.0001
GCHC2_2	0.02112	0.00408	5.18	0.0001
ACH1_1_1	0.07178	0.10153	0.71	0.4796
ACH1_2_1	0.22679	0.09285	2.44	0.0147
ACH1_1_2	-0.09556	0.11262	-0.85	0.3962
ACH1_2_2	0.41214	0.10167	4.05	0.0001
GCH1_1_1	-0.95018	0.03580	-26.55	0.0001
GCH1_2_1	0.01069	0.03266	0.33	0.7434
GCH1_1_2	0.03746	0.04018	0.93	0.3513
GCH1_2_2	-0.97038	0.03589	-27.04	0.0001

Figure 42.35 shows the log likelihood and information criteria of the reestimated BEKK GARCH model. As expected, the log likelihood of the reestimated BEKK GARCH model is larger than that of the diagonal BEKK GARCH model. Moreover, the reestimated BEKK GARCH model has a smaller SBC, compared to the SBC of the diagonal BEKK GARCH model (which is shown in Figure 42.33). The smaller SBC means that the BEKK GARCH model should be chosen instead of the diagonal BEKK GARCH model.

Figure 42.35 Reestimated BEKK GARCH Log Likelihood and Information Criteria

Log-likelihood	1542.362

Information Criteria	
AICC	-3062.62
HQC	-3039.45
AIC	-3062.72
SBC	-2998.59
FPEC	0.080617

The number of parameters for a BEKK GARCH model increases very quickly as the number of dependent variables increases. There are $(p+q)k^2 + k(k+1)/2$ parameters for a k-variate BEKK GARCH(p, q) model. For example, a 16-variate BEKK GARCH(1,1) model has 648 parameters to be estimated.

Compared with BEKK GARCH models, CCC GARCH models are much more parsimonious. In a CCC GARCH model, each series follows a GARCH process and their composition through the constant conditional

correlation matrix constructs the conditional covariance matrices. A bivariate CCC GARCH(1,1) has the form

$$
\begin{aligned}
r_t &= H_t^{\frac{1}{2}} \epsilon_t \\
H_t &= D_t S D_t \\
D_t &= \begin{bmatrix} \sqrt{h_{11,t}} & 0 \\ 0 & \sqrt{h_{22,t}} \end{bmatrix} \\
S &= \begin{bmatrix} 1 & s_{12} \\ s_{12} & 1 \end{bmatrix} \\
h_{11,t} &= c_{11} + a_{11,1} r_{1,t-1}^2 + g_{11,1} h_{11,t-1} \\
h_{22,t} &= c_{22} + a_{22,1} r_{2,t-1}^2 + g_{22,1} h_{22,t-1}
\end{aligned}
$$

where D_t is the diagonal matrix with conditional standard deviations and S is the time-invariant conditional correlation matrix. Hence, there are seven parameters to be estimated; that is, a vector $\left(s_{12}, c_{11}, c_{22}, a_{11,1}, a_{22,1}, g_{11,1}, g_{22,1}\right)'$. A k-variate CCC GARCH(p, q) model has $(p + q + 1)k + k(k-1)/2$ parameters. For example, a 16-variate CCC GARCH(1,1) model has 168 parameters to be estimated, about $1/4$ of the number that a BEKK GARCH(1,1) model has.

The following statements estimate a CCC GARCH(1,1) model:

```
/*--- CCC ---*/

proc varmax data=indices outest=oeccc outcov;
   model rDJIA rSP500 / noint;
   garch p=1 q=1 form=ccc outht=ohccc;
run;
```

Figure 42.36 shows the parameter estimates for the CCC GARCH(1,1) model. For the constant conditional correlation matrix S, `CCC1_2` corresponds to the parameter s_{12}.

Figure 42.36 CCC GARCH Parameter Estimates

The VARMAX Procedure

GARCH Model Parameter Estimates

Parameter	Estimate	Standard Error	t Value	Pr > |t|
CCC1_2	0.97294	0.00109	890.75	0.0001
GCHC1_1	0.03713	0.00504	7.37	0.0001
GCHC2_2	0.04004	0.00536	7.47	0.0001
ACH1_1_1	0.06862	0.00737	9.31	0.0001
ACH1_2_2	0.06684	0.00690	9.68	0.0001
GCH1_1_1	0.88472	0.01183	74.81	0.0001
GCH1_2_2	0.88916	0.01099	80.92	0.0001

Figure 42.37 shows the log likelihood and the information criteria. Compared to the SBC for the BEKK GARCH model (shown in Figure 42.35), the SBC for the CCC GARCH model is much larger, which means the CCC GARCH model should not be preferred.

Figure 42.37 CCC GARCH Log Likelihood and Information Criteria

Log-likelihood	1474.578

Information Criteria	
AICC	-2935.11
HQC	-2920.34
AIC	-2935.16
SBC	-2894.34
FPEC	0.080617

The CCC GARCH model is not preferred over the BEKK GARCH model in this case because the basic assumption in the CCC GARCH model—that the conditional correlation matrix is time-invariant—might not hold. A DCC GARCH model relaxes this assumption and models the time-varying conditional correlation matrix in an ARMA form. A bivariate DCC GARCH(1,1) has the form

$$
\begin{aligned}
r_t &= H_t^{\frac{1}{2}} \epsilon_t \\
H_t &= D_t S_t D_t \\
D_t &= \begin{bmatrix} \sqrt{h_{11,t}} & 0 \\ 0 & \sqrt{h_{22,t}} \end{bmatrix} \\
h_{11,t} &= c_{11} + a_{11,1} r_{1,t-1}^2 + g_{11,1} h_{11,t-1} \\
h_{22,t} &= c_{11} + a_{22,1} r_{2,t-1}^2 + g_{22,1} h_{22,t-1} \\
S_t &= \begin{bmatrix} 1 & s_{12,t} \\ s_{12,t} & 1 \end{bmatrix} \\
s_{12,t} &= \frac{q_{12,t}}{\sqrt{q_{11,t} q_{22,t}}} \\
q_{12,t} &= (1-\alpha-\beta) s_{12} + \alpha \frac{r_{1,t-1}}{\sqrt{h_{11,t-1}}} \frac{r_{2,t-1}}{\sqrt{h_{22,t-1}}} + \beta q_{12,t-1} \\
q_{11,t} &= (1-\alpha-\beta) + \alpha \frac{r_{1,t-1}^2}{h_{11,t-1}} + \beta q_{11,t-1} \\
q_{22,t} &= (1-\alpha-\beta) + \alpha \frac{r_{2,t-1}^2}{h_{22,t-1}} + \beta q_{22,t-1}
\end{aligned}
$$

where S_t is the time-varying conditional correlation matrix at time t. Compared to the CCC GARCH model, two more parameters, α and β, are added into the DCC GARCH model. There are nine parameters in total; that is, a vector $\left(\alpha, \beta, s_{12}, c_{11}, c_{22}, a_{11,1}, a_{22,1}, g_{11,1}, g_{22,1}\right)'$.

The following statements estimate a DCC GARCH model:

```
/*--- DCC ---*/

proc varmax data=indices outest=oedcc outcov;
   model rDJIA rSP500 / noint;
   garch p=1 q=1 form=dcc outht=ohdcc;
run;
```

Figure 42.38 shows the parameter estimates for the DCC GARCH(1,1) model. **DCCA** corresponds to the parameter α, **DCCB** corresponds to the parameter β, and **DCCS1_2** corresponds to the parameter s_{12}, the off-diagonal element in the unconditional correlation matrix. The standard errors of many parameter estimates are zeros, which might be a sign that the model does not converge to the global minimum.

Figure 42.38 DCC GARCH Parameter Estimates

The VARMAX Procedure

GARCH Model Parameter Estimates

Parameter	Estimate	Standard Error	t Value	Pr > |t|
DCCA	0.01540	0.00000		
DCCB	0.00000	0.00000		
DCCS1_2	0.98743	0.00040	999.00	0.0001
GCHC1_1	1.28530	0.00000		
GCHC2_2	1.50117	0.00000		
ACH1_1_1	0.03378	0.00216	15.62	0.0001
ACH1_2_2	0.02694	0.00084	32.07	0.0001
GCH1_1_1	0.07596	0.00000		
GCH1_2_2	0.09939	0.00000		

Figure 42.39 shows the log likelihood and the information criteria.

Figure 42.39 DCC GARCH Log Likelihood and Information Criteria

Log-likelihood	700.3131

Information Criteria	
AICC	-1382.55
HQC	-1363.58
AIC	-1382.63
SBC	-1330.16
FPEC	0.080617

Because a CCC GARCH model can be regarded as a restricted DCC GARCH model in which α and β in the conditional correlation equations are restricted to zeros, it is expected that the log likelihood of the "unrestricted" DCC GARCH model should always be larger than (or at least equal to) the log likelihood of the

corresponding CCC GARCH model, even though DCC might have a larger information criterion and not be chosen. Hence, it is unexpected that the log likelihood of the DCC GARCH model (shown in Figure 42.39) is smaller than that of the CCC GARCH model (shown in Figure 42.37). This unexpected phenomenon confirms that the numerical optimization for the DCC GARCH model converges to a local minimum instead of the global minimum. Different initial values should be tried. In addition to some reasonable values for parameters DCCA and DCCB, the INITIAL statement specifies the initial values for the DCC GARCH model parameters in the following statements; these values are based on the corresponding parameter estimates of the CCC GARCH model (shown in Figure 42.36):

```
/*--- DCC with Initial Values ---*/

proc varmax data=indices outest=oedcc outcov;
   model rDJIA rSP500 / noint;
   garch p=1 q=1 form=dcc outht=ohdcc;
   initial DCCA=0.01, DCCB=0.98, DCCS(1,2)  = 0.97294,
           GCHC(1,1)  = 0.03713, GCHC(2,2)  = 0.04004,
           ACH(1,1,1) = 0.06862, ACH(1,2,2) = 0.06684,
           GCH(1,1,1) = 0.88472, GCH(1,2,2) = 0.88916;
run;
```

Figure 42.40 shows the parameter estimates for the reestimated DCC GARCH(1,1) model. All standard errors of parameter estimates are valid.

Figure 42.40 Reestimated DCC GARCH Parameter Estimates

The VARMAX Procedure

GARCH Model Parameter Estimates

Parameter	Estimate	Standard Error	t Value	Pr > |t|
DCCA	0.03802	0.00634	6.00	0.0001
DCCB	0.93782	0.01084	86.49	0.0001
DCCS1_2	0.97401	0.00247	394.22	0.0001
GCHC1_1	0.02193	0.00370	5.93	0.0001
GCHC2_2	0.02395	0.00401	5.97	0.0001
ACH1_1_1	0.07842	0.00787	9.97	0.0001
ACH1_2_2	0.07758	0.00770	10.07	0.0001
GCH1_1_1	0.89540	0.01046	85.58	0.0001
GCH1_2_2	0.89738	0.01012	88.64	0.0001

As shown in Figure 42.41, the log likelihood of the DCC GARCH model increases dramatically. Compared to the SBC of the CCC GARCH model (shown in Figure 42.37), the SBC for the DCC GARCH model is much smaller, and the DCC GARCH model is chosen. However, compared to the SBC for the BEKK GARCH model (shown in Figure 42.35), the SBC for the DCC GARCH model is a little larger, The BEKK GARCH model should be chosen although it has two more parameters than the DCC GARCH model.

Figure 42.41 Reestimated DCC GARCH Log Likelihood and Information Criteria

Log-likelihood	1531.454

Information Criteria	
AICC	-3044.84
HQC	-3025.86
AIC	-3044.91
SBC	-2992.44
FPEC	0.080617

Compared to the BEKK GARCH model, in addition to parsimony, another advantage of the DCC (and also the CCC) GARCH model is that you can use subforms other than GARCH to model the conditional covariance of each series. For example, you can use the threshold GARCH (TGARCH) model for modeling the conditional covariances of rDJIA and rSP500. A bivariate DCC TGARCH(1,1) has the same form as the bivariate DCC GARCH(1,1) except that the conditional covariance equations are replaced by

$$
\begin{aligned}
h_{11,t} &= c_{11} + a_{11,1} r_{1,t-1}^2 + \mathbf{1}_{r_{1,t-1}<0} b_{11,1} r_{1,t-1}^2 + g_{11,1} h_{11,t-1} \\
h_{22,t} &= c_{11} + a_{22,1} r_{2,t-1}^2 + \mathbf{1}_{r_{2,t-1}<0} b_{22,1} r_{2,t-1}^2 + g_{22,1} h_{22,t-1}
\end{aligned}
$$

where the indicator function $\mathbf{1}_A$ is 1 if A is true and 0 otherwise. Compared to the DCC GARCH model, two more parameters, $b_{11,1}$ and $b_{22,1}$, are added to the DCC TGARCH model to catch the so-called leverage effect (that is, the positive and negative returns have different impacts on future volatility).

The following statements include the SUBFORM=TARCH option to fit a bivariate DCC TGACH(1,1) model with the same initial values that are used for the previous DCC GARCH(1,1) model. Because the LEAD=10 option is specified in the OUTPUT statement, the 1- to 10-step-ahead forecasts of rDJIA and rSP500 are output to the OUT= data set odcct and the 1- to 10-step-ahead forecasts of conditional covariance matrices of rDJIA and rSP500 are output to the OUTHT= data set ohdcct.

```
proc varmax data=indices outest=oedcct outcov;
   model rDJIA rSP500 / noint;
   garch p=1 q=1 form=dcc outht=ohdcct subform=tgarch;
   initial DCCA=0.01, DCCB=0.98, DCCS(1,2)  = 0.97294,
           GCHC(1,1)  = 0.03713, GCHC(2,2)  = 0.04004,
           ACH(1,1,1) = 0.06862, ACH(1,2,2) = 0.06684,
           GCH(1,1,1) = 0.88472, GCH(1,2,2) = 0.88916;
   output out=odcct lead=10;
run;
```

Figure 42.42 shows the parameter estimates for the DCC TGARCH(1,1) model. **TACH1_1_1** and **TACH1_2_2** correspond to the parameters $b_{11,1}$ and $b_{22,1}$, respectively. They are significant, which means that the leverage effect exists.

Figure 42.42 DCC TGARCH Parameter Estimates

The VARMAX Procedure

GARCH Model Parameter Estimates

Parameter	Estimate	Standard Error	t Value	Pr > |t|
DCCA	0.04302	0.00669	6.43	0.0001
DCCB	0.92807	0.01142	81.26	0.0001
DCCS1_2	0.97309	0.00248	392.01	0.0001
GCHC1_1	0.02068	0.00305	6.78	0.0001
GCHC2_2	0.02329	0.00346	6.73	0.0001
ACH1_1_1	0.00104	0.00684	0.15	0.8787
ACH1_2_2	0.00314	0.00698	0.45	0.6525
TACH1_1_1	0.11443	0.01207	9.48	0.0001
TACH1_2_2	0.10805	0.01166	9.27	0.0001
GCH1_1_1	0.91490	0.00956	95.68	0.0001
GCH1_2_2	0.91574	0.00964	95.03	0.0001

Figure 42.43 shows the log likelihood and the information criteria. The SBC for the DCC TGARCH model is smaller than the SBC for the BEKK GARCH model (which is shown in Figure 42.35). The smaller SBC means that the DCC TGARCH model is the final winner.

Figure 42.43 DCC TGARCH Log Likelihood and Information Criteria

Log-likelihood	1587.793

Information Criteria	
AICC	-3153.48
HQC	-3130.31
AIC	-3153.59
SBC	-3089.46
FPEC	0.080617

Other subforms of GARCH models—the exponential GARCH (EGARCH) model, the quadratic GARCH (QGARCH) model, and the power GARCH (PGARCH) model—are also supported for the CCC and DCC GARCH models. Furthermore, the multivariate GARCH models can be used together with VARMAX or vector error correction models. For more information and examples, see the sections "Multivariate GARCH Modeling" on page 3110 and "Example 42.4: Analysis of Euro Foreign Exchange Reference Rates" on page 3175.

Syntax: VARMAX Procedure

```
PROC VARMAX options ;
  BOUND restriction, . . . , restriction ;
  BY variables ;
  CAUSAL GROUP1=(variables)GROUP2=(variables) ;
  COINTEG RANK=number < options > ;
  CONDFORE < options > ;
  GARCH options ;
  ID variable INTERVAL=value < ALIGN=value > ;
  INITIAL equation, . . . , equation ;
  MODEL  dependents < = regressors > < , dependents < = regressors > . . . > < / options > ;
  NLOPTIONS options ;
  OUTPUT < options > ;
  RESTRICT restriction, . . . , restriction ;
  TEST restriction, . . . , restriction ;
```

Functional Summary

The statements and options available in the VARMAX procedure are summarized in Table 42.1.

Table 42.1 Functional Summary

Description	Statement	Option
Data Set Options		
Specifies the input data set	VARMAX	DATA=
Writes parameter estimates to an output data set	VARMAX	OUTEST=
Includes covariances in the OUTEST= data set	VARMAX	OUTCOV
Writes the diagnostic checking tests for a model and the cointegration test results to an output data set	VARMAX	OUTSTAT=
Specifies the input data set for scenarios	CONDFORE	SDATA=
Writes the statistics of simulated forecasts to an output data set	CONDFORE	OUT=
Writes the simulated forecasts to an output data set	CONDFORE	OUTSIM=
Writes the conditional covariance matrix to an output data set	GARCH	OUTHT=
Writes actuals, predictions, residuals, and confidence limits to an output data set	OUTPUT	OUT=
BY Groups		
Specifies BY-group processing	BY	
ID Variable		
Specifies the identifying variable	ID	
Specifies the time interval between observations	ID	INTERVAL=

Table 42.1 *continued*

Description	Statement	Option
Controls the alignment of SAS date values	ID	ALIGN=
Options to Control the Optimization Process		
Specifies the optimization options	NLOPTIONS	
Printing Control Options		
Specifies how many lags to print results	MODEL	LAGMAX=
Suppresses the printed output	MODEL	NOPRINT
Requests all printing options	MODEL	PRINTALL
Requests the printing format	MODEL	PRINTFORM=
Controls plots produced through ODS GRAPHICS	VARMAX	PLOTS=
PRINT= Option		
Prints the correlation matrix of parameter estimates	MODEL	CORRB
Prints the cross-correlation matrices of independent variables	MODEL	CORRX
Prints the cross-correlation matrices of dependent variables	MODEL	CORRY
Prints the covariance matrices of prediction errors	MODEL	COVPE
Prints the cross-covariance matrices of the independent variables	MODEL	COVX
Prints the cross-covariance matrices of the dependent variables	MODEL	COVY
Prints the covariance matrix of parameter estimates	MODEL	COVB
Prints the decomposition of the prediction error covariance matrix	MODEL	DECOMPOSE
Prints the residual diagnostics	MODEL	DIAGNOSE
Prints the contemporaneous relationships among the components of the vector time series	MODEL	DYNAMIC
Prints the parameter estimates	MODEL	ESTIMATES
Prints the infinite order AR representation	MODEL	IARR
Prints the impulse response function	MODEL	IMPULSE=
Prints the impulse response function in the transfer function	MODEL	IMPULSX=
Prints the partial autoregressive coefficient matrices	MODEL	PARCOEF
Prints the partial canonical correlation matrices	MODEL	PCANCORR
Prints the partial correlation matrices	MODEL	PCORR
Prints the eigenvalues of the companion matrix	MODEL	ROOTS
Prints the Yule-Walker estimates	MODEL	YW
Model Estimation and Order Selection Options		
Specifies the initial parameter values for non-linear optimization when the model is estimated through the maximum likelihood method	INITIAL	

Table 42.1 *continued*

Description	Statement	Option
Centers the dependent variables	MODEL	CENTER
Specifies the degrees of differencing for the specified model variables	MODEL	DIF=
Specifies the degrees of differencing for all independent variables	MODEL	DIFX=
Specifies the degrees of differencing for all dependent variables	MODEL	DIFY=
Specifies the estimation method	MODEL	METHOD=
Selects the tentative order	MODEL	MINIC=
Suppresses the current values of independent variables	MODEL	NOCURRENTX
Suppresses the intercept parameters	MODEL	NOINT
Specifies the number of seasonal periods	MODEL	NSEASON=
Specifies the order of autoregressive polynomial	MODEL	P=
Specifies the Bayesian prior model	MODEL	PRIOR=
Specifies the order of moving-average polynomial	MODEL	Q=
Centers the seasonal dummies	MODEL	SCENTER
Specifies the degree of time trend polynomial	MODEL	TREND=
Specifies the denominator for error covariance matrix estimates	MODEL	VARDEF=
Specifies the lag order of independent variables	MODEL	XLAG=
GARCH-Related Options		
Specifies how to calculate the constant (unconditional) correlation matrix in the CCC (DCC) GARCH model	GARCH	CORRCONSTANT=
Specifies the type of the multivariate GARCH model	GARCH	FORM=
Specifies the order of the GARCH polynomial	GARCH	P=
Specifies the order of the ARCH polynomial	GARCH	Q=
Specifies the type of the univariate GARCH model for each innovation in the CCC or DCC GARCH model	GARCH	SUBFORM=
Cointegration-Related Options		
Specifies the restriction on the drift in the VECM	COINTEG	ECTREND
Prints the results from the weak exogeneity test of the long-run parameters	COINTEG	EXOGENEITY
Specifies the restriction on the cointegrated coefficient matrix	COINTEG	H=
Specifies the restriction on the adjustment coefficient matrix	COINTEG	J=

Table 42.1 *continued*

Description	Statement	Option
Specifies the nonlinear constraints that the adjustment coefficient matrix and the cointegrated coefficient matrix are both full rank	COINTEG	NLC
Specifies the variable name whose cointegrating vectors are normalized	COINTEG	NORMALIZE=
Specifies a cointegration rank	COINTEG	RANK=
Prints the Johansen cointegration rank test	MODEL	COINTTEST= (JOHANSEN=)
Prints the Stock-Watson common trends test	MODEL	COINTTEST=(SW=)
Prints the Dickey-Fuller unit root test	MODEL	DFTEST=
Specifies the vector error correction model (obsolete)[1]	MODEL	ECM=
Long Memory Options		
Specifies the Vector autoregressive fractionally integrated moving average model	MODEL	FI
Tests and Restrictions on Parameters		
Tests the Granger causality	CAUSAL	GROUP1= GROUP2=
Places and tests restrictions on parameter estimates	BOUND	
Places and tests restrictions on parameter estimates	RESTRICT	
Tests hypotheses on parameter estimates	TEST	
Forecasting Control Options		
Specifies the size of confidence limits for forecasting	OUTPUT	ALPHA=
Starts forecasting before end of the input data	OUTPUT	BACK=
Specifies how many periods to forecast	OUTPUT	LEAD=
Suppresses the printed forecasts	OUTPUT	NOPRINT
Conditional Forecasts and Scenario Analysis Options		
Specifies the size of the credible interval	CONDFORE	ALPHA=
Specifies the number of multistep forecast values to compute	CONDFORE	LEAD=
Specifies the number of burn-in iterations	CONDFORE	NBI=
Specifies the number of Monte Carlo iterations	CONDFORE	NMC=
Specifies whether and how to consider the uncertainty of parameters	CONDFORE	PARM=
Specifies a nonnegative integer to use as the seed for generating random number sequences	CONDFORE	SEED=
Specifies the numeric variable that identifies each scenario	CONDFORE	SID=

[1] Starting with SAS/ETS 14.1, it is recommended that you use the COINTEG statement instead.

PROC VARMAX Statement

PROC VARMAX *options* **;**

The following options can be used in the PROC VARMAX statement:

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

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

COVOUT

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

OUTSTAT=*SAS-data-set*
writes residual diagnostic results to an output data set. If the COINTTEST=(JOHANSEN) option is specified, the results of this option are also written to the output data set.

The following statements are the examples of these options in the PROC VARMAX statement:

```
proc varmax data=one outest=est outcov outstat=stat;
   model y1-y3 / p=1;
run;
```

```
proc varmax data=one outest=est outstat=stat;
   model y1-y3 / p=1 cointtest=(johansen);
run;
```

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

PLOTS<**(***global-plot-option***)**> **= (** *plot-request-option* <**(***options***)**> **...** *plot-request-option* <**(***options***)**> **)**
controls the plots produced through ODS Graphics. When you specify only one plot, you can omit the parentheses around the plot request. Some examples follow:

```
plots=none
plots=all
plots=condcorr
plots(unpack)=residual(residual normal)
plots=(forecasts model)
```

For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS" (*SAS/STAT User's Guide*).

```
proc varmax data=one plots=impulse(simple);
   model y1-y3 / p=1;
run;
```

```
proc varmax data=one plots=(model residual);
   model y1-y3 / p=1;
run;

proc varmax data=one plots=forecasts;
   model y1-y3 / p=1;
   output lead=12;
run;
```

The first VARMAX program produces the simple response impulse plots. The second VARMAX program produces the plots associated with the model and prediction errors. The plots associated with prediction errors are the ACF, PACF, IACF, distribution, white-noise, and Normal quantile plots and the prediction error plot. The third VARMAX program produces the FORECASTS and FORECASTSONLY plots.

The *global-plot-option* applies to the impulse and prediction error analysis plots generated by the VARMAX procedure. The following *global-plot-option* is available:

UNPACK displays each graph separately. (By default, some graphs can appear together in a single panel.)

The following *plot-request-options* are available:

ALL produces all plots appropriate for the particular analysis.

CONDCORR produces dynamic conditional covariance plots. This option is available only when the DCC GARCH model is specified. This option is experimental in this release.

FORECASTS <(*forecasts-plot-options*)> produces plots of the forecasts. The forecasts-only plot that shows the multistep forecasts in the forecast region is produced by default. The following *forecasts-plot-options* are available:

- **ALL** produces the FORECASTSONLY and the FORECASTS plots. This is the default.
- **FORECASTS** produces a plot that shows the one-step-ahead as well as the multistep forecasts.
- **FORECASTSONLY** produces a plot that shows only the multistep forecasts.

IMPULSE <(*impulse-plot-options*)> produces the plots of impulse response function and the impulse response of the transfer function.

- **ALL** produces all impulse plots. This is the default.
- **ACCUM** produces the accumulated impulse plot.
- **ORTH** produces the orthogonalized impulse plot.
- **SIMPLE** produces the simple impulse plot.

MODEL produces plots of dependent variables listed in the MODEL statement and plots of the one-step-ahead predicted values for each dependent variables.

NONE suppresses all plots.

RESIDUAL <(*residual-plot-options*)> produces plots associated with the prediction errors obtained after modeling the data. The following *residual-plot-options* are available:

ALL produces all plots associated with the analysis of the prediction errors. This is the default.

RESIDUAL produces prediction error plot.

DIAGNOSTICS produces a panel of plots useful in assessing the autocorrelations and white-noise of the prediction errors. The panel consists of the following:

- the autocorrelation plot of the prediction errors
- the partial autocorrelation plot of the prediction errors
- the inverse autocorrelation plot of the prediction errors
- the log scaled white noise plot of the prediction errors

NORMAL produces a panel of plots useful in assessing normality of the prediction errors. The panel consists of the following:

- distribution of the prediction errors with overlaid the normal curve
- normal quantile plot of the prediction errors

Other Options

In addition, any of the following MODEL statement options can be specified in the PROC VARMAX statement, which is equivalent to specifying the option for every MODEL statement: CENTER, DFTEST=, DIF=, DIFX=, DIFY=, LAGMAX=, METHOD=, MINIC=, NOCURRENTX, NOINT, NOPRINT, NSEASON=, P=, PRINT=, PRINTALL, PRINTFORM=, Q=, SCENTER, TREND=, VARDEF=, and XLAG= options.

The following is an example of the options in the PROC VARMAX statement:

```
proc varmax data=one lagmax=3 method=ml;
   model y1-y3 / p=1;
run;
```

BOUND Statement

BOUND *restriction, . . . , restriction* **;**

The BOUND statement sets up linear bounds for parameters when the maximum likelihood method is applied to the estimation of VARMAX, VECM, VARMAX-GARCH, and VEC-ARMAX-GARCH models. Only one BOUND statement is allowed. If you specify more than one *restriction*, separate them with commas. The *restrictions* are specified in the same manner as the *restrictions* in the RESTRICT statement. For information about how to define restrictions by using matrix expressions, operators, and functions, see the section "RESTRICT Statement" on page 3025. Both equality and inequality constraints are allowed in the BOUND statement, although usually equality constraints are specified in the RESTRICT statement and inequality constraints are specified in the BOUND statement.

To use the BOUND statement, you need to know the form of the model. If you do not specify the GARCH statement, the COINTEG statement, or the ECM=, P=, Q=, or XLAG= option in the MODEL statement, then the BOUND statement is not applicable. If you specify the ECM=(NORMALIZE=), METHOD=LS, or

PRIOR= option in the MODEL statement, or if you specify the EXOGENEITY, H=, J=, or NORMALIZE= option in the COINTEG statement, the BOUND statement is ignored. Nonlinear restrictions on parameters are not supported.

The following is an example of the BOUND statement for a bivariate (k=2) zero-mean VARMA(1,1) model, which is by default estimated by maximum likelihood method because the MA term is present:

```
proc varmax data=one;
   model y1 y2 / noint p=1 q=1;
   bound -1<=AR<=1, 0<MA;
run;
```

This BOUND statement specifies that all AR parameters must be between -1 and 1 and that all MA parameters must be positive.

You can use the BOUND statement together with the RESTRICT statement, as in the following bivariate (k=2) zero-mean VARMA(1,1) model:

```
proc varmax data=one;
   model y1 y2 / noint p=1 q=1;
   bound AR+MA>=0.001;
   restrict AR(1,1,2) = 0.5;
run;
```

BY Statement

BY *variables* **;**

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

When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

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

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

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

The following is an example of the BY statement:

```
proc varmax data=one;
   by region;
   model y1-y3 / p=1;
run;
```

CAUSAL Statement

CAUSAL GROUP1=(*variables***)GROUP2=(***variables***) ;**

A CAUSAL statement prints the Granger causality test by fitting the VAR(p) model by using all variables defined in GROUP1 and GROUP2. Any number of CAUSAL statements can be specified. The CAUSAL statement proceeds with the MODEL statement and uses the variables and the autoregressive order, p, specified in the MODEL statement. Variables in the GROUP1= and GROUP2= options should be defined in the MODEL statement. If the P=0 option is specified in the MODEL statement, the CAUSAL statement is not applicable.

The null hypothesis of the Granger causality test is that GROUP1 is influenced only by itself, and not by GROUP2. If the hypothesis test fails to reject the null, then the variables listed in GROUP1 might be considered as independent variables.

For more information, see the section "VAR and VARX Modeling" on page 3067.

The following is an example of the CAUSAL statement. You specify the CAUSAL statement with the GROUP1= and GROUP2= options.

```
proc varmax data=one;
   model y1-y3 = x1 / p=1;
   causal group1=(x1) group2=(y1-y3);
   causal group1=(y2) group2=(y1 y3);
run;
```

The first CAUSAL statement fits the VAR(1) model by using the variables $y1$, $y2$, $y3$, and $x1$ and tests the null hypothesis that $x1$ causes the other variables, $y1$, $y2$, and $y3$, but the other variables do not cause $x1$. The second CAUSAL statement fits the VAR(1) model by using the variables $y1$, $y3$, and $y2$ and tests the null hypothesis that $y2$ causes the other variables, $y1$ and $y3$, but the other variables do not cause $y2$.

COINTEG Statement

COINTEG RANK=*number* < *options* > **;**

The COINTEG statement fits the vector error correction model to the data, tests the restrictions of the long-run parameters and the adjustment parameters, and tests for weak exogeneity in the long-run parameters. The P= option in the MODEL statement specifies the autoregressive order of the VECM. Only one COINTEG statement is allowed.

The cointegrated system uses maximum likelihood estimation. If there are no moving average (MA) terms specified by the Q= option in the MODEL statement, no GARCH terms specified in the GARCH statement, and no general restrictions specified in the BOUND and RESTRICT statements, then PROC VARMAX applies the maximum likelihood analysis proposed by Johansen and Juselius (1990); Johansen (1995a, b).

Otherwise, the likelihood is maximized using an optimizer whose options can be specified in the NLOPTIONS statement.

The following statements fit a VECM(2):

```
proc varmax data=one;
   model y1-y3 / p=2;
   cointeg rank=1;
run;
```

To test restrictions on $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, you specify the J= option and the H= option, respectively. You specify the EXOGENEITY option in the COINTEG statement for tests of weak exogeneity in the long-run parameters.

The following example of the COINTEG statement specifies tests of restrictions on $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, along with tests of weak exogeneity:

```
proc varmax data=one;
   model y1-y3 / p=2;
   cointeg rank=1 h=(1 0, -1 0, 0 1)
           j=(1 0, 0 0, 0 1) exogeneity;
run;
```

You must specify the following option:

RANK=*number*

specifies the cointegration rank of the cointegrated system. The rank of cointegration should be greater than 0 and less than the number of dependent (endogenous) variables. If *number* is different from the value of the RANK= option specified in the ECM= option in the MODEL statement, the *number* specified here is used for the rank.

You can also specify the following *options* in the COINTEG statement:

ECTREND

specifies the restriction on the drift in the VECM. This option is used in the following cases:

- There is no separate drift in the VECM, but a constant enters only through the error correction term. For example, for VECM(p),

$$\Delta \mathrm{y}_t = \boldsymbol{\alpha}(\boldsymbol{\beta}', \boldsymbol{\beta}_0)(\mathrm{y}'_{t-1}, 1)' + \sum_{i=1}^{p-1} \Phi_i^* \Delta \mathrm{y}_{t-i} + \boldsymbol{\epsilon}_t$$

  An example of the ECTREND option follows:

```
model y1 y2 / p=2;
cointeg rank=1 ectrend;
```

- There is a separate drift and no separate linear trend in the VECM, but a linear trend enters only through the error correction term. For example, for VECM(p),

$$\Delta \mathrm{y}_t = \boldsymbol{\alpha}(\boldsymbol{\beta}', \boldsymbol{\beta}_1)(\mathrm{y}'_{t-1}, t)' + \sum_{i=1}^{p-1} \Phi_i^* \Delta \mathrm{y}_{t-i} + \boldsymbol{\delta}_0 + \boldsymbol{\epsilon}_t$$

  An example of the ECTREND option with the TREND= option follows:

```
model y1 y2 / p=2 trend=linear;
cointeg rank=1 ectrend;
```

If you specify both this option and the NSEASON option in the MODEL statement, then the NSEASON option is ignored. If you specify the NOINT option in the MODEL statement, then this option is ignored.

EXOGENEITY

formulates the likelihood ratio tests for testing weak exogeneity in the long-run parameters. The null hypothesis is that one variable is weakly exogenous for the others.

H=(*matrix*)

specifies the restrictions $\mathbf{H}$ on the $k \times r$ or $(k+1) \times r$ cointegrated coefficient matrix $\tilde{\boldsymbol{\beta}}$ such that $\tilde{\boldsymbol{\beta}} = \mathbf{H}\boldsymbol{\phi}$, where $\mathbf{H}$ is known and $\boldsymbol{\phi}$ is unknown. If you do not specify the ECTREND option, then the cointegrated coefficient matrix $\tilde{\boldsymbol{\beta}}$ is the cointegrating matrix $\boldsymbol{\beta}$ and the $\mathbf{H}$ matrix has dimension $k \times m$. If you specify the ECTREND option, then the cointegrated coefficient matrix $\tilde{\boldsymbol{\beta}}$ is the cointegrating matrix $\boldsymbol{\beta}$ stacked with the coefficient row vector $\boldsymbol{\beta}_0$ or $\boldsymbol{\beta}_1$ for the constant or linear trend in the error correction term, and the $\mathbf{H}$ matrix has dimension $(k+1) \times m$. Here k is the number of dependent variables and m is $r \leq m < k$, where r is defined in the RANK=*r* option.

For example, consider a VECM(2) with rank equal to 1 on four dependent variables. Then, $\boldsymbol{\beta} = (\beta_{11}, \beta_{21}, \beta_{31}, \beta_{41})'$. To test the null hypothesis $\beta_{11} + \beta_{21} = 0$ (that is, $\mathbf{H}_\perp'\boldsymbol{\beta} = 0$, where $\mathbf{H}_\perp = (1\ -1\ 0\ 0)'$), you can use the following statements to specify the restriction matrix $\mathbf{H}$:

```
model y1-y4 / p=2;
cointeg rank=1 h=(1 0 0, -1 0 0, 0 1 0, 0 0 1);
```

Here the dimension of matrix $\mathbf{H}$ is 4×3 because $k = 4$ and $m = 3$, and each row of the matrix $\mathbf{H}$ is separated by commas. Note that $\mathbf{H}_\perp'\mathbf{H} = 0$; that is, the $\mathbf{H}$ and $\mathbf{H}_\perp$ matrices are orthogonal.

When the series has no separate deterministic trend, and therefore you specify the ECTREND option, the constant term should be restricted by $\boldsymbol{\alpha}_\perp'\boldsymbol{\delta} = 0$. The matrix $\boldsymbol{\alpha}_\perp$ is a $k \times (k-r)$ full-rank matrix orthogonal to $\boldsymbol{\alpha}$, such that $rank(\boldsymbol{\alpha}_\perp) = k - r$ and $\boldsymbol{\alpha}_\perp'\boldsymbol{\alpha} = 0$. The $\tilde{\boldsymbol{\beta}}$ becomes $(\boldsymbol{\beta}', \boldsymbol{\beta}_0)'$ or $\tilde{\boldsymbol{\beta}} = (\beta_{11}, \beta_{21}, \beta_{31}, \beta_{41}, \beta_{11}^{(0)})'$. As for the previous test of $\beta_{11} + \beta_{21} = 0$ (that is, $\mathbf{H}_\perp'\tilde{\boldsymbol{\beta}} = 0$, where $\mathbf{H}_\perp = (1\ -1\ 0\ 0\ 0)'$), you can specify the restriction matrix $\mathbf{H}$ as follows:

```
model y1-y4 / p=2;
cointeg rank=1 ectrend
   h=(1 0 0 0, -1 0 0 0, 0 1 0 0, 0 0 1 0, 0 0 0 1);
```

Because the dimension is changed in the $\mathbf{H}_\perp$ matrix, the dimension of $\mathbf{H}$ matrix has to be adjusted accordingly.

When the cointegrated system contains three dependent variables and the RANK=2 option is specified, the test of $\beta_{1j} = -\beta_{2j}$ for $j = 1, 2$ can be run with the following restriction matrix $\mathbf{H}$, where $\mathbf{H}_\perp = (1\ 1\ 0)'$ and $\mathbf{H}_\perp'\boldsymbol{\beta} = 0$:

```
cointeg rank=2 h=(1 0, -1 0, 0 1);
```

There are many ways to achieve a matrix that is orthogonal to a particular matrix. The following statements illustrate how to obtain the orthogonal matrix through QR decomposition:

```
proc iml;
   /* For a given matrix H_dot, */
   H_dot = {1 1 0}`;
   /* get its QR decomposition, i.e., H_dot = QR. */
   call qr(Q, R, piv, lindep, H_dot);
   /* Then, the matrix orthogonal to H_dot
      can be extracted from Q. */
   H = Q[,ncol(H_dot)+1:nrow(H_dot)];
   /* Finally, normalize each column of H if necessary. */
   do i = 1 to ncol(H);
      k = 0;
      do j = nrow(H) to 1 by -1;
         if (H[j,i]^=0) then k=j;
      end;
      if (k=0) then
         print "Error: H is not full rank!";
      else
         do j = nrow(H) to 1 by -1;
            H[j,i] = H[j,i] / H[k,i];
         end;
   end;
   print "The given matrix is:";
   print H_dot;
   print "The matrix orthogonal to it is:";
   print H;
quit;
```

J=(*matrix***)**

specifies the restrictions $\mathbf{J}$ on the $k \times r$ adjustment matrix $\boldsymbol{\alpha}$ such that $\boldsymbol{\alpha} = \mathbf{J}\boldsymbol{\psi}$, where $\mathbf{J}$ is known and $\boldsymbol{\psi}$ is unknown. The $k \times m$ matrix $\mathbf{J}$ is specified by using this option, where k is the number of dependent variables, m is $r \leq m < k$, and r is defined in the RANK=*r* option.

For example, suppose the system contains four variables, the RANK=1 option is specified, and you want to test $\alpha_j = 0$ for $j = 2, 3, 4$ —that is, $\mathbf{J}'_{\perp}\boldsymbol{\alpha} = 0$, where

$$J_{\perp} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Then you can specify the restriction matrix $\mathbf{J}$ as follows:

```
cointeg rank=1 j=(1, 0, 0, 0);
```

Suppose the system contains three variables, the RANK=2 option is specified, and you want to test $\alpha_{2j} = 0$ for $j = 1, 2$—that is, $\mathbf{J}_{\perp}'\boldsymbol{\alpha} = 0$, where $\mathbf{J}_{\perp} = (0\ 1\ 0)'$. Then you can specify the restriction matrix $\mathbf{J}$ as follows:

```
cointeg rank=2 j=(1 0, 0 0, 0 1);
```

NLC

specifies the nonlinear constraints that $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are full column rank. Although the constraints are required for a well-defined VECM, only the TECH=QUANEW and TECH=NMSIMP optimization methods in the NLOPTIONS statement support nonlinear constraints. The full-rank constraints are not imposed by default so that other optimization methods, such as TECH=CONGRA or TECH=TRUREG, can be tried. The NLC option works only when numerical optimization is used for estimating VECM (for example, when the BOUND, INITIAL, or RESTRICT statement is specified, or the VEC-ARMA or VEC-ARMA-GARCH model is estimated). That is, the NLC option is ignored if the closed-form solution of parameter estimates and maximum likelihood analysis, which is provided in Johansen and Juselius (1990) and Johansen (1995a, b), can be applied.

NORMALIZE=*variable*

specifies a single dependent (endogenous) *variable* whose cointegrating vectors are normalized. If the *variable* is different from the variable specified in the COINTTEST=(JOHANSEN=) or ECM= option in the MODEL statement, the *variable* in this option is used. If this option is not specified, cointegrating vectors are not normalized.

If the EXOGENEITY, H=, J=, or NORMALIZE= option is specified, the BOUND, GARCH, INITIAL, and RESTRICT statements are all ignored, and the Q= option in the MODEL statement is also ignored.

CONDFORE Statement

CONDFORE < *options* > **;**

The CONDFORE statement defines the options for conditional forecasts and scenario analysis.

You can apply conditional forecasts and scenario analysis for both Bayesian and non-Bayesian vector autoregressive models and vector error correction models, with or without independent variables. The future values of dependent and independent variables (which define the scenario) are saved in a table, which can then be input by specifying the SDATA= option in the CONDFORE statement. If you do not specify the SDATA= option, unconditional forecasts are performed. If you specify several scenarios (which are distinguished by the variable that is specified in the SID= option), conditional forecasts are performed for each scenario. The statistics of forecasts (including the mean, standard error, median, and lower and upper bounds of credible interval of each forecast) are output to the table that is specified in the OUT= option in the CONDFORE statement, and the simulated forecasts in each iteration are output to the table that is specified in the OUTSIM= option in the CONDFORE statement.

If you specify the BY statement, the scenarios in the SDATA= table are applied to each BY group, and the BY variables are included in the tables that are specified in the OUT= and OUTSIM= options in the CONDFORE

statement. If you specify the ID statement, the ID variable is included in the table that is specified in the OUT= option in the CONDFORE statement. If you specify the SID= option in the CONDFORE statement, the SID= variable is included in the tables that are specified in the OUT= and OUTSIM= options in the CONDFORE statement.

When the GARCH statement is specified or the Q= or FI option is specified in the MODEL statement, the CONDFORE statement is ignored.

You can specify the following *options*:

ALPHA=α
sets the size, α (the probability of falsely rejecting the null hypothesis), of the credible interval $(100(1-\alpha)\%)$, where α is inclusively between 0 and 1. The credible interval is an equal-tailed interval. By default, ALPHA=0.05, which produces a 95% credible interval.

LEAD=*number*
specifies the number of multistep forecast values to compute. By default, LEAD=12.

NBI=*number*
specifies the number of burn-in iterations. By default, NBI=0.

NMC=*number*
specifies the number of Monte Carlo iterations. By default, NMC=1000.

OUT=*SAS-data-set*
specifies the output table for forecasts. The columns of the table are the mean, standard error, median, and lower and upper bounds of credible interval of the forecasts for each dependent variable in each scenario.

OUTSIM=*SAS-data-set*
specifies the output table for the simulated forecasts in each scenario.

PARM=FIXED | **SAMPLING** < **(SCENARIO)** >
specifies whether and how to consider the uncertainty of parameters. You can specify the following values:

FIXED fixes the parameters that are used in conditional forecasts to the parameter estimates for non-Bayesian models or to the expectation of the posterior distribution of parameters for Bayesian models.

SAMPLING < SCENARIO > samples the parameters from the posterior distribution of parameters. If you specify PARM=SAMPLING(SCENARIO), the parameters are sampled through the Gibbs sampling algorithm to consider the effect of the information in each scenario. In theory, it is suggested that the parameter uncertainty should be considered in the conditional forecasts for Bayesian models; however, in practice, the sampling (especially Gibbs sampling) might lead to floating point overflow because of some outlier-like realized parameters. You can specify this value only for Bayesian models.

By default, PARM=FIXED.

SDATA=*SAS-data-set*
specifies the input data table that contains observations for one or multiple scenarios.

SEED=*number*
specifies a nonnegative integer to use as the seed for generating random number sequences. You can use this option to replicate results from different runs if you specify the same positive random seed. If you specify SEED=0, the random seed is determined according to the system clock. By default, SEED=1.

SID=*variable*
SCENARIOID=(*variable***)**
specifies a numeric variable that identifies each scenario. This option is ignored if the SDATA= option is not specified.

Some examples of the CONDFORE statements follow:

```
proc varmax data=one;
   model y1 y2 / p=2;
   condfore out=oucf;
run;

proc varmax data=one;
   model y1 y2 / p=2;
   condfore alpha=0.2 lead=6 sdata=scenarios sid=scenarioIndex
      nbi=1000 nmc=10000 seed=12345 parm=sampling(scenario)
      out=ocf outsim=ocfsim;
run;
```

GARCH Statement

GARCH *options* **;**

The GARCH statement specifies a GARCH-type multivariate conditional heteroscedasticity model.

You can specify the following options:

CORRCONSTANT=ESTIMATE | **EXPECT**
specifies how to calculate the constant or unconditional correlation matrix in the CCC or DCC GARCH model, respectively. If you specify CORRCONSTANT=EXPECT, the constant conditional correlation matrix in the CCC GARCH model or the unconditional correlation matrix in the DCC GARCH model is calculated through the standardized residuals, given the other parameters. After parameter estimates are output, the constant or unconditional correlation matrix for the CCC or DCC GARCH model is output in the CCCCorrConstant or DCCCorrConstant ODS table, respectively. If you specify CORRCONSTANT=ESTIMATE, the correlation matrix is estimated like all other parameters in the model. By default, CORRCONSTANT=ESTIMATE.

FORM=*value*
specifies the representation for a GARCH model. Valid values are as follows:

BEKK specifies a BEKK representation. This is the default.

CCC specifies a constant conditional correlation representation.

DCC specifies a dynamic conditional correlation representation.

OUTHT=*SAS-data-set*

writes the conditional covariance matrix to an output data set. When you use the LEAD= option in the OUTPUT statement together with this option in the GARCH statement, you can obtain the multistep forecast of conditional covariance matrices at any horizons ahead that are of interest.

P=*number*

P=(*number-list***)**

specifies the order of the process or the subset of GARCH terms to be fitted. For example, you can specify the P=(1,3) option. The P=3 option is equivalent to the P=(1,2,3) option. By default, P=0.

Q=*number*

Q=(*number-list***)**

specifies the order of the process or the subset of ARCH terms to be fitted. This option is required in the GARCH statement. For example, you can specify the Q=(2) option. The Q=2 option is equivalent to the Q=(1,2) option.

SUBFORM=*value*

specifies the type of the univariate GARCH model for each innovation in the CCC or DCC GARCH model. If you specify the FORM=BEKK option, the SUBFORM= option is ignored. The values of the SUBFORM= option are as follows:

EGARCH specifies the exponential GARCH, or EGARCH, model.

GARCH specifies the GARCH model with no constraints.

GJR | TGARCH specifies the GJR GARCH (also called threshold GARCH, or TGARCH) model.

PGARCH specifies the power GARCH, or PGARCH, model.

QGARCH specifies the quadratic GARCH, or QGARCH, model.

By default, SUBFORM=GARCH.

If you specify the ECM=(NORMALIZE=) or PRIOR= option in the MODEL statement, or if you specify the EXOGENEITY, H=, J=, or NORMALIZE= option in the COINTEG statement, the GARCH statement is ignored.

For the VAR(1)–ARCH(1) model,

```
model y1 y2 / p=1;
garch q=1 form=bekk;
```

For the multivariate GARCH(1,1) model,

```
model y1 y2;
garch q=1 p=1 form=ccc;
```

Other multivariate GARCH-type models are

```
model y1 y2 = x1 / xlag=1;
garch q=1;

model y1 y2 / q=1;
garch q=1 p=1;
```

For more information, see the section "Multivariate GARCH Modeling" on page 3110.

ID Statement

ID *variable* **INTERVAL=***value* < **ALIGN=***value* > **;**

The ID statement specifies a variable that identifies observations in the input data set. The datetime variable specified in the ID statement is included in the OUT= data set if the OUTPUT statement is specified. The ID *variable* is usually a SAS datetime variable. The values of the ID variable are extrapolated for the forecast observations based on the value of the INTERVAL= option.

ALIGN= *value*

controls the alignment of SAS dates used to identify output observations. The ALIGN= option allows the following values: BEGINNING | BEG | B, MIDDLE | MID | M, and ENDING | END | E. The default is BEGINNING. The ALIGN= option is used to align the ID variable to the beginning, middle, or end of the time ID interval specified by the INTERVAL= option.

INTERVAL=*value*

specifies the time interval between observations. This option is required in the ID statement. The INTERVAL= option is used in conjunction with the ID variable to check that the input data are in order and have no missing periods. The INTERVAL= option is also used to extrapolate the ID values past the end of the input data when the OUTPUT statement is specified.

The following is an example of the ID statement:

```
proc varmax data=one;
   id date interval=qtr align=mid;
   model y1-y3 / p=1;
run;
```

INITIAL Statement

INITIAL *equation, . . . , equation* **;**

The INITIAL statement sets up the initial parameter values for nonlinear optimization when the maximum likelihood method is applied to the estimation of VARMAX, VECM, VARMAX-GARCH, or VEC-ARMAX-GARCH models. Only one INITIAL statement is allowed. If you specify more than one *equation*, separate them with commas. The *equations* are specified in the same manner as the *restrictions* in the RESTRICT statement. For information about how to define equations by using matrix expressions, operators, and functions, see the section "RESTRICT Statement" on page 3025.

To use the INITIAL statement, you need to know the form of the model. If you do not specify the GARCH statement, the COINTEG statement, or the ECM=, P=, Q=, or XLAG= option in the MODEL statement, then the INITIAL statement is not applicable. If you specify the ECM=(NORMALIZE=), METHOD=LS, or PRIOR= option in the MODEL statement, or if you specify the EXOGENEITY, H=, J=, or NORMALIZE= option in the COINTEG statement, the INITIAL statement is ignored. Nonlinear restrictions on parameters are not supported.

The initial parameter values are the solution of the specified linear equations. If you do not specify initial values for all parameters, the default initial value for any parameter that is not specified in the INITIAL statement is 0, except for the following:

- The diagonal elements of the **COV** parameter matrix are set to ones if the **COV** parameter matrix is to be estimated.
- The diagonal elements of the **GCHC** parameter matrix are set to ones if the **GCHC** parameter matrix is to be estimated and the SUBFORM=EGARCH option is not specified.
- The diagonal elements of the **PACH** parameter matrix are set to ones if the SUBFORM=PGARCH option is specified.

The following is an example of the INITIAL statement for a bivariate (k=2) zero-mean VARMA(1,1) model, which is estimated by the maximum likelihood method by default because a moving average (MA) term is present:

```
proc varmax data=one;
   model y1 y2 / noint p=1 q=1;
   initial AR = 0, MA = 0,
           COV={1 0.5, 0.5 4};
run;
```

Like the RESTRICT statement, the preceding INITIAL statement can be abbreviated as follows:

```
initial AR = MA = 0,
        COV={1 0.5, 0.5 4};
```

or

```
initial AR, MA, COV={1 0.5, 0.5 4};
```

Furthermore, you can omit **AR** and **MA** in the INITIAL statement as follows, because by default the **AR** and **MA** matrices are set to zeros if they are not specified in the INITIAL statement:

```
initial COV={1 0.5, 0.5 4};
```

If you use the INITIAL statement for a vector error correction model (VECM), you must specify initial values for both the **ALPHA** and **BETA** matrices and make sure they are both full rank; otherwise, the INITIAL statement is ignored.

In the following example, the INITIAL statement is ignored because initial values for **ALPHA** and **BETA** are not specified:

```
proc varmax data=one;
   model y1 y2 / noint p=1;
   cointeg rank=1;
   initial cov=I(2)*4;
run;
```

In the following example, the INITIAL statement is ignored because initial values for **ALPHA** are not specified:

```
proc varmax data=one;
   model y1 y2 / noint p=1;
   cointeg rank=1;
   initial beta=1;
run;
```

In the following example, the INITIAL statement is ignored because the initial **BETA** matrix is not full rank:

```
proc varmax data=one;
   model y1 y2 y3 / noint p=1;
   cointeg rank=2;
   initial alpha={1 0, 0 1, 0 0},
           beta ={1 2, 2 4, 3 6};
run;
```

In the following example, the INITIAL statement works fine because the specified initial **ALPHA** and **BETA** matrices are both full rank:

```
proc varmax data=one;
   model y1 y2 y3 / noint p=1;
   cointeg rank=2;
   initial alpha={1 0, 0 1, 0 0},
           beta ={1 2, 2 4, 3 5};
run;
```

MODEL Statement

MODEL *dependents* < **=** *regressors* >
< , *dependents* < **=** *regressors* > . . . >
< / *options* > **;**

The MODEL statement specifies dependent (endogenous) variables and independent (exogenous) variables for the VARMAX model. The multivariate model can have the same or different independent variables corresponding to the dependent variables. As a special case, the VARMAX procedure allows you to analyze one dependent variable. Only one MODEL statement is allowed.

For example, the following statements are equivalent ways of specifying the multivariate model for the vector $(y1, y2, y3)$:

```
model y1 y2 y3 </options>;
model y1-y3 </options>;
```

The following statements are equivalent ways of specifying the multivariate model with independent variables, where $y1, y2, y3$, and $y4$ are the dependent variables and $x1, x2, x3, x4$, and $x5$ are the independent variables:

```
model y1 y2 y3 y4 = x1 x2 x3 x4 x5 </options>;
model y1 y2 y3 y4 = x1-x5 </options>;
model y1 = x1-x5, y2 = x1-x5, y3 y4 = x1-x5 </options>;
model y1-y4 = x1-x5 </options>;
```

When the multivariate model has different independent variables that correspond to each of the dependent variables, equations are separated by commas (,) and the model can be specified as illustrated by the following MODEL statement:

```
model y1 = x1-x3, y2 = x3-x5, y3 y4 = x1-x5 </options>;
```

The FI, PRIOR, and Q= options, the GARCH statement, and vector error correction models require that the same independent variables be used for all dependent variables. If you specify different independent variables that correspond to each of the dependent variables together with these options, statement, or models, all independent variables are dropped from the model. You can use the RESTRICT statement to achieve the goal when these options, statement, or models are specified. For example, if you need to specify x1 as the regressor of y1, x2 as the regressor of y2, and x3 as the regressor of y3 in a VMA(1) model, the following statement does not work:

```
model y1 = x1, y2 = x2, y3 = x3 / q=1;
```

But you can use the following statement to achieve your goal:

```
model y1 y2 y3 = x1 x2 x3 / q=1;
restrict x1(,y1,{x2 x3}) = 0,
         x1(,y2,{x1 x3}) = 0,
         x1(,y3,{x1 x2}) = 0;
```

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

CENTER

centers the dependent (endogenous) variables by subtracting their means. Note that centering is done after differencing when the DIF= or DIFY= option is specified. If there are exogenous (independent) variables, this option is not applicable.

```
model y1 y2 / p=1 center;
```

DIF(*variable* **(***number-list***) <...** *variable* **(***number-list***) >)**

DIF=(*variable* **(***number-list***) <...** *variable* **(***number-list***) >)**

specifies the degrees of differencing to be applied to the specified dependent or independent variables. The *number-list* must contain one or more numbers, each of which should be greater than zero. The differencing can be the same for all variables, or it can vary among variables. For example, the DIF=(y_1(1,4) y_3(1) x_2(2)) option specifies that the series y_1 is differenced at lag 1 and at lag 4, which is

$$(1 - B^4)(1 - B)y_{1t} = (y_{1t} - y_{1,t-1}) - (y_{1,t-4} - y_{1,t-5})$$

the series y_3 is differenced at lag 1, which is $(y_{3t} - y_{3,t-1})$; and the series x_2 is differenced at lag 2, which is $(x_{2t} - x_{2,t-2})$.

The following uses the data $dy1$, $y2$, $x1$, and $dx2$, where $dy1 = (1 - B)y_{1t}$ and $dx2 = (1 - B)^2 x_{2t}$:

```
model y1 y2 = x1 x2 / p=1 dif=(y1(1) x2(2));
```

DIFX(*number-list***)**

DIFX=(*number-list***)**

specifies the degrees of differencing to be applied to all independent variables. The *number-list* must contain one or more numbers, each of which should be greater than zero. For example, the DIFX=(1) option specifies that all of the independent series are differenced once at lag 1. The DIFX=(1,4) option specifies that all of the independent series are differenced at lag 1 and at lag 4. If independent variables are specified in the DIF= option, then the DIFX= option is ignored.

The following statement uses the data $y1$, $y2$, $dx1$, and $dx2$, where $dx1 = (1 - B)x_{1t}$ and $dx2 = (1 - B)x_{2t}$:

```
model y1 y2 = x1 x2 / p=1 difx(1);
```

DIFY(*number-list***)**

DIFY=(*number-list***)**

specifies the degrees of differencing to be applied to all dependent (endogenous) variables. The *number-list* must contain one or more numbers, each of which should be greater than zero. For more information, see the DIFX= option. If dependent variables are specified in the DIF= option, then the DIFY= option is ignored.

```
model y1 y2 / p=1 dify(1);
```

FI

uses the vector autoregressive fractionally integrated moving average model with exogenous variables.

```
model y1 y2 / fi method = ML;
```

METHOD=*value*

specifies the type of estimates to compute. You can specify the following *values*:

LS	specifies least squares estimates.
ML	specifies maximum likelihood estimates.
CML	specifies conditional maximum likelihood estimates.

For VARX models, you can apply the least squares method, maximum likelihood method, or conditional maximum likelihood method; for VARMAX models, you can apply either the maximum the likelihood method or the conditional maximum likelihood method; for other type of models, namely, vector error correction models, GARCH models, and Bayesian models, the default maximum likelihood method is applied. The (conditional) log-likelihood equations are solved by iterative numerical methods such as quasi-Newton optimization. The starting values for the AR and MA parameters are obtained from the least squares estimates. Although the small-sample properties of CML estimates might not be as good as the ML estimates, the CML method is much faster than the ML method.

```
model y1 y2 / p=1 method=ml;
```

NOCURRENTX

suppresses the current values x_t of the independent variables. In general, the VARX(p, s) model is

$$\mathbf{y}_t = \boldsymbol{\delta} + \sum_{i=1}^{p} \Phi_i \mathbf{y}_{t-i} + \sum_{i=0}^{s} \Theta_i^* \mathbf{x}_{t-i} + \boldsymbol{\epsilon}_t$$

where p is the number of lags of the dependent variables included in the model, and s is the number of lags of the independent variables included in the model, including the contemporaneous values of x_t.

A VARX(1,2) model can be specified as:

```
model y1 y2 = x1 x2 / p=1 xlag=2;
```

If the NOCURRENTX option is specified, it suppresses the current values $\mathbf{x}_t$ and starts with $\mathbf{x}_{t-1}$. The VARX(p, s) model is redefined as:

$$\mathbf{y}_t = \boldsymbol{\delta} + \sum_{i=1}^{p} \Phi_i \mathbf{y}_{t-i} + \sum_{i=1}^{s} \Theta_i^* \mathbf{x}_{t-i} + \boldsymbol{\epsilon}_t$$

This model with $p = 1$ and $s = 2$ can be specified as:

```
model y1 y2 = x1 x2 / p=1 xlag=2 nocurrentx;
```

NOINT

suppresses the intercept parameter $\boldsymbol{\delta}$.

```
model y1 y2 / p=1 noint;
```

NSEASON=*number*

specifies the number of seasonal periods. When the NSEASON=*number* option is specified, (*number* –1) seasonal dummies are added to the regressors. If the NOINT option is specified, the NSEASON= option is not applicable. For more information, see the section "Seasonal Dummies and Time Trends" on page 3073.

```
model y1 y2 / p=1 nseason=4;
```

SCENTER

centers seasonal dummies specified by the NSEASON= option. The centered seasonal dummies are generated by $c - (1/s)$, where c is a seasonal dummy generated by the NSEASON=s option.

```
model y1 y2 / p=1 nseason=4 scenter;
```

TREND=*value*

specifies the degree of deterministic time trend included in the model. Valid values are as follows:

LINEAR includes a linear time trend as a regressor.

QUAD includes linear and quadratic time trends as regressors.

The TREND=QUAD option is not applicable for a cointegration analysis. For more information, see the section "Seasonal Dummies and Time Trends" on page 3073.

```
model y1 y2 / p=1 trend=linear;
```

VARDEF=*value*

corrects for the degrees of freedom of the denominator for computing an error covariance matrix for the METHOD=LS option. If the METHOD=ML option is specified, the VARDEF=N option is always used. Valid values are as follows:

DF specifies that the number of nonmissing observation minus the number of regressors be used.

N specifies that the number of nonmissing observation be used.

```
model y1 y2 / p=1 vardef=n;
```

Printing Control Options

LAGMAX=*number*

specifies the maximum number of lags for which results are computed and displayed by the PRINT=(CORRX CORRY COVX COVY IARR IMPULSE= IMPULSX= PARCOEF PCANCORR PCORR) options. This option is also used to limit the printed results for the cross covariances and cross-correlations of residuals. The default is LAGMAX=min(12, T-2), where T is the number of nonmissing observations.

```
model y1 y2 / p=1 lagmax=6;
```

NOPRINT

suppresses all printed output.

```
model y1 y2 / p=1 noprint;
```

PRINTALL

requests all printing control options. The options set by the option PRINTALL are DFTEST=, MINIC=, PRINTFORM=BOTH, and PRINT=(CORRB CORRX CORRY COVB COVPE COVX COVY DECOMPOSE DYNAMIC IARR IMPULSE=(ALL) IMPULSX=(ALL) PARCOEF PCANCORR PCORR ROOTS YW).

You can also specify this option as the option ALL.

```
model y1 y2 / p=1 printall;
```

PRINTFORM=*value*

requests the printing format of the output generated by the PRINT= option and cross covariances and cross-correlations of residuals. Valid values are as follows:

BOTH	prints output in both MATRIX and UNIVARIATE forms.
MATRIX	prints output in matrix form. This is the default.
UNIVARIATE	prints output by variables.

```
model y1 y2 / p=1 print=(impulse) printform=univariate;
```

Printing Options

PRINT=(*options***)**

The following options can be used in the PRINT=( ) option. The options are listed within parentheses. If a number in parentheses follows an option listed below, then the option prints the number of lags specified by *number* in parentheses. The default is the number of lags specified by the LAGMAX=*number* option.

CORRB

prints the estimated correlations of the parameter estimates.

CORRX

CORRX(*number***)**

prints the cross-correlation matrices of exogenous (independent) variables. The *number* should be greater than zero.

CORRY

CORRY(*number***)**

prints the cross-correlation matrices of dependent (endogenous) variables. The *number* should be greater than zero.

COVB

prints the estimated covariances of the parameter estimates.

COVPE

COVPE(*number*)

prints the covariance matrices of *number*-ahead prediction errors for the VARMAX(p,q,s) model. The *number* should be greater than zero. If the DIF= or DIFY= option is specified, the covariance matrices of multistep prediction errors are computed based on the differenced data. This option is not applicable when the PRIOR= option is specified. For more information, see the section "Forecasting" on page 3057.

COVX

COVX(*number*)

prints the cross-covariance matrices of exogenous (independent) variables. The *number* should be greater than zero.

COVY

COVY(*number*)

prints the cross-covariance matrices of dependent (endogenous) variables. The *number* should be greater than zero.

DECOMPOSE

DECOMPOSE(*number*)

prints the decomposition of the prediction error covariances using up to the number of lags specified by *number* in parentheses for the VARMA(p,q) model. The *number* should be greater than zero. It can be interpreted as the contribution of innovations in one variable to the mean squared error of the multistep forecast of another variable. The DECOMPOSE option also prints proportions of the forecast error variance.

If the DIF= or DIFY= option is specified, the covariance matrices of multistep prediction errors are computed based on the differenced data. This option is not applicable when the PRIOR= option is specified. For more information, see the section "Forecasting" on page 3057.

DIAGNOSE

prints the residual diagnostics and model diagnostics.

DYNAMIC

prints the contemporaneous relationships among the components of the vector time series.

ESTIMATES

prints the coefficient estimates and a schematic representation of the significance and sign of the parameter estimates.

IARR

IARR(*number*)

prints the infinite order AR representation of a VARMA process. The *number* should be greater than zero. If the ECM= option or the COINTEG statement is specified, then the reparameterized AR coefficient matrices are printed.

IMPULSE
IMPULSE(*number*)
IMPULSE=(SIMPLE ACCUM ORTH STDERR ALL)
IMPULSE(*number*)=(SIMPLE ACCUM ORTH STDERR ALL)

prints the impulse response function. The *number* should be greater than zero. It investigates the response of one variable to an impulse in another variable in a system that involves a number of other variables as well. It is an infinite order MA representation of a VARMA process. For more information, see the section "Impulse Response Function" on page 3046.

You can specify the following options within parentheses:

ACCUM prints the accumulated impulse response function.

ALL is equivalent to specifying SIMPLE, ACCUM, ORTH, and STDERR.

ORTH prints the orthogonalized impulse response function.

SIMPLE prints the impulse response function. This is the default.

STDERR prints the standard errors of the impulse response function, the accumulated impulse response function, or the orthogonalized impulse response function.

IMPULSX
IMPULSX(*number*)
IMPULSX=(SIMPLE ACCUM STDERR ALL)
IMPULSX(*number*)=(SIMPLE ACCUM STDERR ALL)

prints the impulse response function related to exogenous (independent) variables. The *number* should be greater than zero. For more information, see the section "Impulse Response Function" on page 3046.

You can specify the following options within parentheses:

ACCUM prints the accumulated impulse response matrices for the transfer function.

ALL is equivalent to specifying SIMPLE, ACCUM, and STDERR.

SIMPLE prints the impulse response matrices for the transfer function.

STDERR prints the standard errors of the simple impulse response function or the accumulated impulse response function.

By default, IMPULSX(*number*)=(SIMPLE).

PARCOEF
PARCOEF(*number*)

prints the partial autoregression coefficient matrices, Φ_{mm} up to the lag *number*. The *number* should be greater than zero. With a VAR process, this option is useful for the identification of the order since the Φ_{mm} have the property that they equal zero for $m > p$ under the hypothetical assumption of a VAR(p) model. For more information, see the section "Tentative Order Selection" on page 3062.

PCANCORR
PCANCORR(*number*)

prints the partial canonical correlations of the process at lag m and the test for testing Φ_m=0 for $m > p$ up to the lag *number*. The *number* should be greater than zero. The lag m partial canonical correlations are the canonical correlations between $\mathbf{y}_t$ and $\mathbf{y}_{t-m}$, after adjustment for the dependence of these variables on the intervening values $\mathbf{y}_{t-1}, \ldots, \mathbf{y}_{t-m+1}$. For more information, see the section "Tentative Order Selection" on page 3062.

PCORR

PCORR(*number***)**

prints the partial correlation matrices. The *number* should be greater than zero. With a VAR process, this option is useful for a tentative order selection by the same property as the partial autoregression coefficient matrices, as described in the PRINT=(PARCOEF) option. For more information, see the section "Tentative Order Selection" on page 3062.

ROOTS

prints the eigenvalues of the $kp \times kp$ companion matrix associated with the AR characteristic function $\Phi(B)$, where k is the number of dependent (endogenous) variables, and $\Phi(B)$ is the finite order matrix polynomial in the backshift operator B, such that $B^i \mathbf{y}_t = \mathbf{y}_{t-i}$. These eigenvalues indicate the stationary condition of the process since the stationary condition on the roots of $|\Phi(B)| = 0$ in the VAR(p) model is equivalent to the condition in the corresponding VAR(1) representation that all eigenvalues of the companion matrix be less than one in absolute value. Similarly, you can use this option to check the invertibility of the MA process. In addition, when the GARCH statement is specified, this option prints the roots of the GARCH characteristic polynomials to check covariance stationarity for the GARCH process.

YW

prints Yule-Walker estimates of the preliminary autoregressive model for the dependent (endogenous) variables. The coefficient matrices are printed using the maximum order of the autoregressive process.

Some examples of the PRINT= option are as follows:

```
model y1 y2 / p=1 print=(covy(10) corry(10));
model y1 y2 / p=1 print=(parcoef pcancorr pcorr);
model y1 y2 / p=1 print=(impulse(8) decompose(6) covpe(6));
model y1 y2 / p=1 print=(dynamic roots yw);
```

Lag Specification Options

P=*number*

P=(*number-list***)**

specifies the order of the vector autoregressive process. Subset models of vector autoregressive orders can be specified by listing the desired set of lags. For example, you can specify the P=(1,3,4) option. The P=3 option is equivalent to the P=(1,2,3) option. The default is P=0.

If P=0 and there are no exogenous (independent) variables, then the AR polynomial order is automatically determined by minimizing an information criterion. If P=0 and the PRIOR= or ECM= option or COINTEG statement are specified, then the AR polynomial order is determined automatically.

If the ECM= option or the COINTEG statement is specified, then subset models of vector autoregressive orders are not allowed and the AR maximum order specified is used.

Examples illustrating the P= option follow:

```
model y1 y2 / p=3;
model y1 y2 / p=(1,3);
model y1 y2 / p=(1,3) prior;
```

Q=*number*

Q=(*number-list***)**

specifies the order of the moving-average error process. Subset models of moving-average orders can be specified by listing the desired set of lags. For example, you can specify the Q=(1,5) option. The default is Q=0.

```
model y1 y2 / p=1 q=1;
model y1 y2 / q=(2);
```

XLAG=*number*

XLAG=(*number-list***)**

specifies the lags of exogenous (independent) variables. Subset models of distributed lags can be specified by listing the desired set of lags. For example, XLAG=(2) selects only a lag 2 of the exogenous variables. The default is XLAG=0. To exclude the present values of exogenous variables from the model, the NOCURRENTX option must be used.

```
model y1 y2 = x1-x3 / xlag=2 nocurrentx;
model y1 y2 = x1-x3 / p=1 xlag=(2);
```

Tentative Order Selection Options

MINIC

MINIC=(P=*number* **PERROR=***number* **Q=***number* **TYPE=***value* **)**

prints the information criterion for the appropriate AR and MA tentative order selection.

You can specify the following options within parentheses in the MINIC= option:

P=*number*

P=(p_{min} **:** p_{max}**)**

specifies the range of AR orders to be considered in the tentative order selection. The default is P=(0:5). P=3 is equivalent to P=(0:3).

PERROR=*number*

PERROR=($p_{\epsilon,min}$ **:** $p_{\epsilon,max}$**)**

specifies the range of AR orders for obtaining the error series. The default is PERROR=(p_{max} : $p_{max} + q_{max}$).

Q=*number*

Q=(q_{min} **:** q_{max}**)**

specifies the range of MA orders to be considered in the tentative order selection. The default is Q=(0:5).

TYPE= AIC | **AICC** | **FPE** | **HQC** | **SBC**

specifies the criterion for the model order selection. Valid criteria are as follows:

AIC	specifies Akaike's information criterion.
AICC	specifies the corrected Akaike's information criterion.
FPE	specifies the final prediction error criterion.
HQC	specifies the Hanna-Quinn criterion.
SBC	specifies the Schwarz Bayesian criterion. You can also specify this value as TYPE=BIC.

By default, TYPE=AICC.

The following examples show how to use the MINIC or MINIC= option:

```
model y1 y2 / minic;
```

```
model y1 y2 / minic=(type=aic p=13);
```

In the selection of AR and MA orders, the model that has the smallest criterion value is chosen. For the definitions of the information criteria used in the MINIC option, see the section "The Minimum Information Criterion (MINIC) Method" on page 3066.

Cointegration Related Options

Two options are related to integrated time series; one is the DFTEST option to test for a unit root and the other is the COINTTEST option to test for cointegration.

DFTEST

DFTEST=(DLAG=*number***)**

DFTEST=(DLAG=(*number***) . . . (***number***))**

prints the Dickey-Fuller unit root tests. The DLAG=(*number*) . . . (*number*) option specifies the regular or seasonal unit root test. Supported values of *number* are in 1, 2, 4, 12. If the *number* is greater than one, a seasonal Dickey-Fuller test is performed. If the TREND= option is specified, the seasonal unit root test is not available. The default is DLAG=1.

For example, the DFTEST=(DLAG=(1)(12)) option produces two tables: the Dickey-Fuller regular unit root test and the seasonal unit root test.

Some examples of the DFTEST= option follow:

```
model y1 y2 / p=2 dftest;
model y1 y2 / p=2 dftest=(dlag=4);
model y1 y2 / p=2 dftest=(dlag=(1)(12));
model y1 y2 / p=2 dftest cointtest;
```

COINTTEST

COINTTEST=(JOHANSEN <(=*options***)> SW <(=***options***)> SIGLEVEL=***number* **)**

specifies the cointegration tests.

You can specify the following options within parentheses in the COINTTEST= option:

JOHANSEN

JOHANSEN=(TYPE=*value* **IORDER=***number* **NORMALIZE=***variable***)**

prints the cointegration rank test for multivariate time series based on Johansen's method. This test is provided when the number of dependent (endogenous) variables is less than or equal to 64. For more information, see the section "Vector Error Correction Modeling" on page 3090.

The VARX(p,s) model can be written as the error correction model

$$\Delta \mathbf{y}_t = \Pi \mathbf{y}_{t-1} + \sum_{i=1}^{p-1} \Phi_i^* \Delta \mathbf{y}_{t-i} + A D_t + \sum_{i=0}^{s} \Theta_i^* \mathbf{x}_{t-i} + \boldsymbol{\epsilon}_t$$

where Π, Φ_i^*, A, and Θ_i^* are coefficient parameters and D_t is a deterministic term such as a constant, a linear trend, or seasonal dummies.

The $I(1)$ model is defined by one reduced-rank condition. If the cointegration rank is $r < k$, then there exist $k \times r$ matrices $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ of rank r such that $\Pi = \boldsymbol{\alpha}\boldsymbol{\beta}'$.

The $I(1)$ model is rewritten as the $I(2)$ model

$$\Delta^2 \mathbf{y}_t = \Pi \mathbf{y}_{t-1} - \Psi \Delta \mathbf{y}_{t-1} + \sum_{i=1}^{p-2} \Psi_i \Delta^2 \mathbf{y}_{t-i} + A D_t + \sum_{i=0}^{s} \Theta_i^* \mathbf{x}_{t-i} + \boldsymbol{\epsilon}_t$$

where $\Psi = I_k - \sum_{i=1}^{p-1} \Phi_i^*$ and $\Psi_i = -\sum_{j=i+1}^{p-1} \Phi_i^*$.

The $I(2)$ model is defined by two reduced-rank conditions. One is that $\Pi = \boldsymbol{\alpha}\boldsymbol{\beta}'$, where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are $k \times r$ matrices of full-rank r. The other is that $\boldsymbol{\alpha}_\perp' \Psi \boldsymbol{\beta}_\perp = \boldsymbol{\xi}\boldsymbol{\eta}'$, where $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ are $(k-r) \times s$ matrices with $s \leq k - r$, and $\boldsymbol{\alpha}_\perp$ and $\boldsymbol{\beta}_\perp$ are $k \times (k-r)$ matrices of full-rank $k - r$ such that $\boldsymbol{\alpha}'\boldsymbol{\alpha}_\perp = 0$ and $\boldsymbol{\beta}'\boldsymbol{\beta}_\perp = 0$.

You can specify the following options within parentheses in the JOHANSEN= option:

IORDER=1 | 2

specifies the integrated order. You can specify the following values:

1 prints the cointegration rank test for an integrated order 1 and prints the long-run parameter, $\boldsymbol{\beta}$, and the adjustment coefficient, $\boldsymbol{\alpha}$. If you specify IORDER=1, then the AR order should be greater than or equal to 1. If you specify P=0 in the MODEL statement, the value of P is set to 1 for the Johansen test.

2 prints the cointegration rank test for integrated orders 1 and 2. If you specify IORDER=2, then the AR order should be greater than or equal to 2. If you specify P=1 and IORDER=2, then the value of IORDER is set to 1; if you specify P=0 and IORDER=2, then the value of P is set to 2.

By default, IORDER=1.

NORMALIZE=*variable*

specifies the dependent (endogenous) *variable* whose cointegration vectors are to be normalized. If the *variable* is different from the variable specified in the COINTEG statement or in the ECM= option in the MODEL statement, then the value specified in the COINTEG statement is used. If you specify this option and you want to estimate an error correction

model, then the BOUND, GARCH, INITIAL, and RESTRICT statements are all ignored and the Q= option in the MODEL statement is also ignored.

TYPE=MAX | TRACE
specifies the type of cointegration rank test to be printed. You can specify the following values:

MAX prints the cointegration maximum eigenvalue test.

TRACE prints the cointegration trace test.

By default, TYPE=TRACE. If the NOINT option is not specified, PROC VARMAX prints two different cointegration rank tests in the presence of the unrestricted and restricted deterministic terms (constant or linear trend) models. If you specify IORDER=2, the procedure automatically sets the TYPE=TRACE option.

The following examples illustrate the JOHANSEN= option:

```
model y1 y2 / p=2 cointtest=(johansen=(type=max normalize=y1));

model y1 y2 / p=2 cointtest=(johansen=(iorder=2 normalize=y1));
```

SIGLEVEL=*value*
sets the size (the significance level) of the common trends tests.

The SIGLEVEL=*value* can be set to 0.1, 0.05, or 0.01. By default, SIGLEVEL=0.05.

```
model y1 y2 / p=2 cointtest=(sw siglevel=0.1);

model y1 y2 / p=2 cointtest=(sw siglevel=0.01);
```

SW

SW=(TYPE=*value* **LAG=***number***)**
prints common trends tests for a multivariate time series based on the Stock-Watson method. This test is provided when the number of dependent (endogenous) variables is less than or equal to 6. For more information, see the section "Common Trends" on page 3088.

You can specify the following options within parentheses in the SW= option:

LAG=*number*
specifies the number of lags. The default is LAG=max(1,p) for the TYPE=FILTDIF or TYPE=FILTRES option, where p is the AR maximum order specified by the P= option. The default is LAG=$T^{1/4}$ for the TYPE=KERNEL option, where T is the number of nonmissing observations. If the specified LAG=*number* exceeds the default, then it is replaced by the default.

TYPE=FILTDIF | FILTRES | KERNEL
specifies the type of common trends test to be printed. You can specify the following values:

FILTDIF prints the common trends test based on the filtering method applied to the differenced series.

FILTRES prints the common trends test based on the filtering method applied to the residual series.

KERNEL prints the common trends test based on the kernel method.

By default, TYPE=FILTDIF.

The following examples illustrate the SW option:

```
model y1 y2 / p=2 cointtest=(sw);

model y1 y2 / p=2 cointtest=(sw=(type=kernel));

model y1 y2 / p=2 cointtest=(sw=(type=kernel lag=3));
```

Bayesian VARX Estimation Options

PRIOR

PRIOR=(*prior-options***)**

specifies the prior value of parameters for the BVARX(p, s) model. The BVARX model allows for a subset model specification. If the ECM= option or the COINTEG statement is specified with the PRIOR option, the BVECMX(p, s) form is fitted. When the PRIOR option is specified, the Q= option in the MODEL statement is ignored, and the BOUND, GARCH, INITIAL, RESTRICT, and TEST statements are all ignored. For more information, see the section "Bayesian VAR and VARX Modeling" on page 3074.

The following options can be used with the PRIOR=(*prior-options*) option. The *prior-options* are listed within parentheses.

IVAR

IVAR=(*variables***)**

specifies an integrated BVAR(p) model. The *variables* should be specified in the MODEL statement as dependent variables. If you use the IVAR option without *variables*, then it sets the overall prior mean of the first lag of each variable equal to one in its own equation and sets all other coefficients to zero. If *variables* are specified, it sets the prior mean of the first lag of the specified variables equal to one in its own equation and sets all other coefficients to zero. When the series $\mathbf{y}_t = (y_1, y_2)'$ follows a bivariate BVAR(2) process, the IVAR or IVAR=(y_1 y_2) option is equivalent to specifying MEAN=(1 0 0 0 0 1 0 0).

If the PRIOR=(MEAN=) or ECM= option or the COINTEG statement is specified, the IVAR= option is ignored.

LAMBDA=*value*

specifies the prior standard deviation of the AR coefficient parameter matrices. It should be a positive number. The default is LAMBDA=1. As the value of the LAMBDA= option is increased, the BVAR(p) model becomes closer to a VAR(p) model.

MEAN=(*vector*)

specifies the mean vector in the prior distribution for the AR coefficients. If the vector is not specified, the prior value is assumed to be a zero vector. For more information, see the section "Bayesian VAR and VARX Modeling" on page 3074.

You can specify the mean vector by order of the equation. Let $(\delta, \Phi_1, \ldots, \Phi_p)$ be the parameter sets to be estimated and $\Phi = (\Phi_1, \ldots, \Phi_p)$ be the AR parameter sets. The mean vector is specified by row-wise from Φ; that is, the MEAN=$(\mathrm{vec}(\Phi'))$ option.

For the PRIOR=(mean) option in the BVAR(2),

$$\Phi = \begin{pmatrix} \phi_{1,11} & \phi_{1,12} & \phi_{2,11} & \phi_{2,12} \\ \phi_{1,21} & \phi_{1,22} & \phi_{2,21} & \phi_{2,22} \end{pmatrix} = \begin{pmatrix} 2 & 0.1 & 1 & 0 \\ 0.5 & 3 & 0 & -1 \end{pmatrix}$$

where $\phi_{l,ij}$ is an element of Φ, l is a lag, i is associated with the first dependent variable, and j is associated with the second dependent variable.

```
model y1 y2 / p=2 prior=(mean=(2 0.1 1 0 0.5 3 0 -1));
```

The deterministic terms and exogenous variables are considered to shrink toward zero; you must omit prior means of exogenous variables and deterministic terms such as a constant, seasonal dummies, or trends.

For a Bayesian error correction model estimated when both the ECM= option (or the COINTEG statement) and the PRIOR= option are used, a mean vector for only lagged AR coefficients, Φ_i^*, in terms of regressors $\Delta \mathbf{y}_{t-i}$, for $i = 1, \ldots, (p-1)$ is used in the VECM(p) representation. The diffused prior variance of $\boldsymbol{\alpha}$ is used, since $\boldsymbol{\beta}$ is replaced by $\hat{\boldsymbol{\beta}}$ estimated in a nonconstrained VECM(p) form.

$$\Delta \mathbf{y}_t = \boldsymbol{\alpha} \mathbf{z}_{t-1} + \sum_{i=1}^{p-1} \Phi_i^* \Delta \mathbf{y}_{t-i} + AD_t + \sum_{i=0}^{s} \Theta_i^* \mathbf{x}_{t-i} + \boldsymbol{\epsilon}_t$$

where $\mathbf{z}_t = \boldsymbol{\beta}' \mathbf{y}_t$.

For example, in the case of a bivariate ($k = 2$) BVECM(2) form, the option

$$\mathrm{MEAN} = (\phi_{1,11}^* \; \phi_{1,12}^* \; \phi_{1,21}^* \; \phi_{1,22}^*)$$

where $\phi_{1,ij}^*$ is the (i,j) element of the matrix Φ_1^*.

NREP=*number*

determines the number of repetitions that are used to compute the measure of forecast accuracy. For more information, see the equation in the section "Forecasting of BVAR Modeling" on page 3076. The default is NREP=$0.5T$, where T is the number of observations. If NREP is above $0.5T$, it is decreased to $0.5T$; if NREP is below the value of the LEAD= option, it is increased to the value of the LEAD= option.

THETA=*value*

specifies the prior standard deviation of the AR coefficient parameter matrices. The *value* is in the interval (0,1). The default is THETA=0.1. As the value of the THETA= option approaches 1, the specified BVAR(p) model approaches a VAR(p) model.

Some examples of the PRIOR= option follow:

```
model y1 y2 / p=2 prior;
model y1 y2 / p=2 prior=(theta=0.2 lambda=5);
model y1 y2 = x1 / p=2 prior=(theta=0.2 lambda=5);
model y1 y2 = x1 / p=2
   prior=(theta=0.2 lambda=5 mean=(2 0.1 1 0 0.5 3 0 -1));
```

For more information, see the section "Bayesian VAR and VARX Modeling" on page 3074.

Vector Error Correction Model Options

ECM=(RANK=*number* **<ECTREND><NORMALIZE=***variable***>)**

specifies a vector error correction model.

The ECM= option is obsolete. Use the COINTEG statement instead.

You must specify the following option within parentheses in the ECM= option:

RANK=*number*

specifies the cointegration rank of the cointegrated system. The rank of cointegration should be greater than 0 and less than the number of dependent (endogenous) variables. If *number* is different from the RANK= option specified in the COINTEG statement, the value specified in the COINTEG statement is used for the rank.

You can also specify the following options within parentheses in the ECM= option:

ECTREND

specifies the restriction on the drift in the VECM. This option is used in the following cases:

- There is no separate drift in the VECM, but a constant enters only through the error correction term. For example, for VECM(p),

$$\Delta \mathrm{y}_t = \boldsymbol{\alpha}(\boldsymbol{\beta}', \boldsymbol{\beta}_0)(\mathrm{y}'_{t-1}, 1)' + \sum_{i=1}^{p-1} \Phi_i^* \Delta \mathrm{y}_{t-i} + \boldsymbol{\epsilon}_t$$

  An example of the ECTREND option follows:

```
model y1 y2 / p=2 ecm=(rank=1 ectrend);
```

- There is a separate drift and no separate linear trend in the VECM, but a linear trend enters only through the error correction term. For example, for VECM(p),

$$\Delta \mathrm{y}_t = \boldsymbol{\alpha}(\boldsymbol{\beta}', \boldsymbol{\beta}_1)(\mathrm{y}'_{t-1}, t)' + \sum_{i=1}^{p-1} \Phi_i^* \Delta \mathrm{y}_{t-i} + \boldsymbol{\delta}_0 + \boldsymbol{\epsilon}_t$$

  An example of the ECTREND option with the TREND= option follows:

```
model y1 y2 / p=2 ecm=(rank=1 ectrend) trend=linear;
```

If you specify both this option and the NSEASON option in the MODEL statement, then the NSEASON option is ignored. If you specify the NOINT option in the MODEL statement, then this option is ignored.

NORMALIZE=*variable*
specifies a single dependent (endogenous) *variable* whose cointegrating vectors are normalized. If the *variable* is different from the variable specified in the NORMALIZE= option in the COINTEG statement, the variable specified in the NORMALIZE= option in the COINTEG statement is used. If this option is not specified, cointegrating vectors are not normalized. If you specify this option, then the BOUND, GARCH, INITIAL, and RESTRICT statements are all ignored and the Q= option in the MODEL statement is also ignored.

The following examples illustrate the ECM= option:

```
model y1 y2 / p=2 ecm=(rank=1 normalize=y1);
```

```
model y1 y2 / p=2 ecm=(rank=1 ectrend) trend=linear;
```

For more information, see the section "Vector Error Correction Modeling" on page 3090.

NLOPTIONS Statement

NLOPTIONS *options* **;**

The VARMAX procedure uses the nonlinear optimization (NLO) subsystem to perform nonlinear optimization tasks. For a list of all the options in the NLOPTIONS statement, see Chapter 6, "Nonlinear Optimization Methods."

An example of the NLOPTIONS statement is as follows:

```
proc varmax data=one;
   nloptions tech=qn maxit=1000 pall;
   model y1 y2 / p=2;
run;
```

By default, the VARMAX procedure uses the dual quasi-Newton optimization method.

OUTPUT Statement

OUTPUT < *options* > **;**

The OUTPUT statement generates and prints forecasts based on the model estimated in the previous MODEL statement and, optionally, creates an output SAS data set that contains these forecasts.

When the GARCH model is estimated, the upper and lower confidence limits of forecasts are calculated according to the conditional covariance of errors.

ALPHA=*number*
sets the forecast confidence limit size, where *number* is between 0 and 1. When you specify the ALPHA=*number* option, the upper and lower confidence limits define the $100(1-\alpha)\%$ confidence interval. The default is ALPHA=0.05, which produces 95% confidence intervals.

BACK=*number*
specifies the number of observations before the end of the data at which the multistep forecasts begin. The BACK= option value must be less than or equal to the number of observations minus the number of lagged regressors in the model. The default is BACK=0, which means that the forecasts start at the end of the available data.

LEAD=*number*
specifies the number of multistep forecast values to compute. The default is LEAD=12.

NOPRINT
suppresses the printed forecast values of each dependent (endogenous) variable.

OUT=*SAS-data-set*
writes the forecast values to an output data set. If the OUT= option is not included in the OUTPUT statement, then the output data set is named using the DATA*n* naming convention.

Some examples of the OUTPUT statements follow:

```
proc varmax data=one;
   model y1 y2 / p=2;
   output lead=6 back=2;
run;
```

```
proc varmax data=one;
   model y1 y2 / p=2;
   output out=for noprint;
run;
```

RESTRICT Statement

RESTRICT *restriction, . . . , restriction* **;**

The RESTRICT statement places linear restrictions on the parameters and provides constrained estimation. Only one RESTRICT statement is allowed. If you specify more than one *restriction* in a RESTRICT statement, separate them with commas. Both equality and inequality constraints are allowed in the RESTRICT statement, although usually equality constraints are specified in the RESTRICT statement and inequality constraints are specified in the BOUND statement. If the least squares method is used, the inequality constraints are not applicable.

To use the RESTRICT statement, you need to know the form of the model. If you do not specify the GARCH statement, the COINTEG statement, or the ECM=, P=, Q=, or XLAG= option in the MODEL statement then the RESTRICT statement is not applicable. If you specify the ECM=(NORMALIZE=) option or PRIOR= option in the MODEL statement or if you specify the EXOGENEITY, H=, J=, or NORMALIZE= option in

the COINTEG statement, then the RESTRICT statement is ignored. Nonlinear restrictions on parameters are not supported.

Restricted parameter estimates are computed by introducing a Lagrangian parameter for each restriction (Pringle and Rayner 1971). The Lagrangian parameter measures the sensitivity of the sum of squared errors to the restriction. The estimates of these Lagrangian parameters and their significance are printed in the Restrict ODS table.

Matrix Expression

The RESTRICT statement operates on matrices. That is, you can specify the parameter matrices or constant matrices through the RESTRICT statement's built-in operators and functions. You can add elements of the matrices **A** and **B** with the expression **A+B**, and you can perform matrix multiplication with the expression **A*B** and elementwise multiplication with the expression **A#B**. You can get the diagonal elements of the matrix **A** through the function **DIAG(A)**, and you can get the $n \times n$ identity matrix through the function **I(n)**.

Each restriction is written as a matrix expression composed of constants, operators, and functions.

Constants

Constants are either scalar constants (such as –1.2, 0.3, and so on) or matrix constants enclosed in braces (such as the 2×2 matrix, **{1 2, 3 4}**, or the 1×3 row vector, **{-0.2 5.3 12}**). Constants also include the dependent variable names and exogenous variable names that represent their index values and are mostly used in the subscripts or function arguments. For example, in the following PROC VARMAX statements, the dependent and exogenous variables have the following index values (based on their orders in the MODEL statement): GDP is equal to 1, CPI to 2, M2 to 3, FFR to 1, and CP to 2. Hence, the function call **AR(2, GDP, {CPI M2})** is equivalent to **AR(2,1,{2 3})**, and **XL(0, CPI, {FFR CP})** is equivalent to **XL(0,2,{1 2})**. For more information about the use of **AR** and **XL** functions to access parameters, see the section "Functions" on page 3028.

```
proc varmax data=macrodata;
   model GDP CPI M2 = FFR CP / p=12 xlag=12;
   restrict AR(2, GDP, {CPI M2}) = 0,
            XL(0, CPI, {FFR CP}) = 0;
run;
```

The matrix constant cannot be the first item in the RESTRICT statement. For example, you cannot specify the following statement:

```
restrict {-0.1 -0.2, -0.3 -0.4} <= AR <= {0.1 0.2, 0.3 0.4};
```

However, you can put the first matrix constant in parentheses and specify the preceding example in the following way:

```
restrict ({-0.1 -0.2, -0.3 -0.4}) <= AR <= {0.1 0.2, 0.3 0.4};
```

Operators

Operators define the operations on operands. Table 42.2 lists all built-in operators supported by the RESTRICT statement.

Table 42.2 Operators

Operator	Name	Description
+	Addition	Adds corresponding matrix elements
=	Comparison, equal	Compares matrix elements
<	Comparison, less than	Compares matrix elements
<=	Comparison, not greater than	Compares matrix elements
>	Comparison, greater than	Compares matrix elements
>=	Comparison, not less than	Compares matrix elements
||	Concatenation, horizontal	Concatenates matrices horizontally
//	Concatenation, vertical	Concatenates matrices vertically
@	Direct product	Takes the direct product of two matrices
:	Index creation	Creates an index vector
#	Multiplication, elementwise	Performs elementwise multiplication
*	Multiplication, matrix	Performs matrix multiplication
−	Sign reverse	Reverses the signs of elements
[ ]	Subscripts	Selects submatrices
−	Subtraction	Subtracts corresponding matrix elements
`	Transpose	Transposes a matrix

For more information about each operator, see the section "Details of Operators" on page 3032.

Table 42.3 shows the precedence of matrix operators in the RESTRICT statement.

Table 42.3 Operator Precedence

Priority Group	Operators		
I (highest)	[ ] (subscripts)	` (transpose)	
II	− (sign reverse)		
III	*	#	@
IV	− (subtraction)	+	
V	||	//	:
VI (lowest)	=	<	<= > >=

Each restriction can be a compound expression that involves several matrix operators and operands. The rules for evaluating compound expressions are as follows:

- Evaluation follows the order of operator precedence, as described in Table 42.3. Group I has the highest priority; that is, Group I operators are evaluated first. Group II operators are evaluated after Group I operators, and so on. For example, $1 + 2 * 3$ returns 7.
- If neighboring operators in an expression have equal precedence, the expression is evaluated from left to right, except for the Group I operators. For example, $1 - 2 - 3$ returns -4.
- All expressions in parentheses are evaluated first, following the two preceding rules. For example, $3 * (2 + 1)$ returns 9.

Functions

Functions are mainly divided into two categories: one type of function refers to parameters to be estimated, such as **AR(L,I,J)** and **CCC(I,J)**; the other type does not, such as **I(n)** and **DIAG(A)**.

Functions that refer to the parameters are listed in Table 42.4. The arguments for functions can be matrices. The simplest case, scalar arguments, is discussed first. For convenience, the scalar indices **i** and **j** refer to the position of the element in the coefficient matrix, and scalar **l** refers to the lag value.

Table 42.4 Functions That Refer to Parameters

Function	Description
ACH(l,i,j)	ARCH parameter of the lag l value of $\boldsymbol{\epsilon}_t \boldsymbol{\epsilon}_t'$ in a GARCH model
ALPHA(i,j)	The (i, j) element in the adjustment coefficient matrix $\boldsymbol{\alpha}$ for the vector error correction model
AR(l,i,j)	Autoregressive parameter of the lag l value of the jth dependent (endogenous) variable, $y_{j,t-l}$, to the ith dependent variable at time t, y_{it} for models other than error-correction models. For error correction models, **AR(1,i,j)** is the (i, j) element in $\Pi(= \boldsymbol{\alpha}\boldsymbol{\beta}')$ for $\mathbf{y}_{t-1}$, and **AR(l,i,j)**, $l > 1$, is the autoregressive parameter of the lag $(l-1)$ value of the jth differenced dependent (endogenous) variable, $\Delta y_{j,t-(l-1)}$, to the ith differenced dependent variable at time t, Δy_{it}.
BETA(i,j)	The (i, j) element in the cointegrating matrix $\boldsymbol{\beta}$ for the vector error correction model
CCC(i,j)	Constant conditional correlation parameter between the ith and jth standardized error processes for the CCC GARCH model
CONST(i)	Intercept parameter of the ith time series, y_{it}
COV(i,j)	Covariance of innovations parameter between the ith and jth error processes when the maximum likelihood method is used for the fitted non-GARCH model
D(i)	Long-range dependent parameter of the ith time series, y_i, when the FI option is specified. By default, the LRD parameters are restricted between –1/2 and 1/2.
DCCA()	Parameter α in the correlation equation for the DCC GARCH model
DCCB()	Parameter β in the correlation equation for the DCC GARCH model
DCCS(i,j)	Unconditional correlation parameter between the ith and jth standardized error processes for the DCC GARCH model
EACH(l,i,j)	Exponential ARCH parameter of the lag l value of $\epsilon_{it}/\sigma_{it}$ in the CCC or DCC GARCH model when SUBFORM=EGARCH is specified and $i = j$. If $i \neq j$, the value is set to 0.
ECCONST(i)	The ith element for the constant in the error correction term for the vector error correction model when the ECTREND option in the COINTEG statement is specified
ECLTREND(i)	The ith element for the linear trend in the error correction term for vector error correction model when the ECTREND option in the COINTEG statement is specified
GCH(l,i,j)	GARCH parameter of the lag l value of the covariance matrix, H_t, in a GARCH model
GCHC(i,j)	Constant parameter of the covariance matrix, H_t, in a GARCH model

Table 42.4 *continued*

Function	Description
`LAMBDA(i)`	Power parameter for the *i*th error process in the CCC or DCC GARCH model when SUBFORM=PGARCH is specified
`LTREND(i)`	Linear trend parameter of the *i*th time series, y_{it}, when the TREND= option is specified
`MA(l,i,j)`	Moving average parameter of the lag l value of the *j*th error process, $\epsilon_{j,t-l}$, to the *i*th dependent variable at time t, y_{it}
`PACH(l,i,j)`	Power ARCH parameter of the lag l value of ϵ_{it} in the CCC or DCC GARCH model when SUBFORM=PGARCH is specified and $i = j$. If $i \neq j$, the value is set to 0.
`QACH(l,i,j)`	Quadratic ARCH center parameter of the lag l value of ϵ_{it} in the CCC or DCC GARCH model when SUBFORM=QGARCH is specified and $i = j$. If $i \neq j$, the value is set to 0.
`QTREND(i)`	Quadratic trend parameter of the *i*th time series, y_{it}, when TREND=QUAD is specified
`SD(i,j)`	Same as `SDUMMY(i, j)`
`SDUMMY(i,j)`	The *j*th seasonal dummy of the *i*th time series at time t, y_{it}, where $j = 1, \ldots, (nseason-1)$, where *nseason* is the value of the NSEASON= option in the MODEL statement
`TACH(l,i,j)`	Threshold ARCH parameter of the lag l value of $1_{\epsilon_{it}<0}\epsilon_{it}^2$ in the CCC or DCC GARCH model when SUBFORM=GJR is specified and $i = j$. If $i \neq j$, the value is set to 0.
`XL(l,i,j)`	Exogenous parameter of the lag l value of the *j*th exogenous (independent) variable, $x_{j,t-l}$, to the *i*th dependent variable at time t, y_{it}

The functions that refer to parameters, as shown in Table 42.4, accept vector arguments and return the matrix that is constructed by the corresponding parameters. According to the number of arguments, the following list shows what matrix a function returns when the arguments are vectors:

- A function, **FUNC0**, that has zero arguments, always returns the corresponding scalar parameter. **DCCA** and **DCCB** are types of **FUNC0**.
- A function, **FUNC1**, that has one vector argument I, where $I = (i_1\ i_2\ \ldots\ i_{n_I})'$, returns a vector $R = (r_1\ r_2\ \ldots\ r_{n_I})'$, where $r_k =$ **FUNC1**(i_k), $k = 1, \ldots, n_I$. **CONST**, **ECCONST**, **ECLTREND**, **LAMBDA**, **LTREND**, and **QTREND** are types of **FUNC1**.
- A function, **FUNC2**, that has two vector arguments I and J, where $I = (i_1\ i_2\ \ldots\ i_{n_I})'$ and $J = (j_1\ j_2\ \ldots\ j_{n_J})'$, returns a matrix

$$R = \begin{pmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,n_J} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,n_J} \\ \cdots & & & \\ r_{n_I,1} & r_{n_I,2} & \cdots & r_{n_I,n_J} \end{pmatrix}$$

  where $r_{k,m} =$ **FUNC2**(i_k, j_m), $k = 1, \ldots, n_I, m = 1, \ldots, n_J$. **ALPHA**, **BETA**, **CCC**, **COV**, **DCCS**, **GCHC**, **SD**, and **SDUMMY** and types of **FUNC2**.

- A function, **FUNC3**, that has three vector arguments L, I, and J, where $L = (l_1\ l_2\ \ldots\ l_{n_L})'$, $I = (i_1\ i_2\ \ldots\ i_{n_I})'$, and $J = (j_1\ j_2\ \ldots\ j_{n_J})'$, returns a matrix

$$R = \begin{pmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,n_L n_J} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,n_L n_J} \\ \cdots & & & \\ r_{n_I,1} & r_{n_I,2} & \cdots & r_{n_I,n_L n_J} \end{pmatrix}$$

  where $r_{k,m} =$ **FUNC3**(l_m, i_k, j_m), $k = 1, \ldots, n_I, m = 1, \ldots, n_L n_J$, and l_m and j_m are the quotient and remainder of m divided by n_J, respectively. **ACH**, **AR**, **EACH**, **GCH**, **MA**, **PACH**, **QACH**, **TACH**, and **XL** are types of **FUNC3**.

The functions that refer to parameters can accept empty arguments or omit any number of last arguments. The empty or omitted arguments are replaced with all possible values for those arguments. For example, PROC VARMAX is used to fit a bivariate (k=2) VARX(1,1) model with three exogenous variables as follows:

```
model y1 y2 = x1 x2 x3 / p=1 xlag=3;
```

In order to restrict the third exogenous variable from having an effect on the first dependent variable, and to restrict the first exogenous variable from having an effect on the second dependent variable, you can use the following statement:

```
restrict XL({0 1 2 3}, 1, 3) = 0,
         XL({0 1 2 3}, 2, 1) = 0;
```

Taking advantage of empty arguments, you can specify the preceding example as follows:

```
restrict XL( , 1, 3) = 0,
         XL( , 2, 1) = 0;
```

To get all coefficients of the first lag exogenous variables on dependent variables, you can use **XL(1, {1 2}, {1 2 3})** or **XL(1, ,)** or **XL(1)**. To get all coefficients of exogenous variables on dependent variables, you can use **XL({0 1 2 3}, {1 2}, {1 2 3})**, or **XL(, ,)** or **XL()** or even just **XL**.

Another type of function does not refer to parameters but generates useful matrices. Table 42.5 lists all built-in functions supported by the RESTRICT statement.

Table 42.5 Functions Not Referring to Parameters

Function	Description
DIAG(A)	Creates a diagonal matrix from a vector or extracts the diagonal elements of a matrix
I(n)	Creates an $n \times n$ identity matrix
J(m,n,elem)	Creates an $m \times n$ matrix with all elements equal to **elem**
SHAPE(A,m,n)	Creates a $m \times n$ matrix with elements of matrix **A**

For more information about each function in Table 42.5, see the section "Details of Functions" on page 3036.

Examples

The following examples show how to use the RESTRICT statement.

This example shows a bivariate (k=2) VAR(2) model:

```
proc varmax data=one;
   model y1 y2 / p=2;
   restrict AR(1,1,2)=0, AR(2,1,2)=0.3;
run;
```

The `AR(1,1,2)` and `AR(2,1,2)` parameters are fixed as `AR(1,1,2)=0` and `AR(2,1,2)=0.3`, respectively, and other parameters are to be estimated.

The following example shows a bivariate (k=2) VAR(1) model, estimated using the ML method:

```
proc varmax data=two;
   model y1 y2 = / p=1 method=ml;
   restrict cov(1,1)=cov(2,2), cov(1,2)=0;
run;
```

The `COV(1,1)` and `COV(2,2)` parameters are equal, and the correlation between the two series is fixed at 0. You can also express the preceding restrictions in matrix expressions as follows. This approach is very convenient when the number of dependent variables is large:

```
proc varmax data=two;
   model y1 y2 = / p=1 method=ml;
   restrict cov = cov(1,1)*I(2);
run;
```

When restricting a linear combination of parameters to be 0, you can omit the equal sign. For example, the following two RESTRICT statements are equivalent:

```
restrict AR(1)[1,1]-AR(1)[2,2], 2*MA(1)[1,2]-MA(1)[2,1];

restrict AR(1)[1,1]-AR(1)[2,2] = 0, 2*MA(1)[1,2]-MA(1)[2,1] = 0;
```

The following RESTRICT statement constrains four parameter estimates to be equal:

```
restrict AR(1)[1,1] = AR(1)[1,2],
         AR(1)[1,2] = AR(1)[2,1],
         AR(1)[2,1] = AR(1)[2,2];
```

This restriction can be abbreviated as follows:

```
restrict AR(1)[1,1] = AR(1)[1,2] = AR(1)[2,1] = AR(1)[2,2];
```

Or, in matrix expressions,

```
restrict AR(1,1:2,1:2) = J(2,2,AR(1,1,1));
```

The VARMA representation $A(L)y_t = \Theta(L)\varepsilon_t$, where $A(L) = I_k - A_1L - \cdots - A_pL^p$ and $\Theta(L) = I_k - \Theta_1L - \cdots - \Theta_qL^q$, is said to be in final equation form if $A(L) = a(L)I_k$, where $a(L) = 1 - a_1L - \cdots - a_pL^p$ is a scalar operator with $a_p \neq 0$. If p and k are large, it would be difficult and inconvenient to restrict AR parameters element by element in standard form to estimate the VARMA model in final equation form. However, when you use matrix expressions, the restrictions become very simple, as shown in the following statement for a trivariate ($k = 3$) VARMA(p, q) model, where p might be any positive integer:

```
restrict AR = AR(,1,1) @ I(3);
```

For the vector error correction models, the **AR(1,.,.)** parameters (that is, Π) are not supported in the RESTRICT statement, because **AR(1)** is in fact the product of the estimated parameters $\boldsymbol{\alpha}$ and the transpose of $\boldsymbol{\beta}$. Any linear constraints on **AR(1)** should be regarded as nonlinear constraints on the estimated parameters. For the same reason, the **CONST(.)** or **LTREND(.)** functions are not supported in the RESTRICT statement if the ECTREND option in the COINTEG statement is specified. For example, the following statements are supported:

```
model y1-y4 / p=2;
cointeg rank=1 ectrend;
restrict ALPHA + BETA = 1.0,
         ECCONST;
```

However, neither of the following sets of statements is supported:

```
model y1-y4 / p=2;
cointeg rank=1 ectrend;
restrict AR(1,1,1) = 0;
```

```
model y1-y4 / p=2;
cointeg rank=1 ectrend;
restrict CONST(2) = 0.2;
```

Details of Operators

This section describes all operators that are available in the RESTRICT statement. Each subsection shows how the operator is used, followed by a description of the operator.

Addition Operator: +

```
matrix1 + matrix2

matrix + scalar

matrix + vector
```

The addition operator (+) computes a new matrix whose elements are the sums of the corresponding elements of **matrix1** and **matrix2**. If **matrix1** and **matrix2** are both $n \times p$ matrices, then the addition operator adds the element in the ith row and jth column of the first matrix to the element in the ith row and jth column of the second matrix, for $i = 1, \ldots, n$, $j = 1, \ldots, p$. For example, **{1 2 3, 4 5 6} + {7 8 9, 10 11 12}** results in **{8 10 12, 14 16 18}**.

You can also use the addition operator as follows to conveniently add a value to each element of a matrix, to each column of a matrix, or to each row of a matrix:

- When you use the **matrix + scalar** form, the scalar value is added to each element of the matrix.
- When you use the **matrix + vector** form, the vector is added to each row or column of the $n \times p$ matrix.
  - If you add an $n \times 1$ column vector, each row of the vector is added to each row of the matrix.
  - If you add a $1 \times p$ row vector, each column of the vector is added to each column of the matrix.

For example, you can obtain `{2 3 4, 5 6 7}` from `{1 2 3, 4 5 6} + 1` or `{1 2 3, 4 5 6} + {1 1 1}` or `{1 2 3, 4 5 6} + {1, 1}`.

Comparison Operators: =, <, <=, >, >=

```
matrix1 = matrix2

matrix1 < matrix2

matrix1 <= matrix2

matrix1 > matrix2

matrix1 >= matrix2
```

The comparison operators (=, <, <=, >, >=) compare two matrices element by element and return a list of equivalent restrictions on only scalar constants and parameters.

For example, the RESTRICT statement with matrix expressions

```
restrict AR(1,{1,2},{1,2}) = MA(2,{3,4},{3,4});
```

is transformed into the following equivalent RESTRICT statement with scalar parameters:

```
restrict AR(1,1,1) = MA(2,3,3),
         AR(1,1,2) = MA(2,3,4),
         AR(1,2,1) = MA(2,4,3),
         AR(1,2,2) = MA(2,4,4);
```

You can also use the comparison operators to conveniently compare all elements of a matrix with a scalar:

- If either argument is a scalar, then the VARMAX procedure performs an elementwise comparison between each element of the matrix and the scalar.

You can also compare an $n \times p$ matrix with a row or column vector:

- If the comparison is with an $n \times 1$ column vector, the VARMAX procedure compares each row of the vector to each row of the matrix.
- If the comparison is with a $1 \times p$ row vector, the VARMAX procedure compares each column of the vector to each column of the matrix.

For example, the following statements are equivalent:

```
restrict AR(1,1:2,1:3) >= 0.2;

restrict AR(1,1:2,1:3) >= {0.2, 0.2};

restrict AR(1,1:2,1:3) >= {0.2 0.2 0.2};
```

Concatenation Operator, Horizontal: ||

```
matrix1 || matrix2
```

The horizontal concatenation operator (||) produces a new matrix by horizontally joining **matrix1** and **matrix2**. The matrices must have the same number of rows, which is also the number of rows in the new

matrix. The number of columns in the new matrix is the number of columns in **matrix1** plus the number of columns in **matrix2**.

For example, `{1 1 1, 7 7 7} || {0 0 0, 8 8 8}` returns `{1 1 1 0 0 0, 7 7 7 8 8 8}`.

Concatenation Operator, Vertical: //

matrix1 // matrix2

The vertical concatenation operator (//) produces a new matrix by vertically joining **matrix1** and **matrix2**. The matrices must have the same number of columns, which is also the number of columns in the new matrix. The number of rows in the new matrix is the number of rows in **matrix1** plus the number of rows in **matrix2**.

For example, `{1 1 1} // {0 0 0, 8 8 8}` returns `{1 1 1, 0 0 0, 8 8 8}`.

Direct Product Operator: @

matrix1 @ matrix2

The direct product operator (@) computes a new matrix that is the direct product (also called the *Kronecker product*) of **matrix1** and **matrix2**. For matrices $\mathbf{A}$ and $\mathbf{B}$, the direct product is denoted by $\mathbf{A} \otimes \mathbf{B}$. The number of rows in the new matrix equals the product of the number of rows in **matrix1** and the number of rows in **matrix2**; the number of columns in the new matrix equals the product of the number of columns in **matrix1** and the number of columns in **matrix2**.

Specifically, if $\mathbf{A}$ is an $n \times p$ matrix and $\mathbf{B}$ is a $m \times q$ matrix, then the Kronecker product $\mathbf{A} \otimes \mathbf{B}$ is the following $nm \times pq$ block matrix:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} A_{11}B & \cdots & A_{1p}B \\ \vdots & \ddots & \vdots \\ A_{n1}B & \cdots & A_{np}B \end{bmatrix}$$

For example, `{1 2, 3 4} @ {0 2}` returns `{0 2 0 4, 0 6 0 8}`, and `{0 2} @ {1 2, 3 4}` returns `{0 0 2 4, 0 0 6 8}`. Note that the direct product of two matrices is not commutative.

Index Creation Operator: :

value1 : value2

The index creation operator (:) creates a column vector whose first element is **value1**, whose second element is **value1**+1, and so on, until the last element, which is less than or equal to **value2**.

For example, `3 : 6` returns `{3 4 5 6}`.

If **value1** is greater than **value2**, a reverse-order index is created. For example, `6 : 3` returns `{6 5 4 3}`.

Neither **value1** nor **value2** is required to be an integer.

Multiplication Operator, Elementwise:

matrix1 # matrix2

matrix # scalar

matrix # vector

The elementwise multiplication operator (#) computes a new matrix whose elements are the products of the corresponding elements of **matrix1** and **matrix2**.

For example, **{1 2, 3 4} # {4 8, 0 5}** returns **{4 16, 0 20}**.

In addition to multiplying matrices that have the same dimensions, you can use the elementwise multiplication operator to multiply a matrix and a scalar:

- When either argument is a scalar, each element in **matrix** is multiplied by the scalar value.

When you use the **matrix** # **vector** form, each row or column of the $n \times p$ matrix is multiplied by a corresponding element of the vector:

- If you multiply by an $n \times 1$ column vector, each row of the matrix is multiplied by the corresponding row of the vector.
- If you multiply by a $1 \times p$ row vector, each column of the matrix is multiplied by the corresponding column of the vector.

For example, a 2×3 matrix can be multiplied on either side by a 2×3, 1×3, 2×1, or 1×1 scalar.

The product of elementwise multiplication is also known as the Schur or Hadamard product. Elementwise multiplication (which uses the # operator) should not be confused with matrix multiplication (which uses the * operator).

*Multiplication Operator, Matrix: **

```
matrix1 * matrix2
```

The matrix multiplication operator (*) computes a new matrix by performing matrix multiplication. The first matrix must have the same number of columns as the second matrix has rows. The new matrix has the same number of rows as the first matrix and the same number of columns as the second matrix. That is, if $\boldsymbol{A}$ is an $n \times p$ matrix and $\boldsymbol{B}$ is a $p \times m$ matrix, then the product $A * B$ is an $n \times m$ matrix. The (i, j) element of the product is the sum $\sum_{k=1}^{p} A_{ik} B_{kj}$.

For example, **{1 2, 3 4} * {1, 2}** returns **{5, 11}**.

Sign Reversal Operator: –

```
- matrix
```

The sign reversal operator (−) computes a new matrix whose elements are formed by reversing the sign of each element in **matrix**. The sign reversal operator is also called the *unary minus* operator.

For example, **-{-1 7 6, 2 0 -8}** returns **{1 -7 -6, -2 0 8}**.

Subscripts: [ ]

```
matrix[rows, columns]

matrix[elements]
```

Subscripts are used with matrices to select submatrices, where **rows**, **columns**, and **elements** are expressions that evaluate to scalars or vectors. If these expressions are numeric, they must contain valid subscript values of rows and columns, or the indices, in the argument matrix.

For example, `{1 2 3, 4 5 6, 7 8 9}[2,3]` returns `6`, `{1 2 3, 4 5 6, 7 8 9}[2,1:3]` returns `{4 5 6}`, and `{1 2 3, 4 5 6, 7 8 9}[,3]` returns `{3, 6, 9}`. Because the VARMAX procedure stores matrices in row-major order, `{11 22 33, 44 55 66, 77 88 99}[{3 5 9}]` returns `{33, 55, 99}`.

Subtraction Operator: –

matrix1 – matrix2

matrix – scalar

matrix – vector

The subtraction operator (−) computes a new matrix whose elements are formed by subtracting the corresponding elements of **matrix2** from those of **matrix1**.

In addition to subtracting conformable matrices, you can also use the subtraction operator to subtract a scalar from a matrix or subtract a vector from a matrix:

- When either argument is a scalar, the VARMAX procedure performs the subtraction between the scalar and each element of the matrix argument. For example, when you use the **matrix – scalar** form, the scalar value is subtracted from each element of the matrix.
- When you use the **matrix – vector** form, the vector is subtracted from each row or column of the $n \times p$ matrix.
  - If you subtract an $n \times 1$ column vector, each row of the vector is subtracted from each row of the matrix.
  - If you subtract a $1 \times p$ row vector, each column of the vector is subtracted from each column of the matrix.

For example, `{1 2 3, 4 5 6} – {1 1 1, 1 1 1}` returns `{0 1 2, 3 4 5}`. The same results can be obtained by `{1 2 3, 4 5 6} – 1` or `{1 2 3, 4 5 6} – {1 1 1}` or `{1 2 3, 4 5 6} – {1, 1}`.

Transpose Operator: `

matrix`

The transpose operator, denoted by the backquote character (`), exchanges the rows and columns of **matrix**, producing the transpose of **matrix**. If v is the value in the ith row and jth column of **matrix**, then the transpose of **matrix** contains v in the jth row and ith column. If **matrix** contains n rows and p columns, the transpose has p rows and n columns.

For example, `{1 2, 3 4, 5 6}`` returns `{1 3 5, 2 4 6}`.

Details of Functions

DIAG Function

DIAG(matrix)

The **DIAG** function creates a diagonal matrix from a vector or extracts the diagonal elements of a matrix. The **matrix** argument can be either a square matrix or a vector.

If **matrix** is a vector, the **DIAG** function creates a matrix whose diagonal elements are the values in the vector. All off-diagonal elements are zeros.

If **matrix** is a square matrix, the **DIAG** function creates a vector from the diagonal elements of the matrix.

For example, **DIAG({1 2 3, 4 5 6, 7 8 9})** returns **{1, 5, 9}**. Also, **DIAG({1 5 9})** or **DIAG({1, 5, 9})** or **DIAG(DIAG({1 2 3, 4 5 6, 7 8 9}))** returns **{1 0 0, 0 5 0, 0 0 9}**.

I Function

I(dim)

The **I** function creates an identity matrix that contains **dim** rows and columns. The diagonal elements of an identity matrix are ones; all other elements are zeros. The value of **dim** must be an integer greater than or equal to 1. Noninteger operands are rounded to the nearest integer.

For example, **I(3)** returns **{1 0 0, 0 1 0, 0 0 1}**.

J Function

J(nrow, ncol, value)

The **J** function creates a matrix that contains **nrow** rows and **ncol** columns, in which all elements are equal to **value**.

The arguments **nrow** and **ncol** are both integers; **value** can be any expression that returns a linear combination of scalar constants and parameters.

For example, **J(2, 3, 1)** returns **{1 1 1, 1 1 1}**. **J(2, 3, 5+2*AR(1,1,1))** returns the same result as **J(2, 3, 1) * (5+2*AR(1,1,1))**.

SHAPE Function

SHAPE(matrix, nrow, ncol)

The **SHAPE** function creates a new matrix from data in **matrix**. The values **nrow** and **ncol** specify the number of rows and columns, respectively, in the new matrix. The **SHAPE** function produces the result matrix by traversing the argument matrix in row-major order until it reaches the specified number of elements. If necessary, the SHAPE function reuses elements.

For example, **SHAPE({1 2 3, 4 5 6}, 3, 2)** returns **{1 2, 3 4, 5 6}**; **SHAPE({1 2 3, 4 5 6}, 5, 2)** returns **{1 2, 3 4, 5 6, 1 2, 3 4}**; and **SHAPE({1 2 3, 4 5 6}, 1, 4)** returns **{1 2 3 4}**.

TEST Statement

TEST *restriction, . . . , restriction* **;**

The TEST statement performs the Wald test for the joint linear hypothesis that is specified in the statement. Each restriction specifies a linear hypothesis to be tested. If you specify more than one *restriction*, separate them with commas. Specify the *restrictions* in the same manner as in the RESTRICT statement. For information about how to define restriction by using matrix expressions, operators, and functions, see the section "RESTRICT Statement" on page 3025. You can specify any number of TEST statements.

To use the TEST statement, you need to know the form of the model. If you do not specify the GARCH statement, the COINTEG statement, or the ECM=, P=, Q=, or XLAG= option in the MODEL statement, then the TEST statement is not applicable. Nonlinear restrictions on parameters are not supported.

For information about the Wald test, see the section "Granger Causality Test" on page 3070.

The following is an example of the TEST statement for a bivariate (k=2) VAR(2) model:

```
proc varmax data=one;
   model y1 y2 / p=2;
   test AR(1,1,2) = 0, AR(2,1,2) = 0;
run;
```

After estimating the parameters, the TEST statement tests the null hypothesis that **AR(1,1,2)=0** and **AR(2,1,2)=0**. Like the RESTRICT statement, the preceding TEST statement can be abbreviated as follows:

```
test AR(1,1,2) = AR(2,1,2) = 0;
```

or

```
test AR(1,1,2), AR(2,1,2);
```

Note that the following statements are different from the preceding statement:

```
test AR(1,1,2);
test AR(2,1,2);
```

These two TEST statements are to test two null hypotheses separately: one is **AR(1,1,2)=0**, and the other is **AR(2,1,2)=0**.

For the vector error correction models, you can test the hypothesis on the **AR(1,.,.)** parameters (that is, Π) by using the TEST statement, because asymptotically these parameters follow a normal distribution and the Wald test can be applied. For the same reason, you can use the **CONST(.)** or **LTREND(.)** function in the TEST statement if the ECTREND option in the COINTEG statement is specified. However, the **BETA(.,.)**, **ECCONST(.)**, and **ECLTREND(.)** functions are not supported in the TEST statement. For example, the following statements are supported:

```
model y1-y4 / p=2;
cointeg rank=1 ectrend;
test AR(1,1,1);
test CONST(2);
```

However, the following statements are not supported:

```
model y1-y4 / p=2;
cointeg rank=1 ectrend;
test BETA(1,1) = BETA(2,1) = 0;
```

or

```
model y1-y4 / p=2;
cointeg rank=1 ectrend;
test ECCONST(1) = 0.2;
```

Details: VARMAX Procedure

Missing Values

The VARMAX procedure currently does not support missing values. PROC VARMAX uses the first contiguous group of observations that have no missing values for any of the MODEL statement variables. Observations at the beginning of the data set that have missing values for any MODEL statement variables are not used or included in the output data set. At the end of the data set, observations can have dependent (endogenous) variables with missing values and independent (exogenous) variables with nonmissing values.

VARMAX Model

The vector autoregressive moving-average model with exogenous variables is called the VARMAX(p,q,s) model. The form of the model can be written as

$$\mathbf{y}_t = \sum_{i=1}^{p} \Phi_i \mathbf{y}_{t-i} + \sum_{i=0}^{s} \Theta_i^* \mathbf{x}_{t-i} + \boldsymbol{\epsilon}_t - \sum_{i=1}^{q} \Theta_i \boldsymbol{\epsilon}_{t-i}$$

where the output variables of interest, $\mathbf{y}_t = (y_{1t}, \ldots, y_{kt})'$, can be influenced by other input variables, $\mathbf{x}_t = (x_{1t}, \ldots, x_{rt})'$, which are determined outside of the system of interest. The variables $\mathbf{y}_t$ are referred to as dependent, response, or endogenous variables, and the variables $\mathbf{x}_t$ are referred to as independent, input, predictor, regressor, or exogenous variables. The unobserved noise variables, $\boldsymbol{\epsilon}_t = (\epsilon_{1t}, \ldots, \epsilon_{kt})'$, are a vector white noise process.

The VARMAX(p,q,s) model can be written

$$\Phi(B)\mathbf{y}_t = \Theta^*(B)\mathbf{x}_t + \Theta(B)\boldsymbol{\epsilon}_t$$

where

$$\begin{aligned} \Phi(B) &= I_k - \Phi_1 B - \cdots - \Phi_p B^p \\ \Theta^*(B) &= \Theta_0^* + \Theta_1^* B + \cdots + \Theta_s^* B^s \\ \Theta(B) &= I_k - \Theta_1 B - \cdots - \Theta_q B^q \end{aligned}$$

are matrix polynomials in B in the backshift operator, such that $B^i \mathbf{y}_t = \mathbf{y}_{t-i}$, the Φ_i and Θ_i are $k \times k$ matrices, and the Θ_i^* are $k \times r$ matrices.

The following assumptions are made:

- $\mathrm{E}(\boldsymbol{\epsilon}_t) = 0$, $\mathrm{E}(\boldsymbol{\epsilon}_t \boldsymbol{\epsilon}_t') = \Sigma$, which is positive-definite, and $\mathrm{E}(\boldsymbol{\epsilon}_t \boldsymbol{\epsilon}_s') = 0$ for $t \neq s$.
- For stationarity and invertibility of the VARMAX process, the roots of $|\Phi(z)| = 0$ and $|\Theta(z)| = 0$ are outside the unit circle.

- The exogenous (independent) variables $\mathbf{x}_t$ are not correlated with residuals $\boldsymbol{\epsilon}_t$, $\mathrm{E}(\mathbf{x}_t\boldsymbol{\epsilon}_t') = 0$. The exogenous variables can be stochastic or nonstochastic. When the exogenous variables are stochastic and their future values are unknown, forecasts of these future values are needed to forecast the future values of the endogenous (dependent) variables. On occasion, future values of the exogenous variables can be assumed to be known because they are deterministic variables. The VARMAX procedure assumes that the exogenous variables are nonstochastic if future values are available in the input data set. Otherwise, the exogenous variables are assumed to be stochastic and their future values are forecasted by assuming that they follow the VARMA(p,q) model, prior to forecasting the endogenous variables, where p and q are the same as in the VARMAX(p,q,s) model.

State Space Representation

Another representation of the VARMAX(p,q,s) model is in the form of a state variable or a state space model, which consists of a state equation

$$\mathbf{z}_t = F\mathbf{z}_{t-1} + K\mathbf{x}_t + G\boldsymbol{\epsilon}_t$$

and an observation equation

$$\mathbf{y}_t = H\mathbf{z}_t$$

where

$$\mathbf{z}_t = \begin{bmatrix} \mathbf{y}_t \\ \vdots \\ \mathbf{y}_{t-p+1} \\ \mathbf{x}_t \\ \vdots \\ \mathbf{x}_{t-s+1} \\ \boldsymbol{\epsilon}_t \\ \vdots \\ \boldsymbol{\epsilon}_{t-q+1} \end{bmatrix}, \quad K = \begin{bmatrix} \Theta_0^* \\ 0_{k\times r} \\ \vdots \\ 0_{k\times r} \\ I_r \\ 0_{r\times r} \\ \vdots \\ 0_{r\times r} \\ 0_{k\times r} \\ \vdots \\ 0_{k\times r} \end{bmatrix}, \quad G = \begin{bmatrix} I_k \\ 0_{k\times k} \\ \vdots \\ 0_{k\times k} \\ 0_{r\times k} \\ \vdots \\ 0_{r\times k} \\ I_{k\times k} \\ 0_{k\times k} \\ \vdots \\ 0_{k\times k} \end{bmatrix}$$

$$F = \begin{bmatrix} \Phi_1 & \cdots & \Phi_{p-1} & \Phi_p & \Theta_1^* & \cdots & \Theta_{s-1}^* & \Theta_s^* & -\Theta_1 & \cdots & -\Theta_{q-1} & -\Theta_q \\ I_k & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & I_k & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 & I_r & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & I_r & 0 & 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & I_k & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & I_k & 0 \end{bmatrix}$$

and

$$H = [I_k, 0_{k\times k}, \ldots, 0_{k\times k}, 0_{k\times r}, \ldots, 0_{k\times r}, 0_{k\times k}, \ldots, 0_{k\times k}]$$

On the other hand, it is assumed that $\mathbf{x}_t$ follows a VARMA(p,q) model

$$\mathbf{x}_t = \sum_{i=1}^{p} A_i \mathbf{x}_{t-i} + \mathbf{a}_t - \sum_{i=1}^{q} C_i \mathbf{a}_{t-i}$$

The model can also be expressed as

$$A(B)\mathbf{x}_t = C(B)\mathbf{a}_t$$

where $A(B) = I_r - A_1 B - \cdots - A_p B^p$ and $C(B) = I_r - C_1 B - \cdots - C_q B^q$ are matrix polynomials in B, and the A_i and C_i are $r \times r$ matrices. Without loss of generality, the AR and MA orders can be taken to be the same as the VARMAX(p,q,s) model, and $\mathbf{a}_t$ and $\boldsymbol{\epsilon}_t$ are independent white noise processes.

Under suitable conditions such as stationarity, $\mathbf{x}_t$ is represented by an infinite order moving-average process

$$\mathbf{x}_t = A(B)^{-1}C(B)\mathbf{a}_t = \Psi^x(B)\mathbf{a}_t = \sum_{j=0}^{\infty} \Psi_j^x \mathbf{a}_{t-j}$$

where $\Psi^x(B) = A(B)^{-1}C(B) = \sum_{j=0}^{\infty} \Psi_j^x B^j$.

The optimal minimum mean squared error (minimum MSE) i-step-ahead forecast of $\mathbf{x}_{t+i}$ is

$$\begin{aligned} \mathbf{x}_{t+i|t} &= \sum_{j=i}^{\infty} \Psi_j^x \mathbf{a}_{t+i-j} \\ \mathbf{x}_{t+i|t+1} &= \mathbf{x}_{t+i|t} + \Psi_{i-1}^x \mathbf{a}_{t+1} \end{aligned}$$

For $i > q$,

$$\mathbf{x}_{t+i|t} = \sum_{j=1}^{p} A_j \mathbf{x}_{t+i-j|t}$$

The VARMAX(p,q,s) model has an absolutely convergent representation as

$$\begin{aligned} \mathbf{y}_t &= \Phi(B)^{-1}\Theta^*(B)\mathbf{x}_t + \Phi(B)^{-1}\Theta(B)\boldsymbol{\epsilon}_t \\ &= \Psi^*(B)\Psi^x(B)\mathbf{a}_t + \Phi(B)^{-1}\Theta(B)\boldsymbol{\epsilon}_t \\ &= V(B)\mathbf{a}_t + \Psi(B)\boldsymbol{\epsilon}_t \end{aligned}$$

or

$$\mathbf{y}_t = \sum_{j=0}^{\infty} V_j \mathbf{a}_{t-j} + \sum_{j=0}^{\infty} \Psi_j \boldsymbol{\epsilon}_{t-j}$$

where $\Psi(B) = \Phi(B)^{-1}\Theta(B) = \sum_{j=0}^{\infty} \Psi_j B^j$, $\Psi^*(B) = \Phi(B)^{-1}\Theta^*(B)$, and $V(B) = \Psi^*(B)\Psi^x(B) = \sum_{j=0}^{\infty} V_j B^j$.

The optimal (minimum MSE) i-step-ahead forecast of $\mathbf{y}_{t+i}$ is

$$\begin{aligned} \mathbf{y}_{t+i|t} &= \sum_{j=i}^{\infty} V_j \mathbf{a}_{t+i-j} + \sum_{j=i}^{\infty} \Psi_j \boldsymbol{\epsilon}_{t+i-j} \\ \mathbf{y}_{t+i|t+1} &= \mathbf{y}_{t+i|t} + V_{i-1}\mathbf{a}_{t+1} + \Psi_{i-1}\boldsymbol{\epsilon}_{t+1} \end{aligned}$$

for $i = 1, \ldots, v$ with $v = \max(p, q+1)$. For $i > q$,

$$\begin{aligned}
\mathbf{y}_{t+i|t} &= \sum_{j=1}^{p} \Phi_j \mathbf{y}_{t+i-j|t} + \sum_{j=0}^{s} \Theta_j^* \mathbf{x}_{t+i-j|t} \\
&= \sum_{j=1}^{p} \Phi_j \mathbf{y}_{t+i-j|t} + \Theta_0^* \mathbf{x}_{t+i|t} + \sum_{j=1}^{s} \Theta_j^* \mathbf{x}_{t+i-j|t} \\
&= \sum_{j=1}^{p} \Phi_j \mathbf{y}_{t+i-j|t} + \Theta_0^* \sum_{j=1}^{p} A_j \mathbf{x}_{t+i-j|t} + \sum_{j=1}^{s} \Theta_j^* \mathbf{x}_{t+i-j|t} \\
&= \sum_{j=1}^{p} \Phi_j \mathbf{y}_{t+i-j|t} + \sum_{j=1}^{u} (\Theta_0^* A_j + \Theta_j^*) \mathbf{x}_{t+i-j|t}
\end{aligned}$$

where $u = \max(p, s)$.

Define $\Pi_j = \Theta_0^* A_j + \Theta_j^*$. For $i = v > q$ with $v = \max(p, q+1)$, you obtain

$$\begin{aligned}
\mathbf{y}_{t+v|t} &= \sum_{j=1}^{p} \Phi_j \mathbf{y}_{t+v-j|t} + \sum_{j=1}^{u} \Pi_j \mathbf{x}_{t+v-j|t} \quad \text{for } u \le v \\
\mathbf{y}_{t+v|t} &= \sum_{j=1}^{p} \Phi_j \mathbf{y}_{t+v-j|t} + \sum_{j=1}^{r} \Pi_j \mathbf{x}_{t+v-j|t} \quad \text{for } u > v
\end{aligned}$$

From the preceding relations, a state equation is

$$\mathbf{z}_{t+1} = F\mathbf{z}_t + K\mathbf{x}_t^* + G\mathbf{e}_{t+1}$$

and an observation equation is

$$\mathbf{y}_t = H\mathbf{z}_t$$

where

$$\mathbf{z}_t = \begin{bmatrix} \mathbf{y}_t \\ \mathbf{y}_{t+1|t} \\ \vdots \\ \mathbf{y}_{t+v-1|t} \\ \mathbf{x}_t \\ \mathbf{x}_{t+1|t} \\ \vdots \\ \mathbf{x}_{t+v-1|t} \end{bmatrix}, \quad \mathbf{x}_t^* = \begin{bmatrix} \mathbf{x}_{t+v-u} \\ \mathbf{x}_{t+v-u+1} \\ \vdots \\ \mathbf{x}_{t-1} \end{bmatrix}, \quad \mathbf{e}_{t+1} = \begin{bmatrix} \mathbf{a}_{t+1} \\ \boldsymbol{\epsilon}_{t+1} \end{bmatrix}$$

$$F = \begin{bmatrix} 0 & I_k & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & I_k & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \Phi_v & \Phi_{v-1} & \Phi_{v-2} & \cdots & \Phi_1 & \Pi_v & \Pi_{v-1} & \Pi_{v-2} & \cdots & \Pi_1 \\ 0 & 0 & 0 & \cdots & 0 & 0 & I_r & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & I_r & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & A_v & A_{v-1} & A_{v-2} & \cdots & A_1 \end{bmatrix}$$

$$K = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \Pi_u & \Pi_{u-1} & \cdots & \Pi_{v+1} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \quad G = \begin{bmatrix} V_0 & I_k \\ V_1 & \Psi_1 \\ \vdots & \vdots \\ V_{v-1} & \Psi_{v-1} \\ I_r & 0_{r\times k} \\ \Psi_1^x & 0_{r\times k} \\ \vdots & \vdots \\ \Psi_{v-1}^x & 0_{r\times k} \end{bmatrix}$$

and

$$H = [I_k, 0_{k\times k}, \ldots, 0_{k\times k}, 0_{k\times r}, \ldots, 0_{k\times r}]$$

Note that the matrix K and the input vector $\mathbf{x}_t^*$ are defined only when $u > v$.

Dynamic Simultaneous Equations Modeling

In the econometrics literature, the VARMAX(p,q,s) model is sometimes written in a form that is slightly different than the one shown in the previous section. This alternative form is referred to as a *dynamic simultaneous equations* model or a *dynamic structural equations* model.

Because $\mathrm{E}(\boldsymbol{\epsilon}_t \boldsymbol{\epsilon}_t') = \Sigma$ is assumed to be positive-definite, there exists a lower triangular matrix A_0 that has ones on the diagonals such that $A_0 \Sigma A_0' = \Sigma^d$, where Σ^d is a diagonal matrix that has positive diagonal elements.

$$A_0 \mathbf{y}_t = \sum_{i=1}^{p} A_i \mathbf{y}_{t-i} + \sum_{i=0}^{s} C_i^* \mathbf{x}_{t-i} + A_0 \boldsymbol{\epsilon}_t - \sum_{i=1}^{q} C_i A_0 \boldsymbol{\epsilon}_{t-i}$$

where $A_i = A_0 \Phi_i$, $C_i^* = A_0 \Theta_i^*$, and $C_i = A_0 \Theta_i A_0^{-1}$.

As an alternative form,

$$A_0\mathbf{y}_t = \sum_{i=1}^{p} A_i\mathbf{y}_{t-i} + \sum_{i=0}^{s} C_i^*\mathbf{x}_{t-i} + \mathbf{a}_t - \sum_{i=1}^{q} C_i\mathbf{a}_{t-i}$$

where $A_i = A_0\Phi_i$, $C_i^* = A_0\Theta_i^*$, $C_i = A_0\Theta_i A_0^{-1}$, and $\mathbf{a}_t = A_0\boldsymbol{\epsilon}_t$. The covariance matrix of $\mathbf{a}_t$ is the diagonal matrix Σ^d. The PRINT=(DYNAMIC) option returns the parameter estimates that result from estimating the model in this form.

A dynamic simultaneous equations model involves a leading (lower triangular) coefficient matrix for $\mathbf{y}_t$ at lag 0 or a leading coefficient matrix for $\boldsymbol{\epsilon}_t$ at lag 0. Such a representation of the VARMAX(p,q,s) model can be more useful in certain circumstances than the standard representation. From the linear combination of the dependent variables obtained by $A_0\mathbf{y}_t$, you can easily see the relationship between the dependent variables in the current time.

The following statements provide the dynamic simultaneous equations of the VAR(1) model:

```
proc iml;
   sig = {1.0  0.5, 0.5 1.25};
   phi = {1.2 -0.5, 0.6 0.3};
   /* simulate the vector time series */
   call varmasim(y,phi) sigma = sig n = 100 seed = 34657;
   cn = {'y1' 'y2'};
   create simul1 from y[colname=cn];
   append from y;
quit;

data simul1;
   set simul1;
   date = intnx( 'year', '01jan1900'd, _n_-1 );
   format date year4.;
run;

proc varmax data=simul1;
   model y1 y2 / p=1 noint print=(dynamic);
run;
```

This is the same data set and model used in the section "Getting Started: VARMAX Procedure" on page 2954. You can compare the results of the VARMA model form and the dynamic simultaneous equations model form.

Figure 42.44 Dynamic Simultaneous Equations (DYNAMIC Option)

The VARMAX Procedure

Covariances of Innovations		
Variable	**y1**	**y2**
y1	1.28875	0.00000
y2	0.00000	1.29578

AR			
Lag	**Variable**	**y1**	**y2**
0	**y1**	1.00000	0.00000
	y2	-0.30845	1.00000
1	**y1**	1.15977	-0.51058
	y2	0.18861	0.54247

Dynamic Model Parameter Estimates						
Equation	**Parameter**	**Estimate**	**Standard Error**	**t Value**	**Pr > |t|**	**Variable**
y1	**AR1_1_1**	1.15977	0.05508	21.06	0.0001	y1(t-1)
	AR1_1_2	-0.51058	0.07140	-7.15	0.0001	y2(t-1)
y2	**AR0_2_1**	0.30845				y1(t)
	AR1_2_1	0.18861	0.05779	3.26	0.0015	y1(t-1)
	AR1_2_2	0.54247	0.07491	7.24	0.0001	y2(t-1)

In Figure 42.4 in the section "Getting Started: VARMAX Procedure" on page 2954, the covariance of $\boldsymbol{\epsilon}_t$ estimated from the VARMAX model form is

$$\Sigma_{\boldsymbol{\epsilon}} = \begin{pmatrix} 1.28875 & 0.39751 \\ 0.39751 & 1.41839 \end{pmatrix}$$

Figure 42.44 shows the results from estimating the model as a dynamic simultaneous equations model. By the decomposition of $\Sigma_{\boldsymbol{\epsilon}}$, you get a diagonal matrix ($\Sigma_{\mathbf{a}}$) and a lower triangular matrix (A_0) such as $\Sigma_{\mathbf{a}} = A_0 \Sigma_{\boldsymbol{\epsilon}} A_0'$ where

$$\Sigma_{\mathbf{a}} = \begin{pmatrix} 1.28875 & 0 \\ 0 & 1.29578 \end{pmatrix} \text{ and } A_0 = \begin{pmatrix} 1 & 0 \\ -0.30845 & 1 \end{pmatrix}$$

The lower triangular matrix (A_0) is shown in the left side of the simultaneous equations model. The parameter estimates in equations system are shown in the right side of the two-equations system.

The simultaneous equations model is written as

$$\begin{pmatrix} 1 & 0 \\ -0.30845 & 1 \end{pmatrix} \mathbf{y}_t = \begin{pmatrix} 1.15977 & -0.51058 \\ 0.18861 & 0.54247 \end{pmatrix} \mathbf{y}_{t-1} + \mathbf{a}_t$$

The resulting two-equation system can be written as

$$\begin{aligned} y_{1t} &= 1.15977 y_{1,t-1} - 0.51058 y_{2,t-1} + a_{1t} \\ y_{2t} &= 0.30845 y_{1t} + 0.18861 y_{1,t-1} + 0.54247 y_{2,t-1} + a_{2t} \end{aligned}$$

Impulse Response Function

Simple Impulse Response Function (IMPULSE=SIMPLE Option)

The VARMAX(p,q,s) model has a convergent representation

$$\mathbf{y}_t = \Psi^*(B)\mathbf{x}_t + \Psi(B)\boldsymbol{\epsilon}_t$$

where $\Psi^*(B) = \Phi(B)^{-1}\Theta^*(B) = \sum_{j=0}^{\infty} \Psi_j^* B^j$ and $\Psi(B) = \Phi(B)^{-1}\Theta(B) = \sum_{j=0}^{\infty} \Psi_j B^j$.

The elements of the matrices Ψ_j from the operator $\Psi(B)$, called the impulse response, can be interpreted as the response of a variable to a shock in another variable. Let $\psi_{j,in}$ be the (i, n) element of Ψ_j at lag j, where n is the index for the impulse variable, and i is the index for the response variable (impulse $\rightarrow$ response); that is to say, $\psi_{j,in}$ shows the reaction of the i-th variable to a unit shock in variable n, j periods ago, assuming that the effect is not contaminated by other shocks (Lütkepohl 1993). For instance, $\psi_{j,11}$ is an impulse response to $y_{1t} \rightarrow y_{1t}$, and $\psi_{j,12}$ is an impulse response to $y_{2t} \rightarrow y_{1t}$.

Accumulated Impulse Response Function (IMPULSE=ACCUM Option)

The accumulated impulse response function is the cumulative sum of the impulse response function, $\Psi_l^a = \sum_{j=0}^{l} \Psi_j$.

Orthogonalized Impulse Response Function (IMPULSE=ORTH Option)

The MA representation of a VARMA(p,q) model with a standardized white noise innovation process offers another way to interpret a VARMA(p,q) model. Since Σ is positive-definite, there is a lower triangular matrix P such that $\Sigma = PP'$. The alternate MA representation of a VARMA(p,q) model is written as

$$\mathbf{y}_t = \Psi^o(B)\mathbf{u}_t$$

where $\Psi^o(B) = \sum_{j=0}^{\infty} \Psi_j^o B^j$, $\Psi_j^o = \Psi_j P$, and $\mathbf{u}_t = P^{-1}\boldsymbol{\epsilon}_t$.

The elements of the matrices Ψ_j^o, called the *orthogonal impulse response*, can be interpreted as the effects of the components of the standardized shock process $\mathbf{u}_t$ on the process $\mathbf{y}_t$ at lag j.

Impulse Response of Transfer Function (IMPULSX=SIMPLE Option)

The coefficient matrix Ψ_j^* from the transfer function operator $\Psi^*(B)$ can be interpreted as the effects that changes in the exogenous variables $\mathbf{x}_t$ have on the output variable $\mathbf{y}_t$ at lag j; it is called an impulse response matrix in the transfer function.

Accumulated Impulse Response of Transfer Function (IMPULSX=ACCUM Option)

The accumulated impulse response in the transfer function is the cumulative sum of the impulse response in the transfer function, $\Psi_l^{*a} = \sum_{j=0}^{l} \Psi_j^*$.

The asymptotic distributions of the impulse functions can be seen in the section "VAR and VARX Modeling" on page 3067.

The following statements provide the impulse response and the accumulated impulse response in the transfer function for a VARX(1,0) model:

```
proc varmax data=grunfeld plot=impulse;
   model y1-y3 = x1 x2 / p=1 lagmax=5
                         printform=univariate
                         print=(impulsx=(all) estimates);
run;
```

In Figure 42.45, the variables $x1$ and $x2$ are impulses, and the variables $y1$, $y2$, and $y3$ are responses. The keyword STD stands for the standard errors of the elements. You can read the table that matches the $impulse \rightarrow response$ pairs, such as $x1 \rightarrow y1$, $x1 \rightarrow y2$, $x1 \rightarrow y3$, $x2 \rightarrow y1$, $x2 \rightarrow y2$, and $x2 \rightarrow y3$. In the pair $x1 \rightarrow y1$, you can see the long-run responses of $y1$ to an impulse in $x1$ (the values are 1.69281, 0.35399, 0.09090, and so on for lag 0, lag 1, lag 2, and so on, respectively).

Figure 42.45 Impulse Response in Transfer Function (IMPULSX= Option)

The VARMAX Procedure

Simple Impulse Response of Transfer Function by Variable

Variable Response\Impulse	Lag	x1	x2
y1	0	1.69281	-0.00859
	STD	0.54395	0.05361
	1	0.35399	0.01727
	STD	0.36482	0.03762
	2	0.09090	0.00714
	STD	0.17419	0.01592
	3	0.05136	0.00214
	STD	0.08203	0.00524
	4	0.04717	0.00072
	STD	0.07969	0.00229
	5	0.04620	0.00040
	STD	0.08216	0.00170
y2	0	-6.09850	2.57980
	STD	5.07849	0.50056
	1	-5.15484	0.45445
	STD	3.89665	0.40534
	2	-3.04168	0.04391
	STD	1.56519	0.13268
	3	-2.23797	-0.01376
	STD	1.15163	0.08723
	4	-1.98183	-0.01647
	STD	1.08738	0.07844
	5	-1.87415	-0.01453
	STD	0.99384	0.07250
y3	0	-0.02317	-0.01274
	STD	0.20418	0.02012
	1	1.57476	-0.01435
	STD	0.56132	0.05515
	2	1.80231	0.00398
	STD	0.61049	0.05896
	3	1.77024	0.01062
	STD	0.64476	0.06380
	4	1.70435	0.01197
	STD	0.62648	0.06353
	5	1.63913	0.01187
	STD	0.59511	0.06142

Figure 42.46 shows the responses of $y1$, $y2$, and $y3$ to a forecast error impulse in $x1$.

Figure 42.46 Plot of Impulse Response in Transfer Function

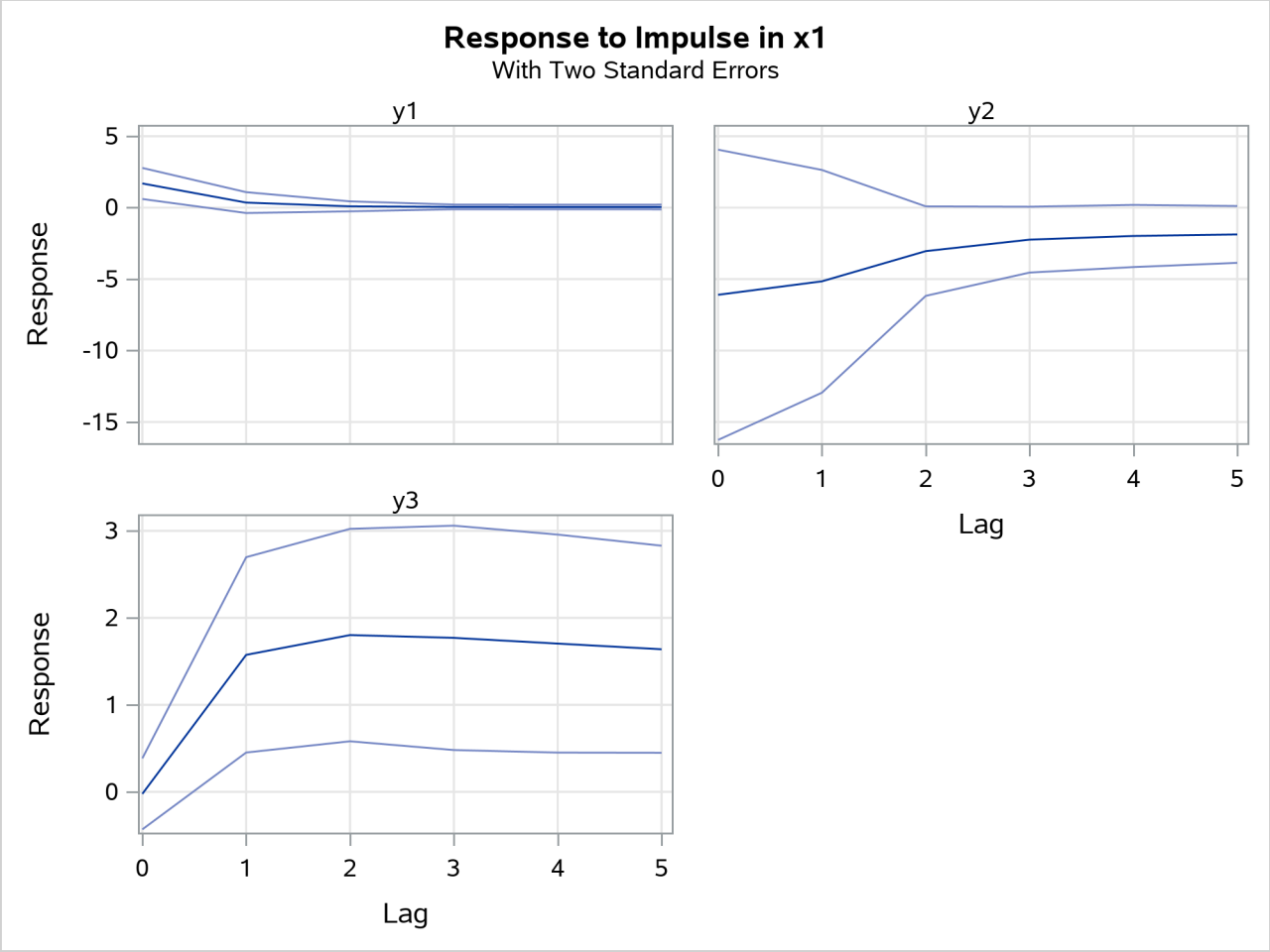


Figure 42.47 shows the accumulated impulse response in transfer function.

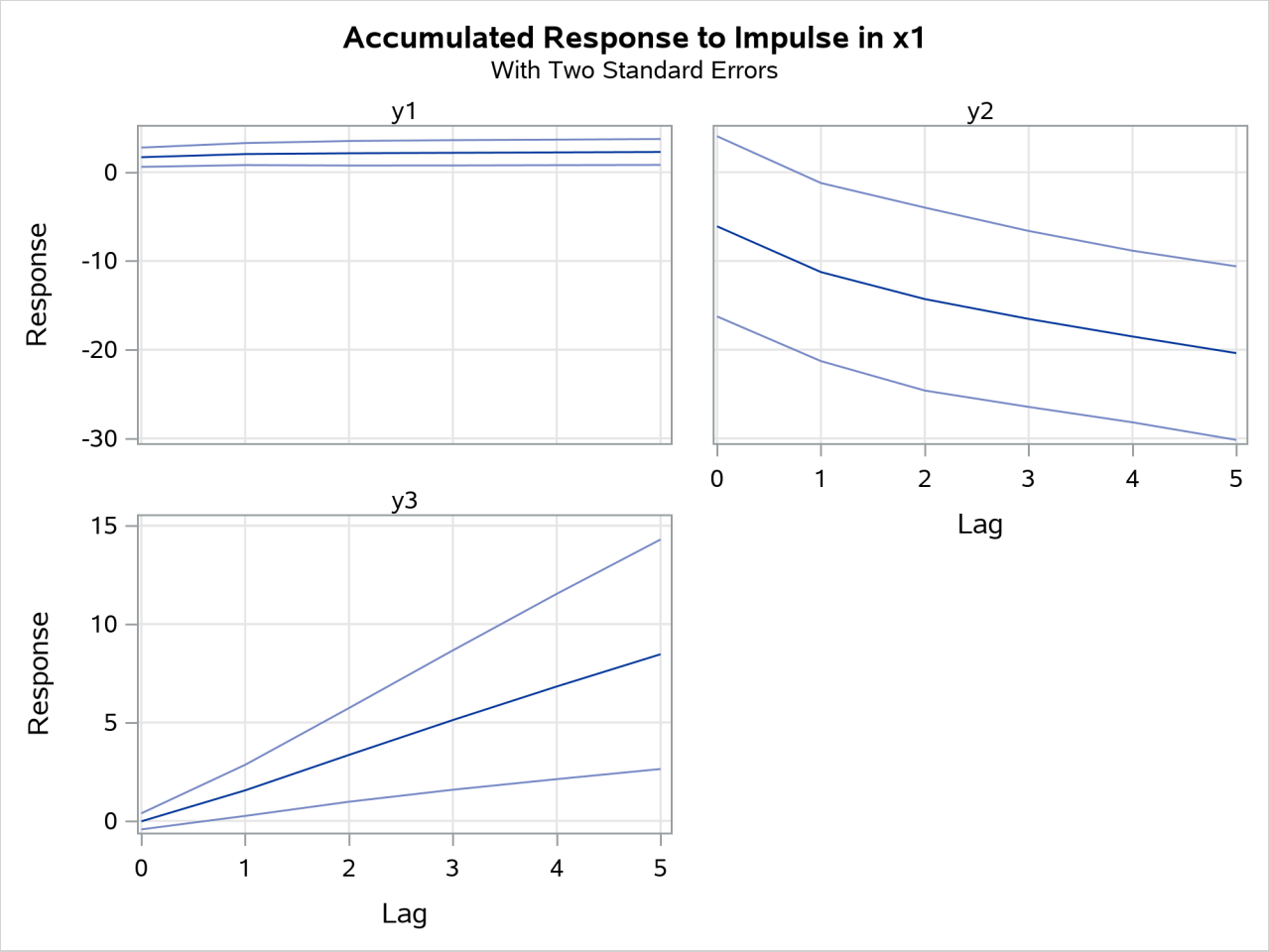
Figure 42.47 Accumulated Impulse Response in Transfer Function (IMPULSX= Option)

Accumulated Impulse Response of Transfer Function by Variable

Variable Response\Impulse	Lag	x1	x2
y1	0	1.69281	-0.00859
	STD	0.54395	0.05361
	1	2.04680	0.00868
	STD	0.36482	0.03762
	2	2.13770	0.01582
	STD	0.17419	0.01592
	3	2.18906	0.01796
	STD	0.08203	0.00524
	4	2.23623	0.01867
	STD	0.07969	0.00229
	5	2.28243	0.01907
	STD	0.08216	0.00170
y2	0	-6.09850	2.57980
	STD	5.07849	0.50056
	1	-11.25334	3.03425
	STD	3.89665	0.40534
	2	-14.29502	3.07816
	STD	1.56519	0.13268
	3	-16.53299	3.06440
	STD	1.15163	0.08723
	4	-18.51482	3.04793
	STD	1.08738	0.07844
	5	-20.38897	3.03340
	STD	0.99384	0.07250
y3	0	-0.02317	-0.01274
	STD	0.20418	0.02012
	1	1.55159	-0.02709
	STD	0.56132	0.05515
	2	3.35390	-0.02311
	STD	0.61049	0.05896
	3	5.12414	-0.01249
	STD	0.64476	0.06380
	4	6.82848	-0.00052
	STD	0.62648	0.06353
	5	8.46762	0.01135
	STD	0.59511	0.06142

Figure 42.48 shows the accumulated responses of $y1$, $y2$, and $y3$ to a forecast error impulse in $x1$.

Figure 42.48 Plot of Accumulated Impulse Response in Transfer Function



The following statements provide the impulse response function, the accumulated impulse response function, and the orthogonalized impulse response function with their standard errors for a VAR(1) model. Parts of the VARMAX procedure output are shown in Figure 42.49, Figure 42.51, and Figure 42.53.

```
proc varmax data=simul1 plot=impulse;
   model y1 y2 / p=1 noint lagmax=5
                 print=(impulse=(all))
                 printform=univariate;
run;
```

Figure 42.49 is the output in a univariate format associated with the PRINT=(IMPULSE=) option for the impulse response function. The keyword STD stands for the standard errors of the elements. The matrix in terms of the lag 0 does not print since it is the identity. In Figure 42.49, the variables $y1$ and $y2$ of the first row are impulses, and the variables $y1$ and $y2$ of the first column are responses. You can read the table matching the $impulse \rightarrow response$ pairs, such as $y1 \rightarrow y1$, $y1 \rightarrow y2$, $y2 \rightarrow y1$, and $y2 \rightarrow y2$. For example, in the pair of $y1 \rightarrow y1$ at lag 3, the response is 0.8055. This represents the impact on y1 of one-unit change in $y1$ after 3 periods. As the lag gets higher, you can see the long-run responses of $y1$ to an impulse in itself.

Figure 42.49 Impulse Response Function (IMPULSE= Option)

The VARMAX Procedure

Simple Impulse Response by Variable

Variable Response\Impulse	Lag	y1	y2
y1	1	1.15977	-0.51058
	STD	0.05508	0.05898
	2	1.06612	-0.78872
	STD	0.10450	0.10702
	3	0.80555	-0.84798
	STD	0.14522	0.14121
	4	0.47097	-0.73776
	STD	0.17191	0.15864
	5	0.14315	-0.52450
	STD	0.18214	0.16115
y2	1	0.54634	0.38499
	STD	0.05779	0.06188
	2	0.84396	-0.13073
	STD	0.08481	0.08556
	3	0.90738	-0.48124
	STD	0.10307	0.09865
	4	0.78943	-0.64856
	STD	0.12318	0.11661
	5	0.56123	-0.65275
	STD	0.14236	0.13482

Figure 42.50 shows the responses of $y1$ and $y2$ to a forecast error impulse in $y1$ with two standard errors.

Figure 42.50 Plot of Impulse Response

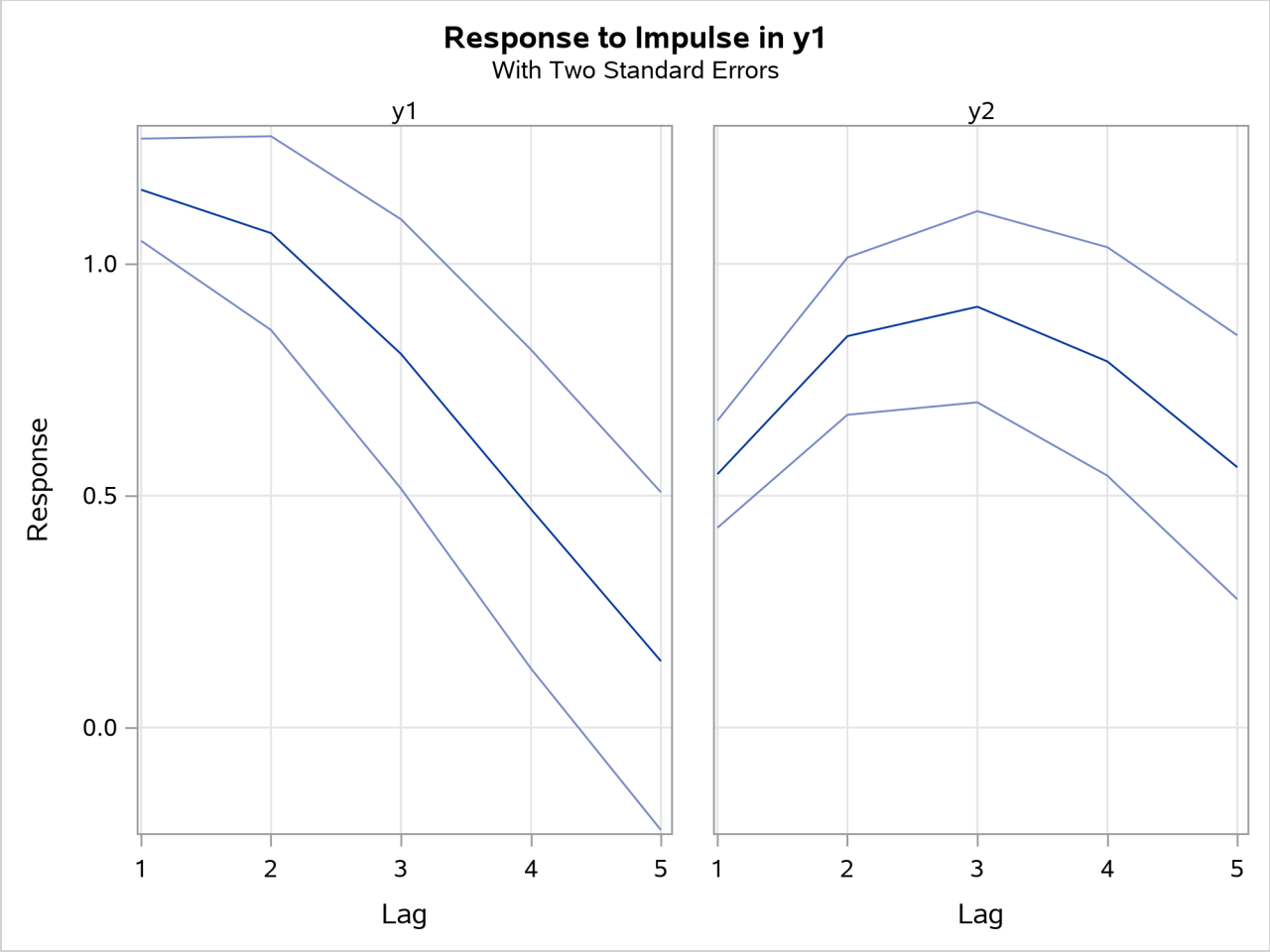


Figure 42.51 is the output in a univariate format associated with the PRINT=(IMPULSE=) option for the accumulated impulse response function. The matrix in terms of the lag 0 does not print since it is the identity.

Figure 42.51 Accumulated Impulse Response Function (IMPULSE= Option)

Accumulated Impulse Response by Variable

Variable Response\Impulse	Lag	y1	y2
y1	1	2.15977	-0.51058
	STD	0.05508	0.05898
	2	3.22589	-1.29929
	STD	0.21684	0.22776
	3	4.03144	-2.14728
	STD	0.52217	0.53649
	4	4.50241	-2.88504
	STD	0.96922	0.97088
	5	4.64556	-3.40953
	STD	1.51137	1.47122
y2	1	0.54634	1.38499
	STD	0.05779	0.06188
	2	1.39030	1.25426
	STD	0.17614	0.18392
	3	2.29768	0.77302
	STD	0.36166	0.36874
	4	3.08711	0.12447
	STD	0.65129	0.65333
	5	3.64834	-0.52829
	STD	1.07510	1.06309

Figure 42.52 shows the accumulated responses of $y1$ and $y2$ to a forecast error impulse in $y1$ with two standard errors.

Figure 42.52 Plot of Accumulated Impulse Response

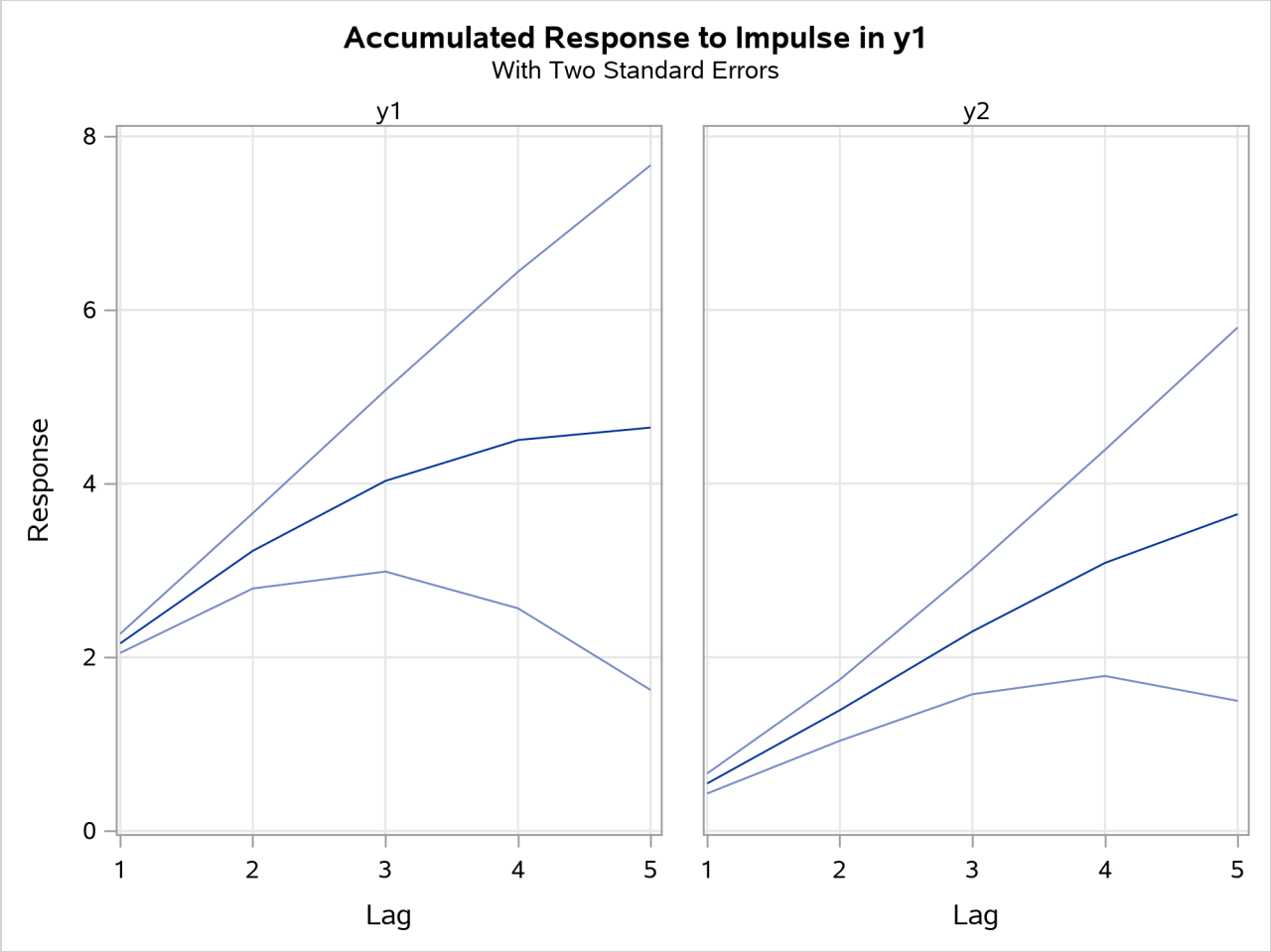


Figure 42.53 is the output in a univariate format associated with the PRINT=(IMPULSE=) option for the orthogonalized impulse response function. The two right-hand side columns, $y1$ and $y2$, represent the $y1\_innovation$ and $y2\_innovation$ variables. These are the impulses variables. The left-hand side column contains responses variables, $y1$ and $y2$. You can read the table by matching the $impulse \rightarrow response$ pairs such as $y1\_innovation \rightarrow y1$, $y1\_innovation \rightarrow y2$, $y2\_innovation \rightarrow y1$, and $y2\_innovation \rightarrow y2$.

Figure 42.53 Orthogonalized Impulse Response Function (IMPULSE= Option)

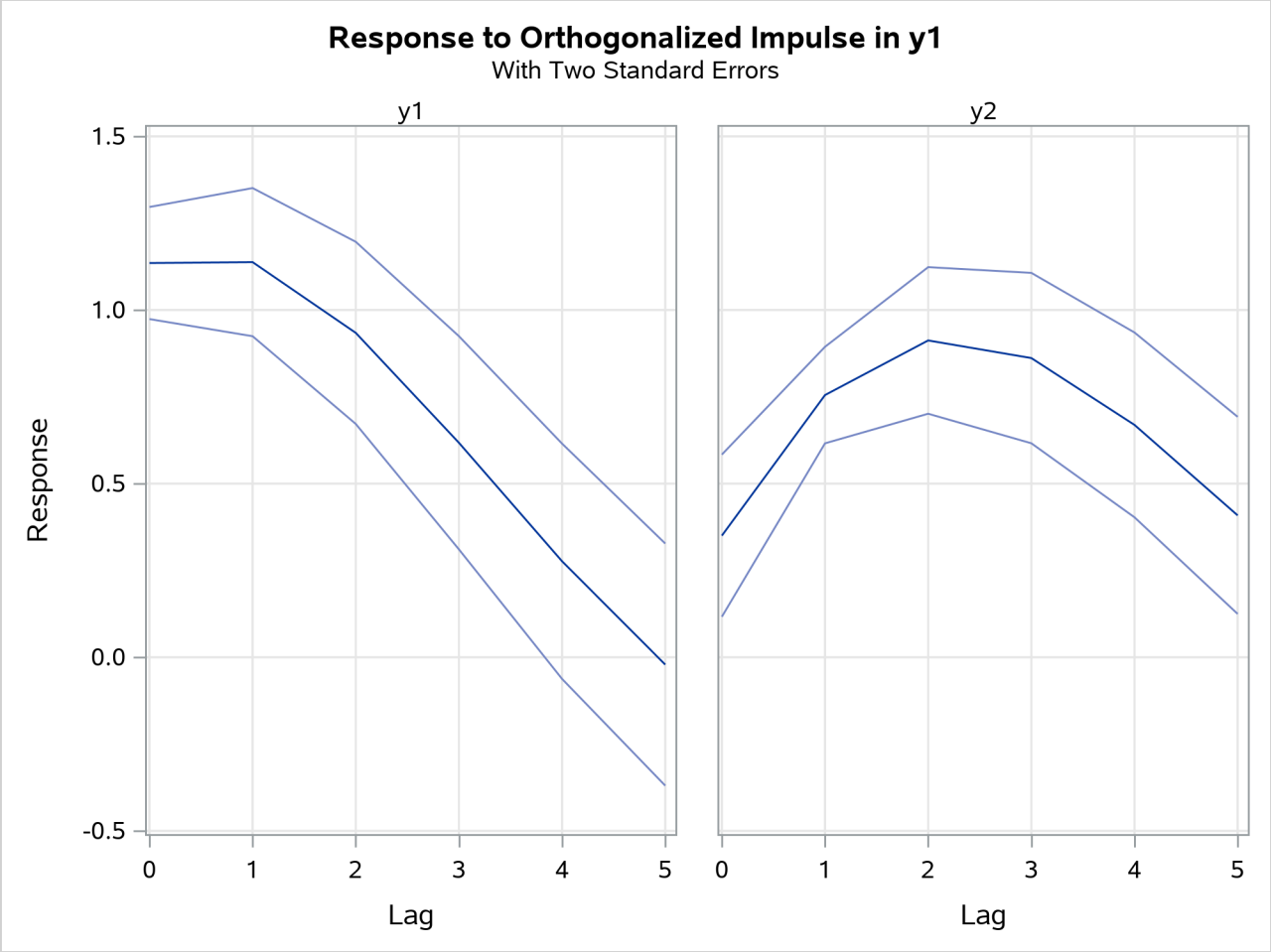
Orthogonalized Impulse Response by Variable

Variable Response\Impulse	Lag	y1	y2
y1	0	1.13523	0.00000
	STD	0.08068	0.00000
	1	1.13783	-0.58120
	STD	0.10666	0.14110
	2	0.93412	-0.89782
	STD	0.13113	0.16776
	3	0.61756	-0.96528
	STD	0.15348	0.18595
	4	0.27633	-0.83981
	STD	0.16940	0.19230
	5	-0.02115	-0.59705
	STD	0.17432	0.18830
y2	0	0.35016	1.13832
	STD	0.11676	0.08855
	1	0.75503	0.43824
	STD	0.06949	0.10937
	2	0.91231	-0.14881
	STD	0.10553	0.13565
	3	0.86158	-0.54780
	STD	0.12266	0.14825
	4	0.66909	-0.73827
	STD	0.13305	0.15846
	5	0.40856	-0.74304
	STD	0.14189	0.16765

In Figure 42.4, there is a positive correlation between ε_{1t} and ε_{2t}. Therefore, shock in $y1$ can be accompanied by a shock in $y2$ in the same period. For example, in the pair of $y1\_innovation \rightarrow y2$, you can see the long-run responses of $y2$ to an impulse in $y1\_innovation$.

Figure 42.54 shows the orthogonalized responses of $y1$ and $y2$ to a forecast error impulse in $y1$ with two standard errors.

Figure 42.54 Plot of Orthogonalized Impulse Response



Forecasting

The optimal (minimum MSE) l-step-ahead forecast of $\mathbf{y}_{t+l}$ is

$$\mathbf{y}_{t+l|t} = \sum_{j=1}^{p} \Phi_j \mathbf{y}_{t+l-j|t} + \sum_{j=0}^{s} \Theta_j^* \mathbf{x}_{t+l-j|t} - \sum_{j=l}^{q} \Theta_j \boldsymbol{\epsilon}_{t+l-j}, \quad l \le q$$

$$\mathbf{y}_{t+l|t} = \sum_{j=1}^{p} \Phi_j \mathbf{y}_{t+l-j|t} + \sum_{j=0}^{s} \Theta_j^* \mathbf{x}_{t+l-j|t}, \quad l > q$$

where $\mathbf{y}_{t+l-j|t} = \mathbf{y}_{t+l-j}$ and $\mathbf{x}_{t+l-j|t} = \mathbf{x}_{t+l-j}$ for $l \le j$. For information about the forecasts $\mathbf{x}_{t+l-j|t}$, see the section "State Space Representation" on page 3040.

Covariance Matrices of Prediction Errors without Exogenous (Independent) Variables

Under the stationarity assumption, the optimal (minimum MSE) l-step-ahead forecast of $\mathbf{y}_{t+l}$ has an infinite moving-average form, $\mathbf{y}_{t+l|t} = \sum_{j=l}^{\infty} \Psi_j \boldsymbol{\epsilon}_{t+l-j}$. The prediction error of the optimal l-step-ahead forecast is $\mathbf{e}_{t+l|t} = \mathbf{y}_{t+l} - \mathbf{y}_{t+l|t} = \sum_{j=0}^{l-1} \Psi_j \boldsymbol{\epsilon}_{t+l-j}$, with zero mean and covariance matrix,

$$\Sigma(l) = \mathrm{Cov}(\mathbf{e}_{t+l|t}) = \sum_{j=0}^{l-1} \Psi_j \Sigma \Psi_j' = \sum_{j=0}^{l-1} \Psi_j^o \Psi_j^{o'}$$

where $\Psi_j^o = \Psi_j P$ with a lower triangular matrix P such that $\Sigma = PP'$. Under the assumption of normality of the $\boldsymbol{\epsilon}_t$, the l-step-ahead prediction error $\mathbf{e}_{t+l|t}$ is also normally distributed as multivariate $N(0, \Sigma(l))$. Hence, it follows that the diagonal elements $\sigma_{ii}^2(l)$ of $\Sigma(l)$ can be used, together with the point forecasts $y_{i,t+l|t}$, to construct l-step-ahead prediction intervals of the future values of the component series, $y_{i,t+l}$.

The following statements use the COVPE option to compute the covariance matrices of the prediction errors for a VAR(1) model. The parts of the VARMAX procedure output are shown in Figure 42.55 and Figure 42.56.

```
proc varmax data=simul1;
   model y1 y2 / p=1 noint lagmax=5
                 printform=both
                 print=(decompose(5) impulse=(all) covpe(5));
run;
```

Figure 42.55 is the output in a matrix format associated with the COVPE option for the prediction error covariance matrices.

Figure 42.55 Covariances of Prediction Errors (COVPE Option)

The VARMAX Procedure

Prediction Error Covariances

Lead	Variable	y1	y2
1	y1	1.28875	0.39751
	y2	0.39751	1.41839
2	y1	2.92119	1.00189
	y2	1.00189	2.18051
3	y1	4.59984	1.98771
	y2	1.98771	3.03498
4	y1	5.91299	3.04856
	y2	3.04856	4.07738
5	y1	6.69463	3.85346
	y2	3.85346	5.07010

Figure 42.56 is the output in a univariate format associated with the COVPE option for the prediction error covariances. This printing format more easily explains the prediction error covariances of each variable.

Figure 42.56 Covariances of Prediction Errors

Prediction Error Covariances by Variable			
Variable	**Lead**	**y1**	**y2**
y1	**1**	1.28875	0.39751
	2	2.92119	1.00189
	3	4.59984	1.98771
	4	5.91299	3.04856
	5	6.69463	3.85346
y2	**1**	0.39751	1.41839
	2	1.00189	2.18051
	3	1.98771	3.03498
	4	3.04856	4.07738
	5	3.85346	5.07010

Covariance Matrices of Prediction Errors in the Presence of Exogenous (Independent) Variables

Exogenous variables can be both stochastic and nonstochastic (deterministic) variables. Considering the forecasts in the VARMAX(p,q,s) model, there are two cases.

When exogenous (independent) variables are stochastic (future values not specified):

As defined in the section "State Space Representation" on page 3040, $\mathbf{y}_{t+l|t}$ has the representation

$$\mathbf{y}_{t+l|t} = \sum_{j=l}^{\infty} V_j \mathbf{a}_{t+l-j} + \sum_{j=l}^{\infty} \Psi_j \boldsymbol{\epsilon}_{t+l-j}$$

and hence

$$\mathbf{e}_{t+l|t} = \sum_{j=0}^{l-1} V_j \mathbf{a}_{t+l-j} + \sum_{j=0}^{l-1} \Psi_j \boldsymbol{\epsilon}_{t+l-j}$$

Therefore, the covariance matrix of the l-step-ahead prediction error is given as

$$\Sigma(l) = \mathrm{Cov}(\mathbf{e}_{t+l|t}) = \sum_{j=0}^{l-1} V_j \Sigma_a V_j' + \sum_{j=0}^{l-1} \Psi_j \Sigma_\epsilon \Psi_j'$$

where Σ_a is the covariance of the white noise series $\mathbf{a}_t$, and $\mathbf{a}_t$ is the white noise series for the VARMA(p,q) model of exogenous (independent) variables, which is assumed not to be correlated with $\boldsymbol{\epsilon}_t$ or its lags.

When future exogenous (independent) variables are specified:

The optimal forecast $\mathbf{y}_{t+l|t}$ of $\mathbf{y}_t$ conditioned on the past information and also on known future values $\mathbf{x}_{t+1}, \ldots, \mathbf{x}_{t+l}$ can be represented as

$$\mathbf{y}_{t+l|t} = \sum_{j=0}^{\infty} \Psi_j^* \mathbf{x}_{t+l-j} + \sum_{j=l}^{\infty} \Psi_j \boldsymbol{\epsilon}_{t+l-j}$$

and the forecast error is

$$\mathbf{e}_{t+l|t} = \sum_{j=0}^{l-1} \Psi_j \boldsymbol{\epsilon}_{t+l-j}$$

Thus, the covariance matrix of the l-step-ahead prediction error is given as

$$\Sigma(l) = \mathrm{Cov}(\mathbf{e}_{t+l|t}) = \sum_{j=0}^{l-1} \Psi_j \Sigma_\epsilon \Psi_j'$$

Decomposition of Prediction Error Covariances

In the relation $\Sigma(l) = \sum_{j=0}^{l-1} \Psi_j^o \Psi_j^{o\prime}$, the diagonal elements can be interpreted as providing a decomposition of the l-step-ahead prediction error covariance $\sigma_{ii}^2(l)$ for each component series y_{it} into contributions from the components of the standardized innovations $\boldsymbol{\epsilon}_t$.

If you denote the (i, n) element of Ψ_j^o by $\psi_{j,in}$, the MSE of $y_{i,t+h|t}$ is

$$\mathrm{MSE}(y_{i,t+h|t}) = \mathrm{E}(y_{i,t+h} - y_{i,t+h|t})^2 = \sum_{j=0}^{l-1} \sum_{n=1}^{k} \psi_{j,in}^2$$

Note that $\sum_{j=0}^{l-1} \psi_{j,in}^2$ is interpreted as the contribution of innovations in variable n to the prediction error covariance of the l-step-ahead forecast of variable i.

The proportion, $\omega_{l,in}$, of the l-step-ahead forecast error covariance of variable i accounting for the innovations in variable n is

$$\omega_{l,in} = \sum_{j=0}^{l-1} \psi_{j,in}^2 / \mathrm{MSE}(y_{i,t+h|t})$$

The following statements use the DECOMPOSE option to compute the decomposition of prediction error covariances and their proportions for a VAR(1) model:

```
proc varmax data=simul1;
   model y1 y2 / p=1 noint print=(decompose(15))
                 printform=univariate;
run;
```

The proportions of decomposition of prediction error covariances of two variables are given in Figure 42.57. The output explains that about 91.356% of the one-step-ahead prediction error covariances of the variable y_{2t} is accounted for by its own innovations and about 8.644% is accounted for by y_{1t} innovations.

Figure 42.57 Decomposition of Prediction Error Covariances (DECOMPOSE Option)

Proportions of Prediction Error Covariances by Variable			
Variable	**Lead**	**y1**	**y2**
y1	**1**	1.00000	0.00000
	2	0.88436	0.11564
	3	0.75132	0.24868
	4	0.64897	0.35103
	5	0.58460	0.41540
y2	**1**	0.08644	0.91356
	2	0.31767	0.68233
	3	0.50247	0.49753
	4	0.55607	0.44393
	5	0.53549	0.46451

Forecasting of the Centered Series

If the CENTER option is specified, the sample mean vector is added to the forecast.

Forecasting of the Differenced Series

If dependent (endogenous) variables are differenced, the final forecasts and their prediction error covariances are produced by integrating those of the differenced series. However, if the PRIOR option is specified, the forecasts and their prediction error variances of the differenced series are produced.

Let $\mathbf{z}_t$ be the original series with some appended zero values that correspond to the unobserved past observations. Let $\Delta(B)$ be the $k \times k$ matrix polynomial in the backshift operator that corresponds to the differencing specified by the MODEL statement. The off-diagonal elements of Δ_i are zero, and the diagonal elements can be different. Then $\mathbf{y}_t = \Delta(B)\mathbf{z}_t$.

This gives the relationship

$$\mathbf{z}_t = \Delta^{-1}(B)\mathbf{y}_t = \sum_{j=0}^{\infty} \Lambda_j \mathbf{y}_{t-j}$$

where $\Delta^{-1}(B) = \sum_{j=0}^{\infty} \Lambda_j B^j$ and $\Lambda_0 = I_k$.

The l-step-ahead prediction of $\mathbf{z}_{t+l}$ is

$$\mathbf{z}_{t+l|t} = \sum_{j=0}^{l-1} \Lambda_j \mathbf{y}_{t+l-j|t} + \sum_{j=l}^{\infty} \Lambda_j \mathbf{y}_{t+l-j}$$

The l-step-ahead prediction error of $\mathbf{z}_{t+l}$ is

$$\sum_{j=0}^{l-1} \Lambda_j \left(\mathbf{y}_{t+l-j} - \mathbf{y}_{t+l-j|t}\right) = \sum_{j=0}^{l-1} \left(\sum_{u=0}^{j} \Lambda_u \Psi_{j-u} \right) \boldsymbol{\epsilon}_{t+l-j}$$

Letting $\Sigma_{\mathbf{z}}(0) = 0$, the covariance matrix of the l-step-ahead prediction error of $\mathbf{z}_{t+l}$, $\Sigma_{\mathbf{z}}(l)$, is

$$\begin{aligned}\Sigma_{\mathbf{z}}(l) &= \sum_{j=0}^{l-1}\left(\sum_{u=0}^{j}\Lambda_u\Psi_{j-u}\right)\Sigma_\epsilon\left(\sum_{u=0}^{j}\Lambda_u\Psi_{j-u}\right)' \\ &= \Sigma_{\mathbf{z}}(l-1) + \left(\sum_{j=0}^{l-1}\Lambda_j\Psi_{l-1-j}\right)\Sigma_\epsilon\left(\sum_{j=0}^{l-1}\Lambda_j\Psi_{l-1-j}\right)'\end{aligned}$$

If there are stochastic exogenous (independent) variables, the covariance matrix of the l-step-ahead prediction error of $\mathbf{z}_{t+l}$, $\Sigma_{\mathbf{z}}(l)$, is

$$\begin{aligned}\Sigma_{\mathbf{z}}(l) &= \Sigma_{\mathbf{z}}(l-1) + \left(\sum_{j=0}^{l-1}\Lambda_j\Psi_{l-1-j}\right)\Sigma_\epsilon\left(\sum_{j=0}^{l-1}\Lambda_j\Psi_{l-1-j}\right)' \\ &\quad + \left(\sum_{j=0}^{l-1}\Lambda_j V_{l-1-j}\right)\Sigma_a\left(\sum_{j=0}^{l-1}\Lambda_j V_{l-1-j}\right)'\end{aligned}$$

Tentative Order Selection

Sample Cross-Covariance and Cross-Correlation Matrices

Given a stationary multivariate time series $\mathbf{y}_t$, cross-covariance matrices are

$$\Gamma(l) = \mathrm{E}[(\mathbf{y}_t - \boldsymbol{\mu})(\mathbf{y}_{t+l} - \boldsymbol{\mu})']$$

where $\boldsymbol{\mu} = \mathrm{E}(\mathbf{y}_t)$, and cross-correlation matrices are

$$\rho(l) = D^{-1}\Gamma(l)D^{-1}$$

where D is a diagonal matrix with the standard deviations of the components of $\mathbf{y}_t$ on the diagonal.

The sample cross-covariance matrix at lag l, denoted as $C(l)$, is computed as

$$\hat{\Gamma}(l) = C(l) = \frac{1}{T}\sum_{t=1}^{T-l}\tilde{\mathbf{y}}_t\tilde{\mathbf{y}}'_{t+l}$$

where $\tilde{\mathbf{y}}_t$ is the centered data and T is the number of nonmissing observations. Thus, the (i, j) element of $\hat{\Gamma}(l)$ is $\hat{\gamma}_{ij}(l) = c_{ij}(l)$. The sample cross-correlation matrix at lag l is computed as

$$\hat{\rho}_{ij}(l) = c_{ij}(l)/[c_{ii}(0)c_{jj}(0)]^{1/2}, \quad i, j = 1, \ldots, k$$

The following statements use the CORRY option to compute the sample cross-correlation matrices and their summary indicator plots in terms of $+$, $-$, and $\cdot$, where + indicates significant positive cross-correlations, $-$ indicates significant negative cross-correlations, and $\cdot$ indicates insignificant cross-correlations:

```
proc varmax data=simul1;
   model y1 y2 / p=1 noint lagmax=3 print=(corry)
                 printform=univariate;
run;
```

Figure 42.58 shows the sample cross-correlation matrices of y_{1t} and y_{2t}. As shown, the sample autocorrelation functions for each variable decay quickly, but are significant with respect to two standard errors.

Figure 42.58 Cross-Correlations (CORRY Option)

The VARMAX Procedure

Cross Correlations of Dependent Series by Variable

Variable	Lag	y1	y2
y1	0	1.00000	0.67041
	1	0.83143	0.84330
	2	0.56094	0.81972
	3	0.26629	0.66154
y2	0	0.67041	1.00000
	1	0.29707	0.77132
	2	-0.00936	0.48658
	3	-0.22058	0.22014

Schematic Representation of Cross Correlations

Variable/Lag	0	1	2	3
y1	++	++	++	++
y2	++	++	.+	−+

+ is > 2*std error, - is < -2*std error, . is between

Partial Autoregressive Matrices

For each $m = 1, 2, \ldots, p$, you can define a sequence of matrices Φ_{mm}, which is called the partial autoregression matrices of lag m, as the solution for Φ_{mm} to the Yule-Walker equations of order m,

$$\Gamma(l) = \sum_{i=1}^{m} \Gamma(l-i)\Phi_{im}', \quad l = 1, 2, \ldots, m$$

The sequence of the partial autoregression matrices Φ_{mm} of order m has the characteristic property that if the process follows the AR(p), then $\Phi_{pp} = \Phi_p$ and $\Phi_{mm} = 0$ for $m > p$. Hence, the matrices Φ_{mm} have the cutoff property for a VAR(p) model, and so they can be useful in the identification of the order of a pure VAR model.

The following statements use the PARCOEF option to compute the partial autoregression matrices:

```
proc varmax data=simul1;
   model y1 y2 / p=1 noint lagmax=3
                 printform=univariate
                print=(corry parcoef pcorr
```

```
                              pcancorr roots);
run;
```

Figure 42.59 shows that the model can be obtained by an AR order $m = 1$ since partial autoregression matrices are insignificant after lag 1 with respect to two standard errors. The matrix for lag 1 is the same as the Yule-Walker autoregressive matrix.

Figure 42.59 Partial Autoregression Matrices (PARCOEF Option)

The VARMAX Procedure

Partial Autoregression			
Lag	Variable	y1	y2
1	y1	1.14844	-0.50954
	y2	0.54985	0.37409
2	y1	-0.00724	0.05138
	y2	0.02409	0.05909
3	y1	-0.02578	0.03885
	y2	-0.03720	0.10149

Schematic Representation of Partial Autoregression			
Variable/Lag	1	2	3
y1	+-	. .	. .
y2	++	. .	. .
+ is > 2*std error, - is < -2*std error, . is between			

Partial Correlation Matrices

Define the forward autoregression

$$\mathbf{y}_t = \sum_{i=1}^{m-1} \Phi_{i,m-1}\mathbf{y}_{t-i} + \mathbf{u}_{m,t}$$

and the backward autoregression

$$\mathbf{y}_{t-m} = \sum_{i=1}^{m-1} \Phi^*_{i,m-1}\mathbf{y}_{t-m+i} + \mathbf{u}^*_{m,t-m}$$

The matrices $P(m)$ defined by Ansley and Newbold (1979) are given by

$$P(m) = \Sigma^{*1/2}_{m-1}\Phi'_{mm}\Sigma^{-1/2}_{m-1}$$

where

$$\Sigma_{m-1} = \mathrm{Cov}(\mathbf{u}_{m,t}) = \Gamma(0) - \sum_{i=1}^{m-1}\Gamma(-i)\Phi'_{i,m-1}$$

and

$$\Sigma^*_{m-1} = \mathrm{Cov}(\mathbf{u}^*_{m,t-m}) = \Gamma(0) - \sum_{i=1}^{m-1} \Gamma(m-i)\Phi^{*'}_{m-i,m-1}$$

$P(m)$ are the partial cross-correlation matrices at lag m between the elements of $\mathbf{y}_t$ and $\mathbf{y}_{t-m}$, given $\mathbf{y}_{t-1}, \ldots, \mathbf{y}_{t-m+1}$. The matrices $P(m)$ have the cutoff property for a VAR(p) model, and so they can be useful in the identification of the order of a pure VAR structure.

The following statements use the PCORR option to compute the partial cross-correlation matrices:

```
proc varmax data=simul1;
   model y1 y2 / p=1 noint lagmax=3
                 print=(pcorr)
                 printform=univariate;
run;
```

The partial cross-correlation matrices in Figure 42.60 are insignificant after lag 1 with respect to two standard errors. This indicates that an AR order of $m = 1$ can be an appropriate choice.

Figure 42.60 Partial Correlations (PCORR Option)

The VARMAX Procedure

Partial Cross Correlations by Variable

Variable	Lag	y1	y2
y1	1	0.80348	0.42672
	2	0.00276	0.03978
	3	-0.01091	0.00032
y2	1	-0.30946	0.71906
	2	0.04676	0.07045
	3	0.01993	0.10676

Schematic Representation of Partial Cross Correlations

Variable/Lag	1	2	3
y1	++	. .	. .
y2	−+	. .	. .

+ is > 2*std error, - is < -2*std error, . is between

Partial Canonical Correlation Matrices

The partial canonical correlations at lag m between the vectors $\mathbf{y}_t$ and $\mathbf{y}_{t-m}$, given $\mathbf{y}_{t-1}, \ldots, \mathbf{y}_{t-m+1}$, are $1 \geq \rho_1(m) \geq \rho_2(m) \cdots \geq \rho_k(m)$. The partial canonical correlations are the canonical correlations between the residual series $\mathbf{u}_{m,t}$ and $\mathbf{u}^*_{m,t-m}$, where $\mathbf{u}_{m,t}$ and $\mathbf{u}^*_{m,t-m}$ are defined in the previous section. Thus, the squared partial canonical correlations $\rho^2_i(m)$ are the eigenvalues of the matrix

$$\{\mathrm{Cov}(\mathbf{u}_{m,t})\}^{-1}\mathrm{E}(\mathbf{u}_{m,t}\mathbf{u}^{*'}_{m,t-m})\{\mathrm{Cov}(\mathbf{u}^*_{m,t-m})\}^{-1}\mathrm{E}(\mathbf{u}^*_{m,t-m}\mathbf{u}'_{m,t}) = \Phi^{*'}_{mm}\Phi'_{mm}$$

It follows that the test statistic to test for $\Phi_m = 0$ in the VAR model of order $m > p$ is approximately

$$(T-m)\ \mathrm{tr}\ \{\Phi^{*'}_{mm}\Phi'_{mm}\} \approx (T-m)\sum_{i=1}^{k}\rho_i^2(m)$$

and has an asymptotic chi-square distribution with k^2 degrees of freedom for $m > p$.

The following statements use the PCANCORR option to compute the partial canonical correlations:

```
proc varmax data=simul1;
   model y1 y2 / p=1 noint lagmax=3 print=(pcancorr);
run;
```

Figure 42.61 shows that the partial canonical correlations $\rho_i(m)$ between $\mathbf{y}_t$ and $\mathbf{y}_{t-m}$ are $\{0.918, 0.773\}$, $\{0.092, 0.018\}$, and $\{0.109, 0.011\}$ for lags m =1 to 3. After lag m =1, the partial canonical correlations are insignificant with respect to the 0.05 significance level, indicating that an AR order of $m = 1$ can be an appropriate choice.

Figure 42.61 Partial Canonical Correlations (PCANCORR Option)

The VARMAX Procedure

Partial Canonical Correlations

Lag	Correlation1	Correlation2	DF	Chi-Square	Pr > ChiSq
1	0.91783	0.77335	4	142.61	<.0001
2	0.09171	0.01816	4	0.86	0.9307
3	0.10861	0.01078	4	1.16	0.8854

The Minimum Information Criterion (MINIC) Method

The minimum information criterion (MINIC) method can tentatively identify the orders of a VARMA(p,q) process (Spliid 1983; Koreisha and Pukkila 1989; Quinn 1980). The first step of this method is to obtain estimates of the innovations series, $\boldsymbol{\epsilon}_t$, from the VAR(p_ϵ), where p_ϵ is chosen sufficiently large. The choice of the autoregressive order, p_ϵ, is determined by use of a selection criterion. From the selected VAR(p_ϵ) model, you obtain estimates of residual series

$$\tilde{\boldsymbol{\epsilon}_t} = \mathbf{y}_t - \sum_{i=1}^{p_\epsilon}\hat{\Phi}_i^{p_\epsilon}\mathbf{y}_{t-i} - \hat{\boldsymbol{\delta}}^{p_\epsilon},\ \ t = p_\epsilon+1,\ldots,T$$

In the second step, you select the order (p,q) of the VARMA model for p in $(p_{min} : p_{max})$ and q in $(q_{min} : q_{max})$

$$\mathbf{y}_t = \boldsymbol{\delta} + \sum_{i=1}^{p}\Phi_i\mathbf{y}_{t-i} - \sum_{i=1}^{q}\Theta_i\tilde{\boldsymbol{\epsilon}}_{t-i} + \boldsymbol{\epsilon}_t$$

which minimizes a selection criterion like SBC or HQ.

According to Lütkepohl (1993), the information criteria, namely Akaike's information criterion (AIC), the corrected Akaike's information criterion (AICC), the final prediction error criterion (FPE), the Hannan-Quinn criterion (HQC), and the Schwarz Bayesian criterion (SBC), are defined as

$$
\begin{aligned}
\text{AIC} &= \log(|\tilde{\Sigma}|) + 2r_b k/T \\
\text{AICC} &= \log(|\tilde{\Sigma}|) + 2r_b k/(T - r_b) \\
\text{FPE} &= \left(\frac{T + r_b}{T - r_b}\right)^k |\tilde{\Sigma}| \\
\text{HQC} &= \log(|\tilde{\Sigma}|) + 2r_b k \log(\log(T))/T \\
\text{SBC} &= \log(|\tilde{\Sigma}|) + r_b k \log(T)/T
\end{aligned}
$$

where $\tilde{\Sigma}$ is the maximum likelihood estimate of the innovation covariance matrix Σ, r_b is the number of parameters in each mean equation, k is the number of dependent variables, and T is the number of observations used to estimate the model. Compared to the definitions of AIC, AICC, HQC, and SBC discussed in the section "Multivariate Model Diagnostic Checks" on page 3085, the preceding definitions omit some constant terms and are normalized by T. More specifically, only the parameters in each of the mean equations are counted; the parameters in the innovation covariance matrix Σ are not counted.

The following statements use the MINIC= option to compute a table that contains the information criterion associated with various AR and MA orders:

```
proc varmax data=simul1;
   model y1 y2 / p=1 noint minic=(p=3 q=3);
run;
```

Figure 42.62 shows the output associated with the MINIC= option. The criterion takes the smallest value at AR order 1.

Figure 42.62 MINIC= Option

The VARMAX Procedure

Minimum Information Criterion Based on AICC				
Lag	**MA 0**	**MA 1**	**MA 2**	**MA 3**
AR 0	3.3574947	3.0331352	2.7080996	2.3049869
AR 1	0.5544431	0.6146887	0.6771732	0.7517968
AR 2	0.6369334	0.6729736	0.7610413	0.8481559
AR 3	0.7235629	0.7551756	0.8053765	0.8654079

VAR and VARX Modeling

The pth-order VAR process is written as

$$
\mathbf{y}_t - \boldsymbol{\mu} = \sum_{i=1}^{p} \Phi_i (\mathbf{y}_{t-i} - \boldsymbol{\mu}) + \boldsymbol{\epsilon}_t \quad \text{or} \quad \Phi(B)(\mathbf{y}_t - \boldsymbol{\mu}) = \boldsymbol{\epsilon}_t
$$

with $\Phi(B) = I_k - \sum_{i=1}^{p} \Phi_i B^i$.

Equivalently, it can be written as

$$\mathbf{y}_t = \boldsymbol{\delta} + \sum_{i=1}^{p} \Phi_i \mathbf{y}_{t-i} + \boldsymbol{\epsilon}_t \quad \text{or} \quad \Phi(B)\mathbf{y}_t = \boldsymbol{\delta} + \boldsymbol{\epsilon}_t$$

with $\boldsymbol{\delta} = (I_k - \sum_{i=1}^{p} \Phi_i)\boldsymbol{\mu}$.

Stationarity

For stationarity, the VAR process must be expressible in the convergent causal infinite MA form as

$$\mathbf{y}_t = \boldsymbol{\mu} + \sum_{j=0}^{\infty} \Psi_j \boldsymbol{\epsilon}_{t-j}$$

where $\Psi(B) = \Phi(B)^{-1} = \sum_{j=0}^{\infty} \Psi_j B^j$ with $\sum_{j=0}^{\infty} ||\Psi_j|| < \infty$, where $||A||$ denotes a norm for the matrix A such as $||A||^2 = \mathrm{tr}\{A'A\}$. The matrix Ψ_j can be recursively obtained from the relation $\Phi(B)\Psi(B) = I$; it is

$$\Psi_j = \Phi_1\Psi_{j-1} + \Phi_2\Psi_{j-2} + \cdots + \Phi_p\Psi_{j-p}$$

where $\Psi_0 = I_k$ and $\Psi_j = 0$ for $j < 0$.

The stationarity condition is satisfied if all roots of $|\Phi(z)| = 0$ are outside of the unit circle. The stationarity condition is equivalent to the condition in the corresponding VAR(1) representation, $\mathbf{Y}_t = \Phi\mathbf{Y}_{t-1} + \boldsymbol{\varepsilon}_t$, that all eigenvalues of the $kp \times kp$ companion matrix Φ be less than one in absolute value, where $\mathbf{Y}_t = (\mathbf{y}_t', \ldots, \mathbf{y}_{t-p+1}')'$, $\boldsymbol{\varepsilon}_t = (\boldsymbol{\epsilon}_t', 0', \ldots, 0')'$, and

$$\Phi = \begin{bmatrix} \Phi_1 & \Phi_2 & \cdots & \Phi_{p-1} & \Phi_p \\ I_k & 0 & \cdots & 0 & 0 \\ 0 & I_k & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I_k & 0 \end{bmatrix}$$

If the stationarity condition is not satisfied, a nonstationary model (a differenced model or an error correction model) might be more appropriate.

The following statements estimate a VAR(1) model and use the ROOTS option to compute the characteristic polynomial roots:

```
proc varmax data=simul1;
   model y1 y2 / p=1 noint print=(roots);
run;
```

Figure 42.63 shows the output associated with the ROOTS option, which indicates that the series is stationary since the modulus of the eigenvalue is less than one.

Figure 42.63 Stationarity (ROOTS Option)

The VARMAX Procedure

Roots of AR Characteristic Polynomial

Index	Real	Imaginary	Modulus	Radian	Degree
1	0.77238	0.35899	0.8517	0.4351	24.9284
2	0.77238	-0.35899	0.8517	-0.4351	-24.9284

Parameter Estimation

Consider the stationary VAR(p) model

$$\mathbf{y}_t = \boldsymbol{\delta} + \sum_{i=1}^{p} \Phi_i \mathbf{y}_{t-i} + \boldsymbol{\epsilon}_t$$

where $\mathbf{y}_{-p+1}, \ldots, \mathbf{y}_0$ are assumed to be available (for convenience of notation). This can be represented by the general form of the multivariate linear model,

$$Y = XB + E \quad \text{or} \quad \mathbf{y} = (X \otimes I_k)\boldsymbol{\beta} + \mathbf{e}$$

where

$$\begin{aligned}
Y &= (\mathbf{y}_1, \ldots, \mathbf{y}_T)' \\
B &= (\boldsymbol{\delta}, \Phi_1, \ldots, \Phi_p)' \\
X &= (X_0, \ldots, X_{T-1})' \\
X_t &= (1, \mathbf{y}_t', \ldots, \mathbf{y}_{t-p+1}')' \\
E &= (\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_T)' \\
\mathbf{y} &= \mathrm{vec}(Y') \\
\boldsymbol{\beta} &= \mathrm{vec}(B') \\
\mathbf{e} &= \mathrm{vec}(E')
\end{aligned}$$

with *vec* denoting the column stacking operator.

The conditional least squares estimator of $\boldsymbol{\beta}$ is

$$\hat{\boldsymbol{\beta}} = ((X'X)^{-1}X' \otimes I_k)\mathbf{y}$$

and the estimate of Σ is

$$\hat{\Sigma} = (T - (kp+1))^{-1} \sum_{t=1}^{T} \hat{\boldsymbol{\epsilon}}_t \hat{\boldsymbol{\epsilon}}_t'$$

where $\hat{\boldsymbol{\epsilon}}_t$ is the residual vectors. Consistency and asymptotic normality of the LS estimator are that

$$\sqrt{T}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \xrightarrow{d} N(0, \Gamma_p^{-1} \otimes \Sigma)$$

where $X'X/T$ converges in probability to Γ_p and $\xrightarrow{d}$ denotes convergence in distribution.

The (conditional) maximum likelihood estimator in the VAR(p) model is equal to the (conditional) least squares estimator on the assumption of normality of the error vectors.

Asymptotic Distributions of Impulse Response Functions

As before, *vec* denotes the column stacking operator and *vech* is the corresponding operator that stacks the elements on and below the diagonal. For any $k \times k$ matrix A, the commutation matrix K_k is defined as $K_k \mathrm{vec}(A) = \mathrm{vec}(A')$; the duplication matrix D_k is defined as $D_k \mathrm{vech}(A) = \mathrm{vec}(A)$; the elimination matrix L_k is defined as $L_k \mathrm{vec}(A) = \mathrm{vech}(A)$.

The asymptotic distribution of the impulse response function (Lütkepohl 1993) is

$$\sqrt{T}\mathrm{vec}(\hat{\Psi}_j - \Psi_j) \xrightarrow{d} N(0, G_j \Sigma_{\boldsymbol{\beta}} G_j') \quad j = 1, 2, \ldots$$

where $\Sigma_{\boldsymbol{\beta}} = \Gamma_p^{-1} \otimes \Sigma$ and

$$G_j = \frac{\partial \mathrm{vec}(\Psi_j)}{\partial \boldsymbol{\beta}'} = \sum_{i=0}^{j-1} \mathbf{J}(\boldsymbol{\Phi}')^{j-1-i} \otimes \Psi_i$$

where $\mathbf{J} = [I_k, 0, \ldots, 0]$ is a $k \times kp$ matrix and $\boldsymbol{\Phi}$ is a $kp \times kp$ companion matrix.

The asymptotic distribution of the accumulated impulse response function is

$$\sqrt{T}\mathrm{vec}(\hat{\Psi}_l^a - \Psi_l^a) \xrightarrow{d} N(0, F_l \Sigma_{\boldsymbol{\beta}} F_l') \quad l = 1, 2, \ldots$$

where $F_l = \sum_{j=1}^{l} G_j$.

The asymptotic distribution of the orthogonalized impulse response function is

$$\sqrt{T}\mathrm{vec}(\hat{\Psi}_j^o - \Psi_j^o) \xrightarrow{d} N(0, C_j \Sigma_{\boldsymbol{\beta}} C_j' + \bar{C}_j \Sigma_{\boldsymbol{\sigma}} \bar{C}_j') \quad j = 0, 1, 2, \ldots$$

where $C_0 = 0$, $C_j = (\Psi_0^{o\prime} \otimes I_k) G_j$, $\bar{C}_j = (I_k \otimes \Psi_j) H$,

$$H = \frac{\partial \mathrm{vec}(\Psi_0^o)}{\partial \boldsymbol{\sigma}'} = L_k' \{L_k (I_{k^2} + K_k)(\Psi_0^o \otimes I_k) L_k'\}^{-1}$$

and $\Sigma_{\boldsymbol{\sigma}} = 2D_k^+ (\Sigma \otimes \Sigma) D_k^{+\prime}$ with $D_k^+ = (D_k' D_k)^{-1} D_k'$ and $\boldsymbol{\sigma} = \mathrm{vech}(\Sigma)$.

Granger Causality Test

Let $\mathbf{y}_t$ be arranged and partitioned in subgroups $\mathbf{y}_{1t}$ and $\mathbf{y}_{2t}$ with dimensions k_1 and k_2, respectively ($k = k_1 + k_2$); that is, $\mathbf{y}_t = (\mathbf{y}_{1t}', \mathbf{y}_{2t}')'$ with the corresponding white noise process $\boldsymbol{\epsilon}_t = (\boldsymbol{\epsilon}_{1t}', \boldsymbol{\epsilon}_{2t}')'$. Consider the VAR($p$) model with partitioned coefficients $\Phi_{ij}(B)$ for $i, j = 1, 2$ as follows:

$$\begin{bmatrix} \Phi_{11}(B) & \Phi_{12}(B) \\ \Phi_{21}(B) & \Phi_{22}(B) \end{bmatrix} \begin{bmatrix} \mathbf{y}_{1t} \\ \mathbf{y}_{2t} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\delta}_1 \\ \boldsymbol{\delta}_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{\epsilon}_{1t} \\ \boldsymbol{\epsilon}_{2t} \end{bmatrix}$$

The variables $\mathbf{y}_{1t}$ are said to cause $\mathbf{y}_{2t}$, but $\mathbf{y}_{2t}$ do not cause $\mathbf{y}_{1t}$ if $\Phi_{12}(B) = 0$. The implication of this model structure is that future values of the process $\mathbf{y}_{1t}$ are influenced only by its own past and not by the past of $\mathbf{y}_{2t}$, where future values of $\mathbf{y}_{2t}$ are influenced by the past of both $\mathbf{y}_{1t}$ and $\mathbf{y}_{2t}$. If the future $\mathbf{y}_{1t}$ are not influenced by the past values of $\mathbf{y}_{2t}$, then it can be better to model $\mathbf{y}_{1t}$ separately from $\mathbf{y}_{2t}$.

Consider testing $H_0: C\boldsymbol{\beta} = c$, where C is a $s \times (k^2 p + k)$ matrix of rank s and c is an s-dimensional vector where $s = k_1 k_2 p$. Assuming that

$$\sqrt{T}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \xrightarrow{d} N(0, \Gamma_p^{-1} \otimes \Sigma)$$

you get the Wald statistic

$$T(C\hat{\boldsymbol{\beta}} - c)'[C(\hat{\Gamma}_p^{-1} \otimes \hat{\Sigma})C']^{-1}(C\hat{\boldsymbol{\beta}} - c) \xrightarrow{d} \chi^2(s)$$

For the Granger causality test, the matrix C consists of zeros or ones and c is the zero vector. For more information about the Granger causality test, see Lütkepohl (1993).

VARX Modeling

The vector autoregressive model with exogenous variables is called the VARX(p,s) model. The form of the VARX(p,s) model can be written as

$$\mathbf{y}_t = \boldsymbol{\delta} + \sum_{i=1}^{p} \Phi_i \mathbf{y}_{t-i} + \sum_{i=0}^{s} \Theta_i^* \mathbf{x}_{t-i} + \boldsymbol{\epsilon}_t$$

The parameter estimates can be obtained by representing the general form of the multivariate linear model,

$$Y = XB + E \quad \text{or} \quad \mathbf{y} = (X \otimes I_k)\boldsymbol{\beta} + \mathbf{e}$$

where

$$\begin{aligned}
Y &= (\mathbf{y}_1, \ldots, \mathbf{y}_T)' \\
B &= (\boldsymbol{\delta}, \Phi_1, \ldots, \Phi_p, \Theta_0^*, \ldots, \Theta_s^*)' \\
X &= (X_0, \ldots, X_{T-1})' \\
X_t &= (1, \mathbf{y}_t', \ldots, \mathbf{y}_{t-p+1}', \mathbf{x}_{t+1}', \ldots, \mathbf{x}_{t-s+1}')' \\
E &= (\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_T)' \\
\mathbf{y} &= \text{vec}(Y') \\
\boldsymbol{\beta} &= \text{vec}(B') \\
\mathbf{e} &= \text{vec}(E')
\end{aligned}$$

The conditional least squares estimator of β can be obtained by using the same method in a VAR(p) modeling. If the multivariate linear model has different independent variables that correspond to dependent variables, the SUR (seemingly unrelated regression) method is used to improve the regression estimates.

The following example fits the ordinary regression model:

```
proc varmax data=one;
   model y1-y3 = x1-x5;
run;
```

This is equivalent to the REG procedure in the SAS/STAT software:

```
proc reg data=one;
   model y1 = x1-x5;
   model y2 = x1-x5;
   model y3 = x1-x5;
run;
```

The following example fits the second-order lagged regression model:

```
proc varmax data=two;
   model y1 y2 = x / xlag=2;
run;
```

This is equivalent to the REG procedure in the SAS/STAT software:

```
data three;
   set two;
   xlag1 = lag1(x);
   xlag2 = lag2(x);
run;

proc reg data=three;
   model y1 = x xlag1 xlag2;
   model y2 = x xlag1 xlag2;
run;
```

The following example fits the ordinary regression model with different regressors:

```
proc varmax data=one;
   model y1 = x1-x3, y2 = x2 x3;
run;
```

This is equivalent to the following SYSLIN procedure statements:

```
proc syslin data=one vardef=df sur;
   endogenous y1 y2;
   model y1 = x1-x3;
   model y2 = x2 x3;
run;
```

From the output in Figure 42.25 in the section "Getting Started: VARMAX Procedure" on page 2954, you can see that the parameters, XL0_1_2, XL0_2_1, XL0_3_1, and XL0_3_2 associated with the exogenous variables, are not significant. The following example fits the VARX(1,0) model with different regressors:

```
proc varmax data=grunfeld;
   model y1 = x1, y2 = x2, y3 / p=1 print=(estimates);
run;
```

Figure 42.64 Parameter Estimates for the VARX(1, 0) Model

The VARMAX Procedure

		XLag	
Lag	Variable	x1	x2
0	y1	1.83231	_
	y2	_	2.42110
	y3	_	_

As you can see in Figure 42.64, the symbol '_' in the elements of matrix corresponds to endogenous variables that do not take the denoted exogenous variables.

Seasonal Dummies and Time Trends

You can use the NSEASON= option to introduce seasonal dummies into the model, and the TREND= option to introduce linear trend or both linear and quadratic trends into the model. The definition of the seasonal dummies and trends starts from the first observation after skipping the presample and the observations that have missing values. The size of the presample is $\max(p, s)$, where p is the maximum number of lags of AR terms and s is the maximum number of lags of exogenous variables; that is, the presample contains $\{\mathbf{y}_{-l+1}, \mathbf{x}_{-l+1}, \ldots, \mathbf{y}_0, \mathbf{x}_0\}$, where $l = \max(p, s)$.

The following statements fit a bivariate VARX(1, 2) model that has four seasonal periods and both linear and quadratic time trends:

```
data One;
   format date date9.;
   do obs = 1 to 100;
      date=intnx('quarter','01Jan1990'd,obs-1);
      y1 = normal(1); y2 = normal(1); x = normal(1);
      output;
   end;
run;

proc varmax data=One;
   model y1 y2 = x / nseason=4 xlag=2 p=1 trend=quad;
run;
```

In the following statements, the seasonal dummies and time trends are explicitly defined in the data set, together with the lags of dependent and exogenous variables, and then the equivalent model is fit by the REG procedure in SAS/STAT software:

```
data Two;
   set one;
   y1lag1 = lag(y1); y2lag1 = lag(y2);
   xlag1 = lag(x); xlag2 = lag2(x);
   if (obs>2) then do;
      ltrend = obs - 2;
      qtrend = ltrend * ltrend;
      const = 1;
      if (mod(ltrend-2,4)=0) then sd1 = 1;
      else sd1 = 0;
      if (mod(ltrend-3,4)=0) then sd2 = 1;
      else sd2 = 0;
      if (mod(ltrend-4,4)=0) then sd3 = 1;
      else sd3 = 0;
   end;
run;

proc reg data=Two(firstobs=3);
   model y1 = const sd1 sd2 sd3 ltrend qtrend
              x xlag1 xlag2 y1lag1 y2lag1 / noint;
```

```
   model y2 = const sd1 sd2 sd3 ltrend qtrend
              x xlag1 xlag2 y1lag1 y2lag1 / noint;
run;
```

The first 11 observations in data set Two are output in Figure 42.65 to show what the seasonal dummies and linear and quadratic time trends look like.

```
proc print data=Two(obs=11);
   var date const sd1 sd2 sd3 ltrend qtrend;
run;
```

Figure 42.65 The First 11 Observations in Data Set Two

Obs	date	const	sd1	sd2	sd3	ltrend	qtrend
1	01JAN1990	.	.	.	.	.	.
2	01APR1990	.	.	.	.	.	.
3	01JUL1990	1	0	0	0	1	1
4	01OCT1990	1	1	0	0	2	4
5	01JAN1991	1	0	1	0	3	9
6	01APR1991	1	0	0	1	4	16
7	01JUL1991	1	0	0	0	5	25
8	01OCT1991	1	1	0	0	6	36
9	01JAN1992	1	0	1	0	7	49
10	01APR1992	1	0	0	1	8	64
11	01JUL1992	1	0	0	0	9	81

Bayesian VAR and VARX Modeling

Consider the VAR(p) model

$$\mathbf{y}_t = \boldsymbol{\delta} + \Phi_1 \mathbf{y}_{t-1} + \cdots + \Phi_p \mathbf{y}_{t-p} + \boldsymbol{\epsilon}_t$$

or

$$\mathbf{y} = (X \otimes I_k)\boldsymbol{\beta} + \mathbf{e}$$

When the parameter vector $\boldsymbol{\beta}$ has a prior multivariate normal distribution with known mean $\boldsymbol{\beta}^*$ and covariance matrix V_β, the prior density is written as

$$f(\boldsymbol{\beta}) = (\frac{1}{2\pi})^{k^2 p/2} |V_\beta|^{-1/2} \exp[-\frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\beta}^*) V_\beta^{-1} (\boldsymbol{\beta} - \boldsymbol{\beta}^*)]$$

The likelihood function for the Gaussian process becomes

$$\begin{aligned} \ell(\boldsymbol{\beta}|\mathbf{y}) &= (\frac{1}{2\pi})^{kT/2} |I_T \otimes \Sigma|^{-1/2} \times \\ &\quad \exp[-\frac{1}{2}(\mathbf{y} - (X \otimes I_k)\boldsymbol{\beta})'(I_T \otimes \Sigma^{-1})(\mathbf{y} - (X \otimes I_k)\boldsymbol{\beta})] \end{aligned}$$

Therefore, the posterior density is derived as

$$f(\boldsymbol{\beta}|\mathbf{y}) \propto \exp[-\frac{1}{2}(\boldsymbol{\beta}-\bar{\boldsymbol{\beta}})'\bar{\Sigma}_{\beta}^{-1}(\boldsymbol{\beta}-\bar{\boldsymbol{\beta}})]$$

where the posterior mean is

$$\bar{\boldsymbol{\beta}} = [V_{\beta}^{-1} + (X'X \otimes \Sigma^{-1})]^{-1}[V_{\beta}^{-1}\boldsymbol{\beta}^{*} + (X' \otimes \Sigma^{-1})\mathbf{y}]$$

and the posterior covariance matrix is

$$\bar{\Sigma}_{\beta} = [V_{\beta}^{-1} + (X'X \otimes \Sigma^{-1})]^{-1}$$

In practice, the prior mean $\boldsymbol{\beta}^{*}$ and the prior variance V_{β} need to be specified. If all the parameters are considered to shrink toward zero, the null prior mean should be specified. According to Litterman (1986), the prior variance can be given by

$$v_{ij}(l) = \begin{cases} (\lambda/l)^2 & \text{if } i = j \\ (\lambda\theta\sigma_{ii}/l\sigma_{jj})^2 & \text{if } i \neq j \end{cases}$$

where $v_{ij}(l)$ is the prior variance of the (i, j) element of Φ_l, λ is the prior standard deviation of the diagonal elements of Φ_l, θ is a constant in the interval $(0, 1)$, and σ_{ii}^2 is the ith diagonal element of Σ. The deterministic terms have diffused prior variance. In practice, you replace the σ_{ii}^2 by the diagonal element of the ML estimator of Σ in the nonconstrained model.

For example, for a bivariate BVAR(2) model,

$$\begin{aligned} y_{1t} &= 0 + \phi_{1,11}y_{1,t-1} + \phi_{1,12}y_{2,t-1} + \phi_{2,11}y_{1,t-2} + \phi_{2,12}y_{2,t-2} + \epsilon_{1t} \\ y_{2t} &= 0 + \phi_{1,21}y_{1,t-1} + \phi_{1,22}y_{2,t-1} + \phi_{2,21}y_{1,t-2} + \phi_{2,22}y_{2,t-2} + \epsilon_{2t} \end{aligned}$$

with the prior covariance matrix

$$\begin{aligned} V_{\beta} = \text{Diag} \quad (\quad & \infty, \lambda^2, (\lambda\theta\sigma_1/\sigma_2)^2, (\lambda/2)^2, (\lambda\theta\sigma_1/2\sigma_2)^2, \\ & \infty, (\lambda\theta\sigma_2/\sigma_1)^2, \lambda^2, (\lambda\theta\sigma_2/2\sigma_1)^2, (\lambda/2)^2 \quad) \end{aligned}$$

For the Bayesian estimation of integrated systems, the prior mean is set to the first lag of each variable equal to one in its own equation and all other coefficients at zero. For example, for a bivariate BVAR(2) model,

$$\begin{aligned} y_{1t} &= 0 + 1\ y_{1,t-1} + 0\ y_{2,t-1} + 0\ y_{1,t-2} + 0\ y_{2,t-2} + \epsilon_{1t} \\ y_{2t} &= 0 + 0\ y_{1,t-1} + 1\ y_{2,t-1} + 0\ y_{1,t-2} + 0\ y_{2,t-2} + \epsilon_{2t} \end{aligned}$$

Forecasting of BVAR Modeling

The mean squared error (MSE) is used to measure forecast accuracy (Litterman 1986). The MSE of the s-step-ahead forecast is

$$\mathrm{MSE}_s = \frac{1}{J-s+1} \sum_{j=1}^{J-s+1} (A_{t_j} - F_{t_j}^s)^2$$

where J is the number specified by NREP= option, t_j is the time index of the observation to be forecasted in repetition j, A_{t_j} is the actual value at time t_j, and $F_{t_j}^s$ is the forecast made s periods earlier. If there are not enough observations, some MSEs might not be calculated.

Bayesian VARX Modeling

The Bayesian vector autoregressive model with exogenous variables is called the BVARX(p,s) model. The form of the BVARX(p,s) model can be written as

$$\mathbf{y}_t = \boldsymbol{\delta} + \sum_{i=1}^{p} \Phi_i \mathbf{y}_{t-i} + \sum_{i=0}^{s} \Theta_i^* \mathbf{x}_{t-i} + \boldsymbol{\epsilon}_t$$

The parameter estimates can be obtained by representing the general form of the multivariate linear model,

$$\mathbf{y} = (X \otimes I_k)\boldsymbol{\beta} + \mathbf{e}$$

The prior means for the AR coefficients are the same as those specified in BVAR(p). The prior means for the exogenous coefficients are set to zero.

Some examples of the Bayesian VARX model are as follows:

```
model y1 y2 = x1 / p=1 xlag=1 prior;

model y1 y2 = x1 / p=(1 3) xlag=1 nocurrentx
                   prior=(lambda=0.9 theta=0.1);
```

VARMA and VARMAX Modeling

A zero-mean VARMA(p, q) process is written as

$$\mathbf{y}_t = \sum_{i=1}^{p} \Phi_i \mathbf{y}_{t-i} + \boldsymbol{\epsilon}_t - \sum_{i=1}^{q} \Theta_i \boldsymbol{\epsilon}_{t-i}$$

or

$$\Phi(B)\mathbf{y}_t = \Theta(B)\boldsymbol{\epsilon}_t$$

where $\Phi(B) = I_k - \sum_{i=1}^{p} \Phi_i B^i$ and $\Theta(B) = I_k - \sum_{i=1}^{q} \Theta_i B^i$.

Stationarity and Invertibility

For stationarity and invertibility of the VARMA process, the roots of $|\Phi(z)| = 0$ and $|\Theta(z)| = 0$ are outside the unit circle.

Parameter Estimation

Under the assumption of normality of the $\boldsymbol{\epsilon}_t$ with zero-mean vector and nonsingular covariance matrix Σ, the conditional (approximate) log-likelihood function of a zero-mean VARMA(p,q) model is considered.

Define $Y = (\mathbf{y}_1, \ldots, \mathbf{y}_T)'$ and $E = (\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_T)'$ with $B^i Y = (\mathbf{y}_{1-i}, \ldots, \mathbf{y}_{T-i})'$ and $B^i E = (\boldsymbol{\epsilon}_{1-i}, \ldots, \boldsymbol{\epsilon}_{T-i})'$; define $\mathbf{y} = \mathrm{vec}(Y')$ and $\mathbf{e} = \mathrm{vec}(E')$. Then

$$\mathbf{y} - \sum_{i=1}^{p} (I_T \otimes \Phi_i) B^i \mathbf{y} = \mathbf{e} - \sum_{i=1}^{q} (I_T \otimes \Theta_i) B^i \mathbf{e}$$

where $B^i \mathbf{y} = \mathrm{vec}[(B^i Y)']$ and $B^i \mathbf{e} = \mathrm{vec}[(B^i E)']$.

Then, the conditional (approximate) log-likelihood function can be written as (Reinsel 1997)

$$\begin{aligned} \ell &= -\frac{T}{2} \log |\Sigma| - \frac{1}{2} \sum_{t=1}^{T} \boldsymbol{\epsilon}_t' \Sigma^{-1} \boldsymbol{\epsilon}_t \\ &= -\frac{T}{2} \log |\Sigma| - \frac{1}{2} \mathbf{w}' \Theta'^{-1} (I_T \otimes \Sigma^{-1}) \Theta^{-1} \mathbf{w} \end{aligned}$$

where $\mathbf{w} = \mathbf{y} - \sum_{i=1}^{p} (I_T \otimes \Phi_i) B^i \mathbf{y}$ and Θ is such that $\mathbf{e} - \sum_{i=1}^{q} (I_T \otimes \Theta_i) B^i \mathbf{e} = \Theta \mathbf{e}$. You can specify METHOD=CML in the MODEL statement to apply conditional maximum likelihood estimation.

For the exact log-likelihood function of a VARMA model, the VARMA model is transformed into the equivalent state space form and then the Kalman filtering method is applied.

The state space form of the zero-mean VARMA(p,q) model consists of a state equation

$$\mathbf{z}_t = F \mathbf{z}_{t-1} + G \boldsymbol{\epsilon}_t$$

and an observation equation

$$\mathbf{y}_t = H \mathbf{z}_t$$

where

$$\mathbf{z}_t = (\mathbf{y}_t', \mathbf{y}_{t-1}', \ldots, \mathbf{y}_{t-(v-1)}', \boldsymbol{\epsilon}_t', \boldsymbol{\epsilon}_{t-1}, \ldots, \boldsymbol{\epsilon}_{t-(q-1)}')'$$

$$F = \begin{bmatrix} \Phi_1 & \cdots & \Phi_{v-1} & \Phi_v & -\Theta_1 & \cdots & -\Theta_{q-1} & -\Theta_q \\ I_k & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & 0 & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & I_k & 0 & 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 & I_k & \cdots & 0 & 0 \\ \vdots & \ddots & 0 & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & I_k & 0 \end{bmatrix}, \quad G = \begin{bmatrix} I_k \\ 0_{k(v-1) \times k} \\ I_k \\ 0_{k(q-1) \times k} \end{bmatrix}$$

and

$$H = [I_k, 0_{k(v+q-1)\times k}]$$

where $v = \max(p, 1)$ and $\Phi_i = 0$ for $i > p$.

The Kalman filtering approach is used to evaluate the likelihood function. The updating equation is

$$\hat{\mathbf{z}}_{t|t} = \hat{\mathbf{z}}_{t|t-1} + K_t \boldsymbol{\epsilon}_{t|t-1}$$

where

$$K_t = P_{t|t-1} H' [H P_{t|t-1} H']^{-1}$$

The prediction equation is

$$\hat{\mathbf{z}}_{t|t-1} = F \hat{\mathbf{z}}_{t-1|t-1}, \quad P_{t|t-1} = F P_{t-1|t-1} F' + G \Sigma G'$$

where $P_{t|t} = [I - K_t H] P_{t|t-1}$ for $t = 1, 2, \ldots, n$.

The log-likelihood function can be expressed as

$$\ell = -\frac{1}{2} \sum_{t=1}^{T} [\log |\Sigma_{t|t-1}| + (\mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1})' \Sigma_{t|t-1}^{-1} (\mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1})]$$

where $\hat{\mathbf{y}}_{t|t-1}$ and $\Sigma_{t|t-1}$ are determined recursively from the Kalman filtering method. To construct the likelihood function from Kalman filtering, you obtain $\hat{\mathbf{y}}_{t|t-1} = H \hat{\mathbf{z}}_{t|t-1}$, $\hat{\boldsymbol{\epsilon}}_{t|t-1} = \mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1}$, and $\Sigma_{t|t-1} = H P_{t|t-1} H'$.

When you specify METHOD=ML in the MODEL statement, the exact log likelihood is evaluated and used in the maximum likelihood estimation.

Define the vector $\boldsymbol{\beta}$ as

$$\boldsymbol{\beta} = (\phi_1', \ldots, \phi_p', \theta_1', \ldots, \theta_q', \mathrm{vech}(\Sigma))'$$

where $\phi_i = \mathrm{vec}(\Phi_i)$ and $\theta_i = \mathrm{vec}(\Theta_i)$. All elements of $\boldsymbol{\beta}$ are estimated through the preceding (conditional) maximum likelihood method. The estimates of $\Phi_i, i = 1, \ldots, p$, and $\Theta_i, i = 1, \ldots, q$, are output in the ParameterEstimates ODS table. The estimates of the covariance matrix (Σ) are output in the CovarianceParameterEstimates ODS table. If you specify the OUTEST=, OUTCOV, PRINT=(COVB), or PRINT=(CORRB) option, you can see all elements of $\boldsymbol{\beta}$, including the covariance matrix Σ, in the parameter estimates, covariance of parameter estimates, or correlation of parameter estimates. You can also apply the BOUND, INITIAL, RESTRICT, and TEST statements to any elements of $\boldsymbol{\beta}$, including the covariance matrix Σ. For more information, see the syntax of the corresponding statement.

The (conditional) log-likelihood equations are solved by iterative numerical methods such as quasi-Newton optimization. The starting values for the AR and MA parameters are obtained from the least squares estimates. Although the small-sample properties of CML estimates might not be as good as the ML estimates, the CML method is much faster than the ML method. Depending on the sample size and number of parameters to be estimated, the CML method can be hundreds or even thousands of times faster than the ML method. In the following example code, the CML method is about 100 times faster than the ML method, with very similar estimation and forecast results:

```
proc iml;
   phi = (0.9 * I(4)) // (-0.7* I(4));
   theta = 0.8 * I(4);
   sig = I(4);
   /* to simulate the vector time series */
   call varmasim(y,phi,theta) sigma=sig n=400 seed=2;

   cn = {'y1' 'y2' 'y3' 'y4'};
   create simul6 from y[colname=cn];
   append from y;
   close;
quit;

proc varmax data=simul6;
   model y1 y2 y3 y4 / noint p=2 q=1 method=cml;
   nloptions pall maxit=5000 tech=qn;
   output out=ocml back=12 lead=24;
run;

proc varmax data=simul6;
   model y1 y2 y3 y4 / noint p=2 q=1 method=ml;
   nloptions pall maxit=5000 tech=qn;
   output out=oml back=12 lead=24;
run;
```

Asymptotic Distribution of the Parameter Estimates

Under the assumptions of stationarity and invertibility for the VARMA model and the assumption that $\boldsymbol{\epsilon}_t$ is a white noise process, $\hat{\boldsymbol{\beta}}$ is a consistent estimator for $\boldsymbol{\beta}$ and $\sqrt{T}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})$ converges in distribution to the multivariate normal $N(0, V^{-1})$ as $T \to \infty$, where V is the asymptotic information matrix of $\boldsymbol{\beta}$.

Asymptotic Distributions of Impulse Response Functions

Defining the vector $\boldsymbol{\beta}$

$$\boldsymbol{\beta} = (\phi_1', \ldots, \phi_p', \theta_1', \ldots, \theta_q')'$$

the asymptotic distribution of the impulse response function for a VARMA(p, q) model is

$$\sqrt{T}\,\mathrm{vec}(\hat{\Psi}_j - \Psi_j) \xrightarrow{d} N(0, G_j \Sigma_{\boldsymbol{\beta}} G_j') \quad j = 1, 2, \ldots$$

where $\Sigma_{\boldsymbol{\beta}}$ is the covariance matrix of the parameter estimates and

$$G_j = \frac{\partial \mathrm{vec}(\Psi_j)}{\partial \boldsymbol{\beta}'} = \sum_{i=0}^{j-1} \mathbf{H}'(\mathbf{A}')^{j-1-i} \otimes \mathbf{J}\mathbf{A}^i\mathbf{J}'$$

where $\mathbf{H} = [I_k, 0, \ldots, 0, I_k, 0, \ldots, 0]'$ is a $k(p+q) \times k$ matrix with the second I_k following after p block matrices; $\mathbf{J} = [I_k, 0, \ldots, 0]$ is a $k \times k(p+q)$ matrix; $\mathbf{A}$ is a $k(p+q) \times k(p+q)$ matrix,

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

where

$$A_{11} = \begin{bmatrix} \Phi_1 & \Phi_2 & \cdots & \Phi_{p-1} & \Phi_p \\ I_k & 0 & \cdots & 0 & 0 \\ 0 & I_k & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I_k & 0 \end{bmatrix} \quad A_{12} = \begin{bmatrix} -\Theta_1 & \cdots & -\Theta_{q-1} & -\Theta_q \\ 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{bmatrix}$$

A_{21} is a $kq \times kp$ zero matrix, and

$$A_{22} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ I_k & 0 & \cdots & 0 & 0 \\ 0 & I_k & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I_k & 0 \end{bmatrix}$$

An Example of a VARMA(1,1) Model

Consider a VARMA(1,1) model with mean zero,

$$\mathbf{y}_t = \Phi_1 \mathbf{y}_{t-1} + \boldsymbol{\epsilon}_t - \Theta_1 \boldsymbol{\epsilon}_{t-1}$$

where $\boldsymbol{\epsilon}_t$ is the white noise process with a mean zero vector and the positive-definite covariance matrix Σ.

The following IML procedure statements simulate a bivariate vector time series from this model to provide test data for the VARMAX procedure:

```
proc iml;
   sig = {1.0  0.5, 0.5 1.25};
   phi = {1.2 -0.5, 0.6 0.3};
   theta = {0.5 -0.2, 0.1 0.3};
   /* to simulate the vector time series */
   call varmasim(y,phi,theta) sigma=sig n=100 seed=34657;
   cn = {'y1' 'y2'};
   create simul3 from y[colname=cn];
   append from y;
run;
```

The following statements fit a VARMA(1,1) model to the simulated data. You specify the order of the autoregressive model by using the P= option and specify the order of moving-average model by using the Q= option. You specify the quasi-Newton optimization in the NLOPTIONS statement as an optimization method.

```
proc varmax data=simul3;
   nloptions tech=qn;
   model y1 y2 / p=1 q=1 noint print=(estimates);
run;
```

Figure 42.66 shows the initial values of parameters. The initial values were estimated by using the least squares method.

Figure 42.66 Start Parameter Estimates for the VARMA(1, 1) Model

The VARMAX Procedure

Optimization Start Parameter Estimates			
N	Parameter	Estimate	Gradient Objective Function
1	AR1_1_1	0.964299	-2.357098
2	AR1_2_1	0.481620	-3.773499
3	AR1_1_2	-0.363819	1.865051
4	AR1_2_2	0.457378	-10.778568
5	MA1_1_1	0.244355	-2.552198
6	MA1_2_1	-0.034093	2.716227
7	MA1_1_2	-0.006261	-0.147004
8	MA1_2_2	0.444636	0.141839
9	COV1_1	1.353584	2.765550
10	COV1_2	0.415649	-1.389416
11	COV2_2	1.445260	2.581735

Figure 42.67 shows the default option settings for the quasi-Newton optimization technique.

Figure 42.67 Default Criteria for the quasi-Newton Optimization

Minimum Iterations	0
Maximum Iterations	200
Maximum Function Calls	2000
ABSGCONV Gradient Criterion	0.00001
GCONV Gradient Criterion	1E-8
ABSFCONV Function Criterion	0
FCONV Function Criterion	2.220446E-16
FCONV2 Function Criterion	0
FSIZE Parameter	0
ABSXCONV Parameter Change Criterion	0
XCONV Parameter Change Criterion	0
XSIZE Parameter	0
ABSCONV Function Criterion	-1.34078E154
Line Search Method	2
Starting Alpha for Line Search	1
Line Search Precision LSPRECISION	0.4
DAMPSTEP Parameter for Line Search	.
Singularity Tolerance (SINGULAR)	1E-8

Figure 42.68 shows the iteration history of parameter estimates.

Figure 42.68 Iteration History of Parameter Estimates

Iteration	Restarts	Function Calls	Active Constraints	Objective Function	Objective Function Change	Max Abs Gradient Element	Step Size	Slope of Search Direction
1	0	3	0	121.22330	0.1526	5.2001	0.00384	-78.688
2	0	5	0	120.97740	0.2459	6.2584	3.214	-0.156
3	0	6	0	120.58286	0.3945	4.1004	0.948	-0.648
4	0	7	0	120.43152	0.1513	3.7834	1.000	-0.346
5	0	8	0	120.32992	0.1016	6.3797	1.000	-0.243
6	0	10	0	120.26832	0.0616	3.1048	0.407	-0.304
7	0	12	0	120.23311	0.0352	1.0747	0.983	-0.0731
8	0	14	0	120.22264	0.0105	0.6370	1.518	-0.0127
9	0	15	0	120.21560	0.00704	1.3563	4.650	-0.0056
10	0	16	0	120.21281	0.00279	1.2963	2.102	-0.0084
11	0	17	0	120.20951	0.00330	0.1634	1.139	-0.0061
12	0	19	0	120.20896	0.000542	0.1349	2.591	-0.0004
13	0	21	0	120.20884	0.000123	0.0662	1.883	-0.0001
14	0	22	0	120.20875	0.000093	0.1399	4.120	-0.0001
15	0	24	0	120.20871	0.000037	0.00917	1.073	-0.0001
16	0	26	0	120.20871	1.643E-6	0.00858	2.115	-155E-8
17	0	27	0	120.20871	7.704E-7	0.00543	5.409	-759E-9

Figure 42.69 shows the final parameter estimates.

Figure 42.69 Results of Parameter Estimates for the VARMA(1, 1) Model

The VARMAX Procedure

Optimization Results

Parameter Estimates

N	Parameter	Estimate	Gradient Objective Function
1	**AR1_1_1**	1.020117	0.003641
2	**AR1_2_1**	0.393557	0.000140
3	**AR1_1_2**	-0.388708	0.001311
4	**AR1_2_2**	0.551644	0.002479
5	**MA1_1_1**	0.330598	0.000131
6	**MA1_2_1**	-0.166999	0.000086321
7	**MA1_1_2**	-0.032507	-0.001133
8	**MA1_2_2**	0.587232	-0.000523
9	**COV1_1**	1.253624	0.005429
10	**COV1_2**	0.382094	-0.001152
11	**COV2_2**	1.322424	-0.000535

Figure 42.70 shows the AR coefficient matrix in terms of lag 1, the MA coefficient matrix in terms of lag 1, the parameter estimates, and their significance, which is one indication of how well the model fits the data.

Figure 42.70 Parameter Estimates for the VARMA(1, 1) Model

The VARMAX Procedure

Type of Model	VARMA(1,1)
Estimation Method	Maximum Likelihood Estimation

		AR	
Lag	**Variable**	**y1**	**y2**
1	**y1**	1.02012	-0.38871
	y2	0.39356	0.55164

		MA	
Lag	**Variable**	**e1**	**e2**
1	**y1**	0.33060	-0.03251
	y2	-0.16700	0.58723

Schematic Representation		
Variable/Lag	**AR1**	**MA1**
y1	+-	+.
y2	++	.+

+ is > 2*std error, - is < -2*std error, . is between, * is N/A

Model Parameter Estimates

Equation	Parameter	Estimate	Standard Error	t Value	Pr > |t|	Variable
y1	**AR1_1_1**	1.02012	0.10076	10.12	0.0001	y1(t-1)
	AR1_1_2	-0.38871	0.09557	-4.07	0.0001	y2(t-1)
	MA1_1_1	0.33060	0.14389	2.30	0.0237	e1(t-1)
	MA1_1_2	-0.03251	0.14146	-0.23	0.8187	e2(t-1)
y2	**AR1_2_1**	0.39356	0.10210	3.85	0.0002	y1(t-1)
	AR1_2_2	0.55164	0.08536	6.46	0.0001	y2(t-1)
	MA1_2_1	-0.16700	0.15801	-1.06	0.2931	e1(t-1)
	MA1_2_2	0.58723	0.14372	4.09	0.0001	e2(t-1)

Covariance Parameter Estimates

Parameter	Estimate	Standard Error	t Value	Pr > |t|
COV1_1	1.25362	0.17788	7.05	0.0001
COV1_2	0.38209	0.13484	2.83	0.0056
COV2_2	1.32242	0.18829	7.02	0.0001

The fitted VARMA(1,1) model with estimated standard errors in parentheses is given as

$$\mathbf{y}_t = \begin{pmatrix} 1.01846 & -0.38682 \\ (0.10256) & (0.09644) \\ 0.39182 & 0.55281 \\ (0.10062) & (0.08422) \end{pmatrix} \mathbf{y}_{t-1} + \boldsymbol{\epsilon}_t - \begin{pmatrix} 0.32292 & -0.02160 \\ (0.14524) & (0.14203) \\ -0.16501 & 0.58576 \\ (0.15704) & (0.14115) \end{pmatrix} \boldsymbol{\epsilon}_{t-1}$$

and

$$\boldsymbol{\epsilon}_t \sim \text{iid } N(0, \begin{pmatrix} 1.25202 & 0.37950 \\ (0.17697) & (0.13401) \\ 0.37950 & 1.31315 \\ (0.13401) & (0.18610) \end{pmatrix}$$

VARMAX Modeling

A general VARMAX(p, q, s) process is written as

$$\mathbf{y}_t = \boldsymbol{\delta}_t + \sum_{i=1}^{p} \Phi_i \mathbf{y}_{t-i} + \boldsymbol{\epsilon}_t - \sum_{i=1}^{q} \Theta_i \boldsymbol{\epsilon}_{t-i}$$

or

$$\Phi(B)\mathbf{y}_t = \boldsymbol{\delta}_t + \Theta(B)\boldsymbol{\epsilon}_t$$

where $\Phi(B) = I_k - \sum_{i=1}^{p} \Phi_i B^i$ and $\Theta(B) = I_k - \sum_{i=1}^{q} \Theta_i B^i$. The vector $\boldsymbol{\delta}_t$ consists of all possible deterministic terms, namely constant, seasonal dummies, linear trend, quadratic trend, and exogenous variables. The vector $\boldsymbol{\delta}_t = \Delta \mathbf{c}_t$, where $\mathbf{c}_t = (D_t' \ \mathbf{x}_t' \ \ldots \ \mathbf{x}_{t-s}')'$; $D_t = (1 \ d_{t,1} \ \ldots \ d_{t,n_s-1} \ t \ t^2)'$; $d_{t,i}, i = 1, \ldots, n_s - 1$, are seasonal dummies and n_s is based on the NSEASON= option; $\Delta = (A \ \Theta_0^* \ \ldots \Theta_s^*)$; A is the parameter matrix corresponding to D_t and Θ_i^* for $\mathbf{x}_{t-i}, i = 0, \ldots, s$.

The state space form of the VARMAX(p,q,s) model consists of a state equation

$$\mathbf{z}_t = F\mathbf{z}_{t-1} + \mathbf{w}_t + G\boldsymbol{\epsilon}_t$$

and an observation equation

$$\mathbf{y}_t = H\mathbf{z}_t$$

where

$$\mathbf{z}_t = (\mathbf{y}_t', \mathbf{y}_{t-1}', \ldots, \mathbf{y}_{t-(v-1)}', \boldsymbol{\epsilon}_t', \boldsymbol{\epsilon}_{t-1}, \ldots, \boldsymbol{\epsilon}_{t-(q-1)}', \mathbf{c}_{t+1}')'$$

$$F = \begin{bmatrix} \Phi_1 & \cdots & \Phi_{v-1} & \Phi_v & -\Theta_1 & \cdots & -\Theta_{q-1} & -\Theta_q & \Delta \\ I_k & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \ddots & 0 & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & I_k & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \cdots & 0 & 0 & I_k & \cdots & 0 & 0 & 0 \\ \vdots & \ddots & 0 & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & I_k & 0 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} I_k \\ 0_{k(v-1)\times k} \\ I_k \\ 0_{k(q-1)\times k} \\ 0_{u\times k} \end{bmatrix}$$

and

$$H = [I_k, 0_{(k(v+q-1)+u)\times k}]$$

where $v = \max(p, 1)$, $\Phi_i = 0$ for $i > p$, and u is the dimension of c_t.

Kalman filtering is used to evaluate the likelihood function. The updating equation is

$$\hat{\mathbf{z}}_{t|t} = \hat{\mathbf{z}}_{t|t-1} + K_t \boldsymbol{\epsilon}_{t|t-1}$$

where

$$K_t = P_{t|t-1} H' [H P_{t|t-1} H']^{-1}$$

The prediction equation is

$$\hat{\mathbf{z}}_{t|t-1} = F \hat{\mathbf{z}}_{t-1|t-1} + \mathbf{w}_t, \quad P_{t|t-1} = F P_{t-1|t-1} F' + G \Sigma G'$$

where $P_{t|t} = [I - K_t H] P_{t|t-1}$ for $t = 1, 2, \ldots, n$.

The log-likelihood function can be expressed as

$$\ell = -\frac{1}{2} \sum_{t=1}^{T} [\log |\Sigma_{t|t-1}| + (\mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1})' \Sigma_{t|t-1}^{-1} (\mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1})]$$

where $\hat{\mathbf{y}}_{t|t-1}$ and $\Sigma_{t|t-1}$ are determined recursively from Kalman filtering. To construct the likelihood function from Kalman filtering, you obtain $\hat{\mathbf{y}}_{t|t-1} = H \hat{\mathbf{z}}_{t|t-1}$, $\hat{\boldsymbol{\epsilon}}_{t|t-1} = \mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1}$, and $\Sigma_{t|t-1} = H P_{t|t-1} H'$.

In the preceding state space form of a VARMAX model, the exogenous variables are treated as determined terms, which implies that the values of the exogenous variables must be provided to forecast the out-of-sample dependent variables. If you do not have the future values of the exogenous variables, either you predict the exogenous variables in a separate model, or you express both the exogenous variables and the dependent variables in one combined model and predict them together (Reinsel 1997).

The dimension of the state space vector of the Kalman filtering method for the VARMAX(*p*,*q*,*s*) model might be large, so it might take a lot of time and memory for computing.

Two examples of VARMAX modeling follow:

```
model y1 y2 = x1 / q=1;
nloptions tech=qn;

model y1 y2 = x1 / p=1 q=1 xlag=1 nocurrentx;
nloptions tech=qn;
```

Model Diagnostic Checks

Multivariate Model Diagnostic Checks

Log Likelihood

The log-likelihood function for the fitted model is reported in the LogLikelihood ODS table. The log-likelihood functions for different models are defined as follows:

- For VARMAX models that are estimated through the (conditional) maximum likelihood method, see the section "VARMA and VARMAX Modeling" on page 3076.

- For Bayesian VAR and VARX models, see the section "Bayesian VAR and VARX Modeling" on page 3074.
- For (Bayesian) vector error correction models, see the section "Vector Error Correction Modeling" on page 3090.
- For multivariate GARCH models, see the section "Multivariate GARCH Modeling" on page 3110.
- For VARFIMA and VARFIMAX models, see the section "VARFIMA and VARFIMAX Modeling" on page 3121.
- For VAR and VARX models that are estimated through the least squares (LS) method, the log likelihood is defined as

  $$\ell = -\frac{1}{2}(T \log|\tilde{\Sigma}| + kT)$$

  where $\tilde{\Sigma}$ is the maximum likelihood estimate of the innovation covariance matrix, k is the number of dependent variables, and T is the number of observations used in the estimation.

Information Criteria

The information criteria include Akaike's information criterion (AIC), the corrected Akaike's information criterion (AICC), the final prediction error criterion (FPE), the Hannan-Quinn criterion (HQC), and the Schwarz Bayesian criterion (SBC, also referred to as BIC). These criteria are defined as

$$\begin{aligned}
\text{AIC} &= -2\ell + 2r \\
\text{AICC} &= -2\ell + 2rT/(T-r-1) \\
\text{FPE} &= \left(\frac{T+r_b}{T-r_b}\right)^k |\tilde{\Sigma}| \\
\text{HQC} &= -2\ell + 2r\log(\log(T)) \\
\text{SBC} &= -2\ell + r\log(T)
\end{aligned}$$

where ℓ is the log likelihood, r is the total number of parameters in the model, k is the number of dependent variables, T is the number of observations that are used to estimate the model, r_b is the number of parameters in each mean equation, and $\tilde{\Sigma}$ is the maximum likelihood estimate of Σ. As suggested by Burnham and Anderson (2004) for least squares estimation, the total number of parameters, r, must include the parameters in the innovation covariance matrix. When comparing models, choose the model that has the smallest criterion values.

For an example of the output, see Figure 42.4 earlier in this chapter.

Portmanteau Statistic

The portmanteau statistic, Q_s, is used to test whether correlation remains on the model residuals. The null hypothesis is that the residuals are uncorrelated. Let $C_\epsilon(l)$ be the residual cross-covariance matrices, $\hat{\rho}_\epsilon(l)$ be the residual cross-correlation matrices as

$$C_\epsilon(l) = T^{-1}\sum_{t=1}^{T-l} \boldsymbol{\epsilon}_t \boldsymbol{\epsilon}'_{t+l}$$

and

$$\hat{\rho}_\epsilon(l) = \hat{V}_\epsilon^{-1/2} C_\epsilon(l) \hat{V}_\epsilon^{-1/2} \text{ and } \hat{\rho}_\epsilon(-l) = \hat{\rho}_\epsilon(l)'$$

where $\hat{V}_\epsilon = \text{Diag}(\hat{\sigma}_{11}^2, \ldots, \hat{\sigma}_{kk}^2)$ and $\hat{\sigma}_{ii}^2$ are the diagonal elements of $\hat{\Sigma}$. The multivariate portmanteau test defined in Hosking (1980) is

$$Q_s = T^2 \sum_{l=1}^{s} (T-l)^{-1} \text{tr}\{\hat{\rho}_\epsilon(l) \hat{\rho}_\epsilon(0)^{-1} \hat{\rho}_\epsilon(-l) \hat{\rho}_\epsilon(0)^{-1}\}$$

The statistic Q_s has approximately the chi-square distribution with $k^2(s-p-q)$ degrees of freedom. An example of the output is displayed in Figure 42.7.

Univariate Model Diagnostic Checks

There are various ways to perform diagnostic checks for a univariate model. For more information, see the section "Testing for Nonlinear Dependence: Heteroscedasticity Tests" on page 403 in Chapter 8, "The AUTOREG Procedure." An example of the output is displayed in Figure 42.8 and Figure 42.9.

- Durbin-Watson (DW) statistics: The DW test statistics test for the first order autocorrelation in the residuals.
- Jarque-Bera normality test: This test is helpful in determining whether the model residuals represent a white noise process. This tests the null hypothesis that the residuals have normality.
- *F* tests for autoregressive conditional heteroscedastic (ARCH) disturbances: *F* test statistics test for the heteroscedastic disturbances in the residuals. This tests the null hypothesis that the residuals have equal covariances
- *F* tests for AR disturbance: These test statistics are computed from the residuals of the univariate AR(1), AR(1,2), AR(1,2,3), and AR(1,2,3,4) models to test the null hypothesis that the residuals are uncorrelated.

Cointegration

This section briefly introduces the concepts of cointegration (Johansen 1995a).

Definition 1. (Engle and Granger 1987): If a series y_t with no deterministic components can be represented by a stationary and invertible ARMA process after differencing d times, the series is integrated of order d, that is, $y_t \sim I(d)$.

Definition 2. (Engle and Granger 1987): If all elements of the vector $\mathbf{y}_t$ are $I(d)$ and there exists a cointegrating vector $\boldsymbol{\beta} \neq 0$ such that $\boldsymbol{\beta}'\mathbf{y}_t \sim I(d-b)$ for any $b > 0$, the vector process is said to be cointegrated $CI(d,b)$.

A simple example of a cointegrated process is the following bivariate system:

$$\begin{aligned} y_{1t} &= \gamma y_{2t} + \epsilon_{1t} \\ y_{2t} &= y_{2,t-1} + \epsilon_{2t} \end{aligned}$$

with ϵ_{1t} and ϵ_{2t} being uncorrelated white noise processes. In the second equation, y_{2t} is a random walk, $\Delta y_{2t} = \epsilon_{2t}$, $\Delta \equiv 1 - B$. Differencing the first equation results in

$$\Delta y_{1t} = \gamma \Delta y_{2t} + \Delta \epsilon_{1t} = \gamma \epsilon_{2t} + \epsilon_{1t} - \epsilon_{1,t-1}$$

Thus, both y_{1t} and y_{2t} are $I(1)$ processes, but the linear combination $y_{1t} - \gamma y_{2t}$ is stationary. Hence $\mathbf{y}_t = (y_{1t}, y_{2t})'$ is cointegrated with a cointegrating vector $\boldsymbol{\beta} = (1, -\gamma)'$.

In general, if the vector process $\mathbf{y}_t$ has k components, then there can be more than one cointegrating vector $\boldsymbol{\beta}'$. It is assumed that there are r linearly independent cointegrating vectors with $r < k$, which make the $k \times r$ matrix $\boldsymbol{\beta}$. The rank of matrix $\boldsymbol{\beta}$ is r, which is called the *cointegration rank* of $\mathbf{y}_t$.

Common Trends

This section briefly discusses the implication of cointegration for the moving-average representation. Let $\mathbf{y}_t$ be cointegrated $CI(1, 1)$, then $\Delta \mathbf{y}_t$ has the Wold representation:

$$\Delta \mathbf{y}_t = \boldsymbol{\delta} + \Psi(B)\boldsymbol{\epsilon}_t$$

where $\boldsymbol{\epsilon}_t$ is iid$(0, \Sigma)$, $\Psi(B) = \sum_{j=0}^{\infty} \Psi_j B^j$ with $\Psi_0 = I_k$, and $\sum_{j=0}^{\infty} j|\Psi_j| < \infty$.

Assume that $\boldsymbol{\epsilon}_t = 0$ if $t \leq 0$ and $\mathbf{y}_0$ is a nonrandom initial value. Then the difference equation implies that

$$\mathbf{y}_t = \mathbf{y}_0 + \boldsymbol{\delta} t + \Psi(1) \sum_{i=0}^{t} \boldsymbol{\epsilon}_i + \Psi^*(B)\boldsymbol{\epsilon}_t$$

where $\Psi^*(B) = (1 - B)^{-1}(\Psi(B) - \Psi(1))$ and $\Psi^*(B)$ is absolutely summable.

Assume that the rank of $\Psi(1)$ is $m = k - r$. When the process $\mathbf{y}_t$ is cointegrated, there is a cointegrating $k \times r$ matrix $\boldsymbol{\beta}$ such that $\boldsymbol{\beta}'\mathbf{y}_t$ is stationary.

Premultiplying $\mathbf{y}_t$ by $\boldsymbol{\beta}'$ results in

$$\boldsymbol{\beta}'\mathbf{y}_t = \boldsymbol{\beta}'\mathbf{y}_0 + \boldsymbol{\beta}'\Psi^*(B)\boldsymbol{\epsilon}_t$$

because $\boldsymbol{\beta}'\Psi(1) = 0$ and $\boldsymbol{\beta}'\boldsymbol{\delta} = 0$.

Stock and Watson (1988) showed that the cointegrated process $\mathbf{y}_t$ has a common trends representation derived from the moving-average representation. Since the rank of $\Psi(1)$ is $m = k - r$, there is a $k \times r$ matrix H_1 with rank r such that $\Psi(1)H_1 = 0$. Let H_2 be a $k \times m$ matrix with rank m such that $H_2'H_1 = 0$; then $A = C(1)H_2$ has rank m. The $H = (H_1, H_2)$ has rank k. By construction of $\mathbf{H}$,

$$\Psi(1)H = [0, A] = AS_m$$

where $S_m = (0_{m\times r}, I_m)$. Since $\boldsymbol{\beta}'\Psi(1) = 0$ and $\boldsymbol{\beta}'\boldsymbol{\delta} = 0$, $\boldsymbol{\delta}$ lies in the column space of $\Psi(1)$ and can be written

$$\boldsymbol{\delta} = \Psi(1)\tilde{\boldsymbol{\delta}}$$

where $\tilde{\boldsymbol{\delta}}$ is a k-dimensional vector. The common trends representation is written as

$$\begin{aligned}
\mathbf{y}_t &= \mathbf{y}_0 + \Psi(1)[\tilde{\boldsymbol{\delta}}t + \sum_{i=0}^{t}\boldsymbol{\epsilon}_i] + \Psi^*(B)\boldsymbol{\epsilon}_t \\
&= \mathbf{y}_0 + \Psi(1)H[H^{-1}\tilde{\delta}t + H^{-1}\sum_{i=0}^{t}\boldsymbol{\epsilon}_i] + \mathbf{a}_t \\
&= \mathbf{y}_0 + A\boldsymbol{\tau}_t + \mathbf{a}_t
\end{aligned}$$

and

$$\boldsymbol{\tau}_t = \pi + \boldsymbol{\tau}_{t-1} + \mathbf{v}_t$$

where $\mathbf{a}_t = \Psi^*(B)\boldsymbol{\epsilon}_t$, $\pi = S_m H^{-1}\tilde{\boldsymbol{\delta}}$, $\boldsymbol{\tau}_t = S_m[H^{-1}\tilde{\boldsymbol{\delta}}t + H^{-1}\sum_{i=0}^{t}\boldsymbol{\epsilon}_i]$, and $\mathbf{v}_t = S_m H^{-1}\boldsymbol{\epsilon}_t$.

Stock and Watson showed that the common trends representation expresses $\mathbf{y}_t$ as a linear combination of m random walks ($\boldsymbol{\tau}_t$) with drift π plus $I(0)$ components ($\mathbf{a}_t$).

Test for the Common Trends

Stock and Watson (1988) proposed statistics for common trends testing. The null hypothesis is that the k-dimensional time series $\mathbf{y}_t$ has m common stochastic trends, where $m \leq k$ and the alternative is that it has s common trends, where $s < m$. The test procedure of m versus s common stochastic trends is performed based on the first-order serial correlation matrix of $\mathbf{y}_t$. Let $\boldsymbol{\beta}_\perp$ be a $k \times m$ matrix orthogonal to the cointegrating matrix such that $\boldsymbol{\beta}_\perp'\boldsymbol{\beta} = 0$ and $\boldsymbol{\beta}_\perp\boldsymbol{\beta}_\perp' = I_m$. Let $\mathbf{z}_t = \boldsymbol{\beta}'\mathbf{y}_t$ and $\mathbf{w}_t = \boldsymbol{\beta}_\perp'\mathbf{y}_t$. Then

$$\mathbf{w}_t = \boldsymbol{\beta}_\perp'\mathbf{y}_0 + \boldsymbol{\beta}_\perp'\boldsymbol{\delta}t + \boldsymbol{\beta}_\perp'\Psi(1)\sum_{i=0}^{t}\boldsymbol{\epsilon}_i + \boldsymbol{\beta}_\perp'\Psi^*(B)\boldsymbol{\epsilon}_t$$

Combining the expression of $\mathbf{z}_t$ and $\mathbf{w}_t$,

$$\begin{aligned}
\begin{bmatrix} \mathbf{z}_t \\ \mathbf{w}_t \end{bmatrix} &= \begin{bmatrix} \boldsymbol{\beta}'\mathbf{y}_0 \\ \boldsymbol{\beta}_\perp'\mathbf{y}_0 \end{bmatrix} + \begin{bmatrix} 0 \\ \boldsymbol{\beta}_\perp'\boldsymbol{\delta} \end{bmatrix} t + \begin{bmatrix} 0 \\ \boldsymbol{\beta}_\perp'\Psi(1) \end{bmatrix} \sum_{i=1}^{t}\boldsymbol{\epsilon}_i \\
&+ \begin{bmatrix} \boldsymbol{\beta}'\Psi^*(B) \\ \boldsymbol{\beta}_\perp'\Psi^*(B) \end{bmatrix} \boldsymbol{\epsilon}_t
\end{aligned}$$

The Stock-Watson common trends test is performed based on the component $\mathbf{w}_t$ by testing whether $\boldsymbol{\beta}_\perp'\Psi(1)$ has rank m against rank s.

The following statements perform the Stock-Watson test for common trends:

```
proc iml;
   sig = 100*i(2);
   phi = {-0.2 0.1, 0.5 0.2, 0.8 0.7, -0.4 0.6};
   call varmasim(y,phi) sigma=sig n=100 initial=0
                        seed=45876;
   cn = {'y1' 'y2'};
   create simul2 from y[colname=cn];
   append from y;
quit;

data simul2;
   set simul2;
   date = intnx( 'year', '01jan1900'd, _n_-1 );
   format date year4. ;
run;

proc varmax data=simul2;
   model y1 y2 / p=2 cointtest=(sw);
run;
```

In Figure 42.71, the first column is the null hypothesis that $\mathbf{y}_t$ has $m \leq k$ common trends; the second column is the alternative hypothesis that $\mathbf{y}_t$ has $s < m$ common trends; the third column contains the eigenvalues used for the test statistics; the fourth column contains the test statistics using AR(p) filtering of the data. The table shows the output of the case $p = 2$.

Figure 42.71 Common Trends Test (COINTTEST=(SW) Option)

The VARMAX Procedure

Common Trend Test

H0: Rank=m	H1: Rank=s	Eigenvalue	Filter	5% Critical Value	Lag
1	**0**	1.000906	0.09	-14.10	2
2	**0**	0.996763	-0.32	-8.80	
	1	0.648908	-35.11	-23.00	

The test statistic for testing for 2 versus 1 common trends is more negative (–35.1) than the critical value (–23.0). Therefore, the test rejects the null hypothesis, which means that the series has a single common trend.

Vector Error Correction Modeling

This section discusses the implication of cointegration for the autoregressive representation.

Consider the vector autoregressive process that has Gaussian errors defined by

$$\mathbf{y}_t = \sum_{i=1}^{p} \Phi_i \mathbf{y}_{t-i} + \boldsymbol{\epsilon}_t$$

or

$$\Phi(B)\mathbf{y}_t = \boldsymbol{\epsilon}_t$$

where the initial values, $\mathbf{y}_{-p+1}, \ldots, \mathbf{y}_0$, are fixed and $\boldsymbol{\epsilon}_t \sim N(0, \Sigma)$. The AR operator $\Phi(B)$ can be re-expressed as

$$\Phi(B) = \Phi^*(B)(1 - B) + \Phi(1)B$$

where

$$\Phi(1) = I_k - \Phi_1 - \Phi_2 - \cdots - \Phi_p, \Phi^*(B) = I_k - \sum_{i=1}^{p-1} \Phi_i^* B^i, \Phi_i^* = -\sum_{j=i+1}^{p} \Phi_j$$

The vector error correction model (VECM), also called the vector equilibrium correction model, is defined as

$$\Phi^*(B)(1 - B)\mathbf{y}_t = \boldsymbol{\alpha}\boldsymbol{\beta}'\mathbf{y}_{t-1} + \boldsymbol{\epsilon}_t$$

or

$$\Delta\mathbf{y}_t = \boldsymbol{\alpha}\boldsymbol{\beta}'\mathbf{y}_{t-1} + \sum_{i=1}^{p-1} \Phi_i^* \Delta\mathbf{y}_{t-i} + \boldsymbol{\epsilon}_t$$

where $\boldsymbol{\alpha}\boldsymbol{\beta}' = -\Phi(1)$.

Granger Representation Theorem

Engle and Granger (1987) define

$$\Pi(z) \equiv (1 - z)I_k - \boldsymbol{\alpha}\boldsymbol{\beta}'z - \sum_{i=1}^{p-1} \Phi_i^*(1 - z)z^i$$

and the following assumptions hold:

1. $|\Pi(z)| = 0 \Rightarrow |z| > 1$ or $z = 1$.
2. The number of unit roots, $z = 1$, is exactly $k - r$.
3. $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are $k \times r$ matrices, and their ranks are both r.

Then y_t has the representation

$$y_t = C\sum_{i=1}^{t} \boldsymbol{\epsilon}_i + C^*(B)\boldsymbol{\epsilon}_t + y_0^*$$

where the Granger representation coefficient, C, is

$$C = \boldsymbol{\beta}_\perp \left[\boldsymbol{\alpha}'_\perp \Phi(1)\boldsymbol{\beta}_\perp\right]^{-1} \boldsymbol{\alpha}'_\perp$$

where the full-rank $k \times (k-r)$ matrix $\boldsymbol{\beta}_\perp$ is orthogonal to $\boldsymbol{\beta}$ and the full-rank $k \times (k-r)$ matrix $\boldsymbol{\alpha}_\perp$ is orthogonal to $\boldsymbol{\alpha}$. $C^*(B)\boldsymbol{\epsilon}_t = \sum_{j=1}^{\infty} C_j^* \boldsymbol{\epsilon}_{t-j}$ is an $I(0)$ process, and y_0^* depends on the initial values.

The Granger representation coefficient C can be defined only when the $(k-r) \times (k-r)$ matrix $\boldsymbol{\alpha}_\perp' \Phi(1) \boldsymbol{\beta}_\perp$ is invertible.

One motivation for the VECM(p) form is to consider the relation $\boldsymbol{\beta}' \mathbf{y}_t = \mathbf{c}$ as defining the underlying economic relations. Assume that agents react to the disequilibrium error $\boldsymbol{\beta}' \mathbf{y}_t - \mathbf{c}$ through the adjustment coefficient $\boldsymbol{\alpha}$ to restore equilibrium. The cointegrating vector, $\boldsymbol{\beta}$, is sometimes called the long-run parameter.

Consider a vector error correction model that has a deterministic term, D_t, which can contain a constant, a linear trend, and seasonal dummy variables. Exogenous variables can also be included in the model. The model has the form

$$\Delta \mathbf{y}_t = \Pi \mathbf{y}_{t-1} + \sum_{i=1}^{p-1} \Phi_i^* \Delta \mathbf{y}_{t-i} + A D_t + \sum_{i=0}^{s} \Theta_i^* \mathbf{x}_{t-i} + \boldsymbol{\epsilon}_t$$

where $\Pi = \boldsymbol{\alpha}\boldsymbol{\beta}'$.

The alternative vector error correction representation considers the error correction term at lag $t-p$ and is written as

$$\Delta \mathbf{y}_t = \sum_{i=1}^{p-1} \Phi_i^\sharp \Delta \mathbf{y}_{t-i} + \Pi^\sharp \mathbf{y}_{t-p} + A D_t + \sum_{i=0}^{s} \Theta_i^* \mathbf{x}_{t-i} + \boldsymbol{\epsilon}_t$$

If the matrix Π has a full rank ($r = k$), all components of $\mathbf{y}_t$ are $I(0)$. On the other hand, $\mathbf{y}_t$ are stationary in difference if $\mathrm{rank}(\Pi) = 0$. When the rank of the matrix Π is $r < k$, there are $k - r$ linear combinations that are nonstationary and r stationary cointegrating relations. Note that the linearly independent vector $\mathbf{z}_t = \boldsymbol{\beta}' \mathbf{y}_t$ is stationary and this transformation is not unique unless $r = 1$. There does not exist a unique cointegrating matrix $\boldsymbol{\beta}$ because the coefficient matrix Π can also be decomposed as

$$\Pi = \boldsymbol{\alpha} M M^{-1} \boldsymbol{\beta}' = \boldsymbol{\alpha}^* \boldsymbol{\beta}^{*'}$$

where M is an $r \times r$ nonsingular matrix.

Test for Cointegration

The cointegration rank test determines the linearly independent columns of Π. Johansen and Juselius proposed the cointegration rank test by using the reduced rank regression (Johansen 1988, 1995b; Johansen and Juselius 1990).

Different Specifications of Deterministic Trends

When you construct the VECM(p) form from the VAR(p) model, the deterministic terms in the VECM(p) form can differ from those in the VAR(p) model. When there are deterministic cointegrated relationships among variables, deterministic terms in the VAR(p) model are not present in the VECM(p) form. On the other hand, if there are stochastic cointegrated relationships in the VAR(p) model, deterministic terms appear in the VECM(p) form via the error correction term or as an independent term in the VECM(p) form. There are five different specifications of deterministic trends in the VECM(p) form.

- **Case 1:** There is no separate drift in the VECM(p) form.

$$\Delta \mathbf{y}_t = \boldsymbol{\alpha}\boldsymbol{\beta}'\mathbf{y}_{t-1} + \sum_{i=1}^{p-1} \Phi_i^* \Delta \mathbf{y}_{t-i} + \boldsymbol{\epsilon}_t$$

- **Case 2:** There is no separate drift in the VECM(p) form, but a constant enters only via the error correction term.

$$\Delta \mathbf{y}_t = \boldsymbol{\alpha}(\boldsymbol{\beta}', \boldsymbol{\beta}_0)(\mathbf{y}_{t-1}', 1)' + \sum_{i=1}^{p-1} \Phi_i^* \Delta \mathbf{y}_{t-i} + \boldsymbol{\epsilon}_t$$

- **Case 3:** There is a separate drift and no separate linear trend in the VECM(p) form.

$$\Delta \mathbf{y}_t = \boldsymbol{\alpha}\boldsymbol{\beta}'\mathbf{y}_{t-1} + \sum_{i=1}^{p-1} \Phi_i^* \Delta \mathbf{y}_{t-i} + \boldsymbol{\delta}_0 + \boldsymbol{\epsilon}_t$$

- **Case 4:** There is a separate drift and no separate linear trend in the VECM(p) form, but a linear trend enters only via the error correction term.

$$\Delta \mathbf{y}_t = \boldsymbol{\alpha}(\boldsymbol{\beta}', \boldsymbol{\beta}_1)(\mathbf{y}_{t-1}', t)' + \sum_{i=1}^{p-1} \Phi_i^* \Delta \mathbf{y}_{t-i} + \boldsymbol{\delta}_0 + \boldsymbol{\epsilon}_t$$

- **Case 5:** There is a separate linear trend in the VECM(p) form.

$$\Delta \mathbf{y}_t = \boldsymbol{\alpha}\boldsymbol{\beta}'\mathbf{y}_{t-1} + \sum_{i=1}^{p-1} \Phi_i^* \Delta \mathbf{y}_{t-i} + \boldsymbol{\delta}_0 + \boldsymbol{\delta}_1 t + \boldsymbol{\epsilon}_t$$

First, focus on Cases 1, 3, and 5 to test the null hypothesis that there are at most r cointegrating vectors. Let

$$\begin{aligned}
Z_{0t} &= \Delta \mathbf{y}_t \\
Z_{1t} &= \mathbf{y}_{t-1} \\
Z_{2t} &= [\Delta \mathbf{y}_{t-1}', \ldots, \Delta \mathbf{y}_{t-p+1}', D_t]' \\
Z_0 &= [Z_{01}, \ldots, Z_{0T}]' \\
Z_1 &= [Z_{11}, \ldots, Z_{1T}]' \\
Z_2 &= [Z_{21}, \ldots, Z_{2T}]'
\end{aligned}$$

where D_t can be empty for Case 1, 1 for Case 3, and $(1, t)$ for Case 5.

In Case 2, Z_{1t} and Z_{2t} are defined as

$$\begin{aligned}
Z_{1t} &= [\mathbf{y}_{t-1}', 1]' \\
Z_{2t} &= [\Delta \mathbf{y}_{t-1}', \ldots, \Delta \mathbf{y}_{t-p+1}']'
\end{aligned}$$

In Case 4, Z_{1t} and Z_{2t} are defined as

$$\begin{aligned} Z_{1t} &= [\mathbf{y}'_{t-1}, t]' \\ Z_{2t} &= [\Delta\mathbf{y}'_{t-1}, \ldots, \Delta\mathbf{y}'_{t-p+1}, 1]' \end{aligned}$$

Let Ψ be the matrix of parameters consisting of $\Phi^*_1, \ldots, \Phi^*_{p-1}$, A, and $\Theta^*_0, \ldots, \Theta^*_s$, where parameter A corresponds with the regressors D_t. Then the VECM(p) form is rewritten in these variables as

$$Z_{0t} = \boldsymbol{\alpha}\boldsymbol{\beta}' Z_{1t} + \Psi Z_{2t} + \boldsymbol{\epsilon}_t$$

The log-likelihood function is given by

$$\begin{aligned} \ell &= -\frac{kT}{2}\log 2\pi - \frac{T}{2}\log|\Sigma| \\ &\quad -\frac{1}{2}\sum_{t=1}^{T}(Z_{0t} - \boldsymbol{\alpha}\boldsymbol{\beta}' Z_{1t} - \Psi Z_{2t})'\Sigma^{-1}(Z_{0t} - \boldsymbol{\alpha}\boldsymbol{\beta}' Z_{1t} - \Psi Z_{2t}) \end{aligned}$$

The residuals, R_{0t} and R_{1t}, are obtained by regressing Z_{0t} and Z_{1t} on Z_{2t}, respectively. The regression equation of residuals is

$$R_{0t} = \boldsymbol{\alpha}\boldsymbol{\beta}' R_{1t} + \hat{\boldsymbol{\epsilon}}_t$$

The crossproducts matrices are computed

$$S_{ij} = \frac{1}{T}\sum_{t=1}^{T} R_{it} R'_{jt}, \quad i, j = 0, 1$$

Then the maximum likelihood estimator for $\boldsymbol{\beta}$ is obtained from the eigenvectors that correspond to the r largest eigenvalues of the following equation:

$$|\lambda S_{11} - S_{10} S_{00}^{-1} S_{01}| = 0$$

The eigenvalues of the preceding equation are squared canonical correlations between R_{0t} and R_{1t}, and the eigenvectors that correspond to the r largest eigenvalues are the r linear combinations of $\mathbf{y}_{t-1}$, which have the largest squared partial correlations with the stationary process $\Delta\mathbf{y}_t$ after correcting for lags and deterministic terms. Such an analysis calls for a reduced rank regression of $\Delta\mathbf{y}_t$ on $\mathbf{y}_{t-1}$ corrected for $(\Delta\mathbf{y}_{t-1}, \ldots, \Delta\mathbf{y}_{t-p+1}, D_t)$, as discussed by Anderson (1951). Johansen (1988) suggests two test statistics to test the null hypothesis that there are at most r cointegrating vectors

$$\mathrm{H_0} : \lambda_i = 0 \text{ for } i = r+1, \ldots, k$$

Trace Test

The trace statistic for testing the null hypothesis that there are at most r cointegrating vectors is as follows:

$$\lambda_{\text{trace}} = -T \sum_{i=r+1}^{k} \log(1 - \lambda_i)$$

The asymptotic distribution of this statistic is given by

$$\text{tr}\left\{ \int_0^1 (dW)\tilde{W}' \left(\int_0^1 \tilde{W}\tilde{W}' dr \right)^{-1} \int_0^1 \tilde{W}(dW)' \right\}$$

where $\text{tr}(A)$ is the trace of a matrix A, W is the $k - r$ dimensional Brownian motion, and $\tilde{W}$ is the Brownian motion itself, or the de-meaned or detrended Brownian motion according to the different specifications of deterministic trends in the vector error correction model.

Maximum Eigenvalue Test

The maximum eigenvalue statistic for testing the null hypothesis that there are at most r cointegrating vectors is as follows:

$$\lambda_{\max} = -T \log(1 - \lambda_{r+1})$$

The asymptotic distribution of this statistic is given by

$$\max\{ \int_0^1 (dW)\tilde{W}' (\int_0^1 \tilde{W}\tilde{W}' dr)^{-1} \int_0^1 \tilde{W}(dW)' \}$$

where $\max(A)$ is the maximum eigenvalue of a matrix A. Osterwald-Lenum (1992) provided detailed tables of the critical values of these statistics.

The following statements use the JOHANSEN option to compute the Johansen cointegration rank trace test of integrated order 1:

```
proc varmax data=simul2;
   model y1 y2 / p=2 cointtest=(johansen=(normalize=y1));
run;
```

Figure 42.72 shows the output based on the model specified in the MODEL statement. An intercept term is assumed. In the "Cointegration Rank Test Using Trace" table, the column Drift in ECM indicates that there is no separate drift in the error correction model, and the column Drift in Process indicates that the process has a constant drift before differencing. The "Cointegration Rank Test Using Trace" table shows the trace statistics and p-values based on Case 3, and the "Cointegration Rank Test Using Trace under Restriction" table shows the trace statistics and p-values based on Case 2. For a specified significance level, such as 5%, the output indicates that the null hypothesis that the series are not cointegrated (H0: Rank = 0) can be rejected, because the p-values for both Case 2 and Case 3 are less than 0.05. The output also shows that the null hypothesis that the series are cointegrated with rank 1 (H0: Rank = 1) cannot be rejected for either Case 2 or Case 3, because the p-values for these tests are both greater than 0.05.

Figure 42.72 Cointegration Rank Test (COINTTEST=(JOHANSEN=) Option)

The VARMAX Procedure

Cointegration Rank Test Using Trace

H0: Rank=r	H1: Rank>r	Eigenvalue	Trace	Pr > Trace	Drift in ECM	Drift in Process
0	0	0.4644	61.7522	<.0001	Constant	Linear
1	1	0.0056	0.5552	0.4559		

Cointegration Rank Test Using Trace Under Restriction

H0: Rank=r	H1: Rank>r	Eigenvalue	Trace	Pr > Trace	Drift in ECM	Drift in Process
0	0	0.5209	76.3788	<.0001	Constant	Constant
1	1	0.0426	4.2680	0.3741		

Figure 42.73 shows which result, either Case 2 (the hypothesis H0) or Case 3 (the hypothesis H1), is appropriate depending on the significance level. Since the cointegration rank is chosen to be 1 by the result in Figure 42.72, look at the last row that corresponds to rank=1. Since the *p*-value is 0.054, the Case 2 cannot be rejected at the significance level 5%, but it can be rejected at the significance level 10%. For modeling of the two Case 2 and Case 3, see Figure 42.76 and Figure 42.77.

Figure 42.73 Cointegration Rank Test, Continued

Hypothesis of the Restriction

Hypothesis	Drift in ECM	Drift in Process
H0(Case 2)	Constant	Constant
H1(Case 3)	Constant	Linear

Hypothesis Test of the Restriction

Rank	Eigenvalue	Restricted Eigenvalue	DF	Chi-Square	Pr > ChiSq
0	0.4644	0.5209	2	14.63	0.0007
1	0.0056	0.0426	1	3.71	0.0540

Figure 42.74 shows the estimates of long-run parameter (Beta) and adjustment coefficients (Alpha) based on Case 3.

Figure 42.74 Cointegration Rank Test, Continued

Beta

Variable	1	2
y1	1.00000	1.00000
y2	-2.04869	-0.02854

Alpha

Variable	1	2
y1	-0.46421	-0.00502
y2	0.17535	-0.01275

Using the NORMALIZE= option, the first row of the "Beta" table has 1. Considering that the cointegration rank is 1, the long-run relationship of the series is

$$\begin{aligned}
\boldsymbol{\beta}' y_t &= \begin{bmatrix} 1 & -2.04869 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \\
&= y_{1t} - 2.04869 y_{2t} \\
y_{1t} &= 2.04869 y_{2t}
\end{aligned}$$

Figure 42.75 shows the estimates of long-run parameter (Beta) and adjustment coefficients (Alpha) based on Case 2.

Figure 42.75 Cointegration Rank Test, Continued

Beta Under Restriction		
Variable	**1**	**2**
y1	1.00000	1.00000
y2	-2.04366	-2.75773
1	6.75919	101.37051

Alpha Under Restriction		
Variable	**1**	**2**
y1	-0.48015	0.01091
y2	0.12538	0.03722

Considering that the cointegration rank is 1, the long-run relationship of the series is

$$\begin{aligned}
\boldsymbol{\beta}' y_t &= \begin{bmatrix} 1 & -2.04366 & 6.75919 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ 1 \end{bmatrix} \\
&= y_{1t} - 2.04366\, y_{2t} + 6.75919 \\
y_{1t} &= 2.04366\, y_{2t} - 6.75919
\end{aligned}$$

Estimation of Vector Error Correction Model

The preceding log-likelihood function is maximized for

$$\begin{aligned}
\hat{\boldsymbol{\beta}} &= S_{11}^{-1/2}[v_1, \ldots, v_r] \\
\hat{\boldsymbol{\alpha}} &= S_{01}\hat{\boldsymbol{\beta}}(\hat{\boldsymbol{\beta}}' S_{11} \hat{\boldsymbol{\beta}})^{-1} \\
\hat{\Pi} &= \hat{\boldsymbol{\alpha}}\hat{\boldsymbol{\beta}}' \\
\hat{\Psi}' &= (Z_2' Z_2)^{-1} Z_2' (Z_0 - Z_1 \hat{\Pi}') \\
\hat{\Sigma} &= (Z_0 - Z_2\hat{\Psi}' - Z_1\hat{\Pi}')'(Z_0 - Z_2\hat{\Psi}' - Z_1\hat{\Pi}')/T
\end{aligned}$$

The estimators of the orthogonal complements of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are

$$\hat{\boldsymbol{\beta}}_\perp = S_{11}[v_{r+1}, \ldots, v_k]$$

and

$$\hat{\boldsymbol{\alpha}}_{\perp} = S_{00}^{-1} S_{01}[v_{r+1}, \ldots, v_k]$$

Let ϑ denote the parameter vector $(\mathrm{vec}(\boldsymbol{\alpha}, \Psi)', \mathrm{vech}(\Sigma)')'$. The covariance of parameter estimates $\hat{\vartheta}$ is obtained as the inverse of the negative Hessian matrix $H \equiv \frac{\partial^2 \ell}{\partial \vartheta \partial \vartheta'}$. Because $\hat{\Pi} = \hat{\boldsymbol{\alpha}}\hat{\boldsymbol{\beta}}'$, the variance of $\hat{\Pi}$ and the covariance between $\hat{\Pi}$ and $\hat{\vartheta}$ are calculated as follows:

$$\mathrm{cov}(\mathrm{vec}(\hat{\Pi}), \mathrm{vec}(\hat{\Pi})) = (\hat{\boldsymbol{\beta}} \otimes I_k)\mathrm{cov}(\mathrm{vec}(\hat{\boldsymbol{\alpha}}), \mathrm{vec}(\hat{\boldsymbol{\alpha}}))(\hat{\boldsymbol{\beta}} \otimes I_k)'$$

$$\mathrm{cov}(\mathrm{vec}(\hat{\Pi}), \hat{\vartheta}) = (\hat{\boldsymbol{\beta}} \otimes I_k)\mathrm{cov}(\mathrm{vec}(\hat{\boldsymbol{\alpha}}), \hat{\vartheta})$$

For Case 2 (Case 4), because the coefficient vector $\hat{\boldsymbol{\delta}}_0$ ($\hat{\boldsymbol{\delta}}_1$) for the constant term (the linear trend term) is the product of $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}_0$ ($\hat{\boldsymbol{\beta}}_1$), the variance of $\hat{\boldsymbol{\delta}}_0$ ($\hat{\boldsymbol{\delta}}_1$) and the covariance between $\hat{\boldsymbol{\delta}}_0$ ($\hat{\boldsymbol{\delta}}_1$) and $\hat{\vartheta}$ are calculated as follows:

$$\mathrm{cov}(\hat{\boldsymbol{\delta}}_i, \hat{\boldsymbol{\delta}}_i) = (\hat{\boldsymbol{\beta}}_i' \otimes I_k)\mathrm{cov}(\mathrm{vec}(\hat{\boldsymbol{\alpha}}), \mathrm{vec}(\hat{\boldsymbol{\alpha}}))(\hat{\boldsymbol{\beta}}_i' \otimes I_k)', \quad i = 0 \text{ or } 1$$

$$\mathrm{cov}(\hat{\boldsymbol{\delta}}_i, \hat{\vartheta}) = (\hat{\boldsymbol{\beta}}_i' \otimes I_k)\mathrm{cov}(\mathrm{vec}(\hat{\boldsymbol{\alpha}}), \hat{\vartheta}), \quad i = 0 \text{ or } 1$$

The following statements are examples of fitting the five different cases of the vector error correction models mentioned in the previous section.

For fitting Case 1,

```
model y1 y2 / p=2 noint;
cointeg rank=1 normalize=y1;
```

For fitting Case 2,

```
model y1 y2 / p=2;
cointeg rank=1 normalize=y1 ectrend;
```

For fitting Case 3,

```
model y1 y2 / p=2;
cointeg rank=1 normalize=y1;
```

For fitting Case 4,

```
model y1 y2 / p=2 trend=linear;
cointeg rank=1 normalize=y1 ectrend;
```

For fitting Case 5,

```
model y1 y2 / p=2 trend=linear;
cointeg rank=1 normalize=y1;
```

In the previous example, the output from the COINTTEST=(JOHANSEN) option shown in Figure 42.73 indicates that you can fit the model by using either Case 2 or Case 3 because the test of the restriction was not significant at the 0.05 level, but was significant at the 0.10 level. Following both models are fit to show the differences in the displayed output. Figure 42.76 is for Case 2, and Figure 42.77 is for Case 3.

For Case 2,

```
proc varmax data=simul2;
   model y1 y2 / p=2 print=(estimates);
   cointeg rank=1 normalize=y1 ectrend;
run;
```

Figure 42.76 Parameter Estimation with the ECTREND Option

The VARMAX Procedure

Parameter Alpha * Beta' Estimates

Variable	y1	y2	1
y1	-0.48015	0.98126	-3.24543
y2	0.12538	-0.25624	0.84748

AR Coefficients of Differenced Lag

DIF Lag	Variable	y1	y2
1	y1	-0.72759	-0.77463
	y2	0.38982	-0.55173

Model Parameter Estimates

Equation	Parameter	Estimate	Standard Error	t Value	Pr > |t|	Variable
D_y1	CONST1	-3.24543	0.33022	-9.83	<.0001	1, EC
	AR1_1_1	-0.48015	0.04886	-9.83	<.0001	y1(t-1)
	AR1_1_2	0.98126	0.09984	9.83	<.0001	y2(t-1)
	AR2_1_1	-0.72759	0.04623	-15.74	<.0001	D_y1(t-1)
	AR2_1_2	-0.77463	0.04978	-15.56	<.0001	D_y2(t-1)
D_y2	CONST2	0.84748	0.35394	2.39	0.0187	1, EC
	AR1_2_1	0.12538	0.05236	2.39	0.0187	y1(t-1)
	AR1_2_2	-0.25624	0.10702	-2.39	0.0187	y2(t-1)
	AR2_2_1	0.38982	0.04955	7.87	<.0001	D_y1(t-1)
	AR2_2_2	-0.55173	0.05336	-10.34	<.0001	D_y2(t-1)

Figure 42.76 can be reported as follows:

$$\Delta \mathbf{y}_t = \begin{bmatrix} -0.48015 & 0.98126 & -3.24543 \\ 0.12538 & -0.25624 & 0.84748 \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ y_{2,t-1} \\ 1 \end{bmatrix} + \begin{bmatrix} -0.72759 & -0.77463 \\ 0.38982 & -0.55173 \end{bmatrix} \Delta \mathbf{y}_{t-1} + \boldsymbol{\epsilon}_t$$

The keyword “EC” in the “Model Parameter Estimates” table means that the ECTREND option is used for fitting the model.

For fitting Case 3,

```
proc varmax data=simul2;
   model y1 y2 / p=2 print=(estimates);
   cointeg rank=1 normalize=y1;
run;
```

Figure 42.77 Parameter Estimation without the ECTREND Option

The VARMAX Procedure

Parameter Alpha * Beta' Estimates		
Variable	y1	y2
y1	-0.46421	0.95103
y2	0.17535	-0.35923

AR Coefficients of Differenced Lag			
DIF Lag	Variable	y1	y2
1	y1	-0.74052	-0.76305
	y2	0.34820	-0.51194

Model Parameter Estimates						
Equation	Parameter	Estimate	Standard Error	t Value	Pr > |t|	Variable
D_y1	CONST1	-2.60825	1.32398	-1.97	0.0518	1
	AR1_1_1	-0.46421	0.05474	-8.48	<.0001	y1(t-1)
	AR1_1_2	0.95103	0.11215	8.48	<.0001	y2(t-1)
	AR2_1_1	-0.74052	0.05060	-14.63	<.0001	D_y1(t-1)
	AR2_1_2	-0.76305	0.05352	-14.26	<.0001	D_y2(t-1)
D_y2	CONST2	3.43005	1.39587	2.46	0.0159	1
	AR1_2_1	0.17535	0.05771	3.04	0.0031	y1(t-1)
	AR1_2_2	-0.35923	0.11824	-3.04	0.0031	y2(t-1)
	AR2_2_1	0.34820	0.05335	6.53	<.0001	D_y1(t-1)
	AR2_2_2	-0.51194	0.05643	-9.07	<.0001	D_y2(t-1)

Figure 42.77 can be reported as follows:

$$\Delta \mathbf{y}_t = \begin{bmatrix} -0.46421 & 0.95103 \\ 0.17535 & -0.35293 \end{bmatrix} \mathbf{y}_{t-1} + \begin{bmatrix} -0.74052 & -0.76305 \\ 0.34820 & -0.51194 \end{bmatrix} \Delta \mathbf{y}_{t-1} + \begin{bmatrix} -2.60825 \\ 3.43005 \end{bmatrix} + \boldsymbol{\epsilon}_t$$

A Test for the Long-Run Relations

Consider the example with the variables m_t log real money, y_t log real income, i_t^d deposit interest rate, and i_t^b bond interest rate. It seems a natural hypothesis that in the long-run relation, money and income have equal coefficients with opposite signs. This can be formulated as the hypothesis that the cointegrated relation contains only m_t and y_t through $m_t - y_t$. For the analysis, you can express these restrictions in the

parameterization of **H** such that $\boldsymbol{\beta} = H\phi$, where **H** is a known $k \times s$ matrix and ψ is the $s \times r (r \leq s < k)$ parameter matrix to be estimated. For this example, **H** is given by

$$H = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Restriction $H_0: \boldsymbol{\beta} = H\phi$

When the linear restriction $\boldsymbol{\beta} = H\phi$ is given, it implies that the same restrictions are imposed on all cointegrating vectors. You obtain the maximum likelihood estimator of $\boldsymbol{\beta}$ by reduced rank regression of $\Delta \mathbf{y}_t$ on $H\mathbf{y}_{t-1}$ corrected for $(\Delta \mathbf{y}_{t-1}, \ldots, \Delta \mathbf{y}_{t-p+1}, D_t)$, solving the following equation,

$$|\rho H' S_{11} H - H' S_{10} S_{00}^{-1} S_{01} H| = 0$$

for the eigenvalues $1 > \rho_1 > \cdots > \rho_s > 0$ and eigenvectors $(v_1, \ldots, v_s)$, S_{ij} given in the preceding section. Then choose $\hat{\phi} = (v_1, \ldots, v_r)$ that corresponds to the r largest eigenvalues, and the $\hat{\boldsymbol{\beta}}$ is $H\hat{\phi}$.

The test statistic for $H_0: \boldsymbol{\beta} = H\phi$ is given by

$$T \sum_{i=1}^{r} \log\{(1 - \rho_i)/(1 - \lambda_i)\} \xrightarrow{d} \chi^2_{r(k-s)}$$

If the series has no deterministic trend, the constant term should be restricted by $\boldsymbol{\alpha}'_{\perp} \boldsymbol{\delta}_0 = 0$ as in Case 2. Then **H** is given by

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The following statements test that $2\,\beta_1 + \beta_2 = 0$:

```
proc varmax data=simul2;
   model y1 y2 / p=2;
   cointeg rank=1 h=(1,-2) normalize=y1;
run;
```

Figure 42.78 shows the results of testing $H_0: 2\beta_1 + \beta_2 = 0$. The input **H** matrix is $H = (1 - 2)'$. The adjustment coefficient is reestimated under the restriction, and the test indicates that you cannot reject the null hypothesis.

Figure 42.78 Testing of Linear Restriction (H= Option)

The VARMAX Procedure

Beta Under Restriction	
Variable	**1**
y1	1.00000
y2	-2.00000

Alpha Under Restriction	
Variable	**1**
y1	-0.47404
y2	0.17534

Hypothesis Test					
Index	**Eigenvalue**	**Restricted Eigenvalue**	**DF**	**Chi-Square**	**Pr > ChiSq**
1	0.4644	0.4616	1	0.51	0.4738

Test for the Weak Exogeneity and Restrictions of Alpha

Consider a vector error correction model:

$$\Delta \mathbf{y}_t = \boldsymbol{\alpha}\boldsymbol{\beta}'\mathbf{y}_{t-1} + \sum_{i=1}^{p-1} \Phi_i^* \Delta \mathbf{y}_{t-i} + AD_t + \boldsymbol{\epsilon}_t$$

Divide the process $\mathbf{y}_t$ into $(\mathbf{y}_{1t}', \mathbf{y}_{2t}')'$ with dimension k_1 and k_2 and the Σ into

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$$

Similarly, the parameters can be decomposed as follows:

$$\boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\alpha}_1 \\ \boldsymbol{\alpha}_2 \end{bmatrix} \quad \Phi_i^* = \begin{bmatrix} \Phi_{1i}^* \\ \Phi_{2i}^* \end{bmatrix} \quad A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$$

Then the VECM(p) form can be rewritten by using the decomposed parameters and processes:

$$\begin{bmatrix} \Delta \mathbf{y}_{1t} \\ \Delta \mathbf{y}_{2t} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\alpha}_1 \\ \boldsymbol{\alpha}_2 \end{bmatrix} \boldsymbol{\beta}'\mathbf{y}_{t-1} + \sum_{i=1}^{p-1} \begin{bmatrix} \Phi_{1i}^* \\ \Phi_{2i}^* \end{bmatrix} \Delta \mathbf{y}_{t-i} + \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} D_t + \begin{bmatrix} \boldsymbol{\epsilon}_{1t} \\ \boldsymbol{\epsilon}_{2t} \end{bmatrix}$$

The conditional model for $\mathbf{y}_{1t}$ given $\mathbf{y}_{2t}$ is

$$\begin{aligned} \Delta \mathbf{y}_{1t} &= \omega \Delta \mathbf{y}_{2t} + (\alpha_1 - \omega\alpha_2)\boldsymbol{\beta}'\mathbf{y}_{t-1} + \sum_{i=1}^{p-1} (\Phi_{1i}^* - \omega\Phi_{2i}^*)\Delta \mathbf{y}_{t-i} \\ &\quad + (A_1 - \omega A_2)D_t + \boldsymbol{\epsilon}_{1t} - \omega\boldsymbol{\epsilon}_{2t} \end{aligned}$$

and the marginal model of $\mathbf{y}_{2t}$ is

$$\Delta \mathbf{y}_{2t} = \alpha_2 \boldsymbol{\beta}' \mathbf{y}_{t-1} + \sum_{i=1}^{p-1} \Phi_{2i}^* \Delta \mathbf{y}_{t-i} + A_2 D_t + \boldsymbol{\epsilon}_{2t}$$

where $\omega = \Sigma_{12}\Sigma_{22}^{-1}$.

The test of weak exogeneity of $\mathbf{y}_{2t}$ for the parameters $(\alpha_1, \boldsymbol{\beta})$ determines whether $\alpha_2 = 0$. Weak exogeneity means that there is no information about $\boldsymbol{\beta}$ in the marginal model or that the variables $\mathbf{y}_{2t}$ do not react to a disequilibrium.

Restriction $H_0: \boldsymbol{\alpha} = J\psi$

Consider the null hypothesis $H_0: \boldsymbol{\alpha} = J\psi$, where J is a $k \times m$ matrix with $r \leq m < k$.

From the previous residual regression equation

$$R_{0t} = \boldsymbol{\alpha}\boldsymbol{\beta}' R_{1t} + \hat{\boldsymbol{\epsilon}}_t = J\psi\boldsymbol{\beta}' R_{1t} + \hat{\boldsymbol{\epsilon}}_t$$

you can obtain

$$\begin{aligned} \bar{J}' R_{0t} &= \psi\boldsymbol{\beta}' R_{1t} + \bar{J}'\hat{\boldsymbol{\epsilon}}_t \\ J_\perp' R_{0t} &= J_\perp'\hat{\boldsymbol{\epsilon}}_t \end{aligned}$$

where $\bar{J} = J(J'J)^{-1}$ and $J_\perp$ is orthogonal to J such that $J_\perp' J = 0$.

Define

$$\Sigma_{JJ_\perp} = \bar{J}'\Sigma J_\perp \text{ and } \Sigma_{J_\perp J_\perp} = J_\perp'\Sigma J_\perp$$

and let $\omega = \Sigma_{JJ_\perp}\Sigma_{J_\perp J_\perp}^{-1}$. Then $\bar{J}' R_{0t}$ can be written as

$$\bar{J}' R_{0t} = \psi\boldsymbol{\beta}' R_{1t} + \omega J_\perp' R_{0t} + \bar{J}'\hat{\boldsymbol{\epsilon}}_t - \omega J_\perp'\hat{\boldsymbol{\epsilon}}_t$$

Using the marginal distribution of $J_\perp' R_{0t}$ and the conditional distribution of $\bar{J}' R_{0t}$, the new residuals are computed as

$$\begin{aligned} \tilde{R}_{Jt} &= \bar{J}' R_{0t} - S_{JJ_\perp} S_{J_\perp J_\perp}^{-1} J_\perp' R_{0t} \\ \tilde{R}_{1t} &= R_{1t} - S_{1J_\perp} S_{J_\perp J_\perp}^{-1} J_\perp' R_{0t} \end{aligned}$$

where

$$S_{JJ_\perp} = \bar{J}' S_{00} J_\perp, \quad S_{J_\perp J_\perp} = J_\perp' S_{00} J_\perp, \quad \text{and} \quad S_{J_\perp 1} = J_\perp' S_{01}$$

In terms of $\tilde{R}_{Jt}$ and $\tilde{R}_{1t}$, the MLE of $\boldsymbol{\beta}$ is computed by using the reduced rank regression. Let

$$S_{ij.J_\perp} = \frac{1}{T}\sum_{t=1}^{T} \tilde{R}_{it}\tilde{R}_{jt}', \quad \text{for } i, j = 1, J$$

Under the null hypothesis $H_0: \boldsymbol{\alpha} = J\psi$, the MLE $\tilde{\boldsymbol{\beta}}$ is computed by solving the equation

$$|\rho S_{11.J_\perp} - S_{1J.J_\perp} S_{JJ.J_\perp}^{-1} S_{J1.J_\perp}| = 0$$

Then $\tilde{\boldsymbol{\beta}} = (v_1, \ldots, v_r)$, where the eigenvectors correspond to the r largest eigenvalues and are normalized such that $\tilde{\boldsymbol{\beta}}' S_{11.J_\perp} \tilde{\boldsymbol{\beta}} = I_r$; $\tilde{\boldsymbol{\alpha}} = J S_{J1.J_\perp} \tilde{\boldsymbol{\beta}}$. The likelihood ratio test for $H_0: \boldsymbol{\alpha} = J\psi$ is

$$T \sum_{i=1}^{r} \log\{(1-\rho_i)/(1-\lambda_i)\} \xrightarrow{d} \chi^2_{r(k-m)}$$

For more information, see Theorem 6.1 in Johansen and Juselius (1990).

The test of weak exogeneity of $\mathbf{y}_{2t}$ is a special case of the test $\boldsymbol{\alpha} = J\psi$, considering $J = (I_{k_1}, 0)'$. Consider the previous example with four variables (m_t, y_t, i_t^b, i_t^d). If $r = 1$, you formulate the weak exogeneity of (y_t, i_t^b, i_t^d) for m_t as $J = [0, I_3]'$ and the weak exogeneity of i_t^d for (m_t, y_t, i_t^b) as $J = [I_3, 0]'$.

The following statements test the weak exogeneity of other variables, assuming $r = 1$:

```
proc varmax data=simul2;
   model y1 y2 / p=2;
   cointeg rank=1 exogeneity normalize=y1;
run;
```

Figure 42.79 shows that each variable is not the weak exogeneity of other variable.

Figure 42.79 Testing of Weak Exogeneity (EXOGENEITY Option)

The VARMAX Procedure

Testing Weak Exogeneity of Each Variable

Variable	DF	Chi-Square	Pr > ChiSq
y1	1	53.46	<.0001
y2	1	8.76	0.0031

General Tests and Restrictions on Parameters

The previous sections discuss some special forms of tests on $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$, namely the long-run relations that are expressed in the form $H_0: \boldsymbol{\beta} = \mathbf{H}\boldsymbol{\phi}$, the weak exogeneity test, and the null hypotheses on $\boldsymbol{\alpha}$ in the form $H_0: \boldsymbol{\alpha} = \mathbf{J}\boldsymbol{\psi}$. In fact, with the help of the RESRICT and BOUND statements, you can estimate the models that have linear restrictions on any parameters to be estimated, which means that you can implement the likelihood ratio (LR) test for any linear relationship between the parameters.

The restricted error correction model must be estimated through numerical optimization. You might need to use the NLOPTIONS statement to try different options for the optimizer and the INITIAL statement to try different starting points. This is essentially important because the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are usually not identifiable.

You can also use the TEST statement to apply the Wald test for any linear relationships between parameters that are not long-run. Even more, you can test the constraints on $\Pi(= \boldsymbol{\alpha}\boldsymbol{\beta}')$ and $\boldsymbol{\delta}_0(= \boldsymbol{\alpha}\boldsymbol{\beta}_0)$ in Case 2 or $\boldsymbol{\delta}_1(= \boldsymbol{\alpha}\boldsymbol{\beta}_1)$ in Case 4 when the constant term or linear trend is restricted to the error correction term.

For more information and examples, see the section "Example 42.3: Analysis of Restricted Cointegrated Systems" on page 3167.

Forecasting of the VECM

Consider the cointegrated moving-average representation of the differenced process of $\mathbf{y}_t$

$$\Delta \mathbf{y}_t = \boldsymbol{\delta} + \Psi(B)\boldsymbol{\epsilon}_t$$

Assume that $\mathbf{y}_0 = 0$. The linear process $\mathbf{y}_t$ can be written as

$$\mathbf{y}_t = \boldsymbol{\delta} t + \sum_{i=1}^{t} \sum_{j=0}^{t-i} \Psi_j \boldsymbol{\epsilon}_i$$

Therefore, for any $l > 0$,

$$\mathbf{y}_{t+l} = \boldsymbol{\delta}(t+l) + \sum_{i=1}^{t} \sum_{j=0}^{t+l-i} \Psi_j \boldsymbol{\epsilon}_i + \sum_{i=1}^{l} \sum_{j=0}^{l-i} \Psi_j \boldsymbol{\epsilon}_{t+i}$$

The l-step-ahead forecast is derived from the preceding equation:

$$\mathbf{y}_{t+l|t} = (t+l) + \sum_{i=1}^{t} \sum_{j=0}^{t+l-i} \Psi_j \boldsymbol{\epsilon}_i$$

Note that

$$\lim_{l \to \infty} \boldsymbol{\beta}' \mathbf{y}_{t+l|t} = 0$$

since $\lim_{l\to\infty} \sum_{j=0}^{t+l-i} \Psi_j = \Psi(1)$ and $\boldsymbol{\beta}'\Psi(1) = 0$. The long-run forecast of the cointegrated system shows that the cointegrated relationship holds, although there might exist some deviations from the equilibrium status in the short-run. The covariance matrix of the predict error $\mathbf{e}_{t+l|t} = \mathbf{y}_{t+l} - \mathbf{y}_{t+l|t}$ is

$$\Sigma(l) = \sum_{i=1}^{l} [(\sum_{j=0}^{l-i} \Psi_j) \Sigma (\sum_{j=0}^{l-i} \Psi_j')]$$

When the linear process is represented as a VECM(p) model, you can obtain

$$\Delta \mathbf{y}_t = \Pi \mathbf{y}_{t-1} + \sum_{j=1}^{p-1} \Phi_j^* \Delta \mathbf{y}_{t-j} + \boldsymbol{\delta} + \boldsymbol{\epsilon}_t$$

The transition equation is defined as

$$\mathbf{z}_t = F \mathbf{z}_{t-1} + \mathbf{e}_t$$

where $\mathbf{z}_t = (\mathbf{y}'_{t-1}, \Delta\mathbf{y}'_t, \Delta\mathbf{y}'_{t-1}, \ldots, \Delta\mathbf{y}'_{t-p+2})'$ is a state vector and the transition matrix is

$$F = \begin{bmatrix} I_k & I_k & 0 & \cdots & 0 \\ \Pi & (\Pi + \Phi_1^*) & \Phi_2^* & \cdots & \Phi_{p-1}^* \\ 0 & I_k & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I_k & 0 \end{bmatrix}$$

where 0 is a $k \times k$ zero matrix. The observation equation can be written

$$\mathbf{y}_t = \boldsymbol{\delta} t + H\mathbf{z}_t$$

where $H = [I_k, I_k, 0, \ldots, 0]$.

The l-step-ahead forecast is computed as

$$\mathbf{y}_{t+l|t} = \boldsymbol{\delta}(t+l) + HF^l\mathbf{z}_t$$

Cointegration with Exogenous Variables

The error correction model with exogenous variables can be written as follows:

$$\Delta\mathbf{y}_t = \boldsymbol{\alpha}\boldsymbol{\beta}'\mathbf{y}_{t-1} + \sum_{i=1}^{p-1} \Phi_i^* \Delta\mathbf{y}_{t-i} + AD_t + \sum_{i=0}^{s} \Theta_i^* \mathbf{x}_{t-i} + \boldsymbol{\epsilon}_t$$

The following statements demonstrate how to fit VECMX(p, s), where $p = 2$ and $s = 1$ from the P=2 and XLAG=1 options:

```
proc varmax data=simul3;
   model y1 y2 = x1 / p=2 xlag=1;
   cointeg rank=1;
run;
```

The following statements demonstrate how to BVECMX(2,1):

```
proc varmax data=simul3;
   model y1 y2 = x1 / p=2 xlag=1
                      prior=(lambda=0.9 theta=0.1);
   cointeg rank=1;
run;
```

I(2) Model

The VARX(p,s) model can be written in the error correction form:

$$\Delta\mathbf{y}_t = \boldsymbol{\alpha}\boldsymbol{\beta}'\mathbf{y}_{t-1} + \sum_{i=1}^{p-1} \Phi_i^* \Delta\mathbf{y}_{t-i} + AD_t + \sum_{i=0}^{s} \Theta_i^* \mathbf{x}_{t-i} + \boldsymbol{\epsilon}_t$$

Let $\Phi^* = I_k - \sum_{i=1}^{p-1} \Phi_i^*$.

If $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ have full-rank r, and $\text{rank}(\boldsymbol{\alpha}'_\perp \Phi^* \boldsymbol{\beta}_\perp) = k - r$, then $\mathbf{y}_t$ is an $I(1)$ process.

If the condition $\text{rank}(\boldsymbol{\alpha}'_\perp \Phi^* \boldsymbol{\beta}_\perp) = k - r$ fails and $\boldsymbol{\alpha}'_\perp \Phi^* \boldsymbol{\beta}_\perp$ has reduced-rank $\boldsymbol{\alpha}'_\perp \Phi^* \boldsymbol{\beta}_\perp = \boldsymbol{\xi}\boldsymbol{\eta}'$ where $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ are $(k-r) \times s$ matrices with $s \leq k - r$, then $\boldsymbol{\alpha}_\perp$ and $\boldsymbol{\beta}_\perp$ are defined as $k \times (k-r)$ matrices of full rank such that $\boldsymbol{\alpha}'\boldsymbol{\alpha}_\perp = 0$ and $\boldsymbol{\beta}'\boldsymbol{\beta}_\perp = 0$.

If $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ have full-rank s, then the process $\mathbf{y}_t$ is $I(2)$, which has the implication of $I(2)$ model for the moving-average representation.

$$\mathbf{y}_t = B_0 + B_1 t + C_2 \sum_{j=1}^{t} \sum_{i=1}^{j} \boldsymbol{\epsilon}_i + C_1 \sum_{i=1}^{t} \boldsymbol{\epsilon}_i + C_0(B)\boldsymbol{\epsilon}_t$$

The matrices C_1, C_2, and $C_0(B)$ are determined by the cointegration properties of the process, and B_0 and B_1 are determined by the initial values. For more information, see Johansen (1995b).

The implication of the $I(2)$ model for the autoregressive representation is given by

$$\Delta^2 \mathbf{y}_t = \Pi \mathbf{y}_{t-1} - \Phi^* \Delta \mathbf{y}_{t-1} + \sum_{i=1}^{p-2} \Psi_i \Delta^2 \mathbf{y}_{t-i} + A D_t + \sum_{i=0}^{s} \Theta_i^* \mathbf{x}_{t-i} + \boldsymbol{\epsilon}_t$$

where $\Psi_i = -\sum_{j=i+1}^{p-1} \Phi_i^*$ and $\Phi^* = I_k - \sum_{i=1}^{p-1} \Phi_i^*$.

Test for I(2)

The $I(2)$ cointegrated model is given by the following parameter restrictions:

$$H_{r,s}: \Pi = \boldsymbol{\alpha}\boldsymbol{\beta}' \text{ and } \boldsymbol{\alpha}'_\perp \Phi^* \boldsymbol{\beta}_\perp = \boldsymbol{\xi}\boldsymbol{\eta}'$$

where $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ are $(k-r) \times s$ matrices with $0 \leq s \leq k - r$. Let H_r^0 represent the $I(1)$ model where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ have full-rank r, let $H_{r,s}^0$ represent the $I(2)$ model where $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ have full-rank s, and let $H_{r,s}$ represent the $I(2)$ model where $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ have rank $\leq s$. Table 42.6 shows the relation between the $I(1)$ models and the $I(2)$ models.

Table 42.6 Relation between the $I(1)$ and $I(2)$ Models

	$I(2)$										$I(1)$
$r \backslash k-r-s$	k		$k-1$		$\cdots$		1				
0	H_{00}	$\subset$	H_{01}	$\subset$	$\cdots$	$\subset$	$H_{0,k-1}$	$\subset$	H_{0k}	$=$	H_0^0
1			H_{10}	$\subset$	$\cdots$	$\subset$	$H_{1,k-2}$	$\subset$	$H_{1,k-1}$	$=$	H_1^0
$\vdots$							$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$k-1$							$H_{k-1,0}$	$\subset$	$H_{k-1,1}$	$=$	H_{k-1}^0

Johansen (1995b) proposed the two-step procedure to analyze the $I(2)$ model. In the first step, the values of $(r, \boldsymbol{\alpha}, \boldsymbol{\beta})$ are estimated using the reduced rank regression analysis, performing the regression analysis $\Delta^2 \mathbf{y}_t$,

$\Delta \mathbf{y}_{t-1}$, and $\mathbf{y}_{t-1}$ on $\Delta^2 \mathbf{y}_{t-1}, \ldots, \Delta^2 \mathbf{y}_{t-p+2}$, and D_t. This gives residuals R_{0t}, R_{1t}, and R_{2t}, and residual product moment matrices

$$M_{ij} = \frac{1}{T} \sum_{t=1}^{T} R_{it} R'_{jt} \quad \text{for } i, j = 0, 1, 2$$

Perform the reduced rank regression analysis $\Delta^2 \mathbf{y}_t$ on $\mathbf{y}_{t-1}$ corrected for $\Delta \mathbf{y}_{t-1}$, $\Delta^2 \mathbf{y}_{t-1}, \ldots, \Delta^2 \mathbf{y}_{t-p+2}$, and D_t, and solve the eigenvalue problem of the equation

$$|\lambda M_{22.1} - M_{20.1} M_{00.1}^{-1} M_{02.1}| = 0$$

where $M_{ij.1} = M_{ij} - M_{i1} M_{11}^{-1} M_{1j}$ for $i, j = 0, 2$.

In the second step, if $(r, \boldsymbol{\alpha}, \boldsymbol{\beta})$ are known, the values of $(s, \boldsymbol{\xi}, \boldsymbol{\eta})$ are determined using the reduced rank regression analysis, regressing $\hat{\boldsymbol{\alpha}}'_{\perp} \Delta^2 \mathbf{y}_t$ on $\hat{\boldsymbol{\beta}}'_{\perp} \Delta \mathbf{y}_{t-1}$ corrected for $\Delta^2 \mathbf{y}_{t-1}, \ldots, \Delta^2 \mathbf{y}_{t-p+2}$, D_t, and $\hat{\boldsymbol{\beta}}' \Delta \mathbf{y}_{t-1}$.

The reduced rank regression analysis reduces to the solution of an eigenvalue problem for the equation

$$|\rho M_{\boldsymbol{\beta}_\perp \boldsymbol{\beta}_\perp . \boldsymbol{\beta}} - M_{\boldsymbol{\beta}_\perp \boldsymbol{\alpha}_\perp . \boldsymbol{\beta}} M^{-1}_{\boldsymbol{\alpha}_\perp \boldsymbol{\alpha}_\perp . \boldsymbol{\beta}} M_{\boldsymbol{\alpha}_\perp \boldsymbol{\beta}_\perp . \boldsymbol{\beta}}| = 0$$

where

$$\begin{aligned}
M_{\boldsymbol{\beta}_\perp \boldsymbol{\beta}_\perp . \boldsymbol{\beta}} &= \boldsymbol{\beta}'_\perp (M_{11} - M_{11}\boldsymbol{\beta}(\boldsymbol{\beta}' M_{11} \boldsymbol{\beta})^{-1} \boldsymbol{\beta}' M_{11}) \boldsymbol{\beta}_\perp \\
M'_{\boldsymbol{\beta}_\perp \boldsymbol{\alpha}_\perp . \boldsymbol{\beta}} &= M_{\boldsymbol{\alpha}_\perp \boldsymbol{\beta}_\perp . \boldsymbol{\beta}} = \bar{\boldsymbol{\alpha}}'_\perp (M_{01} - M_{01}\boldsymbol{\beta}(\boldsymbol{\beta}' M_{11} \boldsymbol{\beta})^{-1} \boldsymbol{\beta}' M_{11}) \boldsymbol{\beta}_\perp \\
M_{\boldsymbol{\alpha}_\perp \boldsymbol{\alpha}_\perp . \boldsymbol{\beta}} &= \bar{\boldsymbol{\alpha}}'_\perp (M_{00} - M_{01}\boldsymbol{\beta}(\boldsymbol{\beta}' M_{11} \boldsymbol{\beta})^{-1} \boldsymbol{\beta}' M_{10}) \bar{\boldsymbol{\alpha}}_\perp
\end{aligned}$$

where $\bar{\boldsymbol{\alpha}} = \boldsymbol{\alpha}(\boldsymbol{\alpha}'\boldsymbol{\alpha})^{-1}$.

The solution gives eigenvalues $1 > \rho_1 > \cdots > \rho_s > 0$ and eigenvectors $(v_1, \ldots, v_s)$. Then, the ML estimators are

$$\begin{aligned}
\hat{\boldsymbol{\eta}} &= (v_1, \ldots, v_s) \\
\hat{\boldsymbol{\xi}} &= M_{\boldsymbol{\alpha}_\perp \boldsymbol{\beta}_\perp . \boldsymbol{\beta}} \hat{\eta}
\end{aligned}$$

The likelihood ratio test for the reduced rank model $H_{r,s}$ with rank $\le s$ in the model $H_{r,k-r} = H_r^0$ is given by

$$Q_{r,s} = -T \sum_{i=s+1}^{k-r} \log(1 - \rho_i), \quad s = 0, \ldots, k - r - 1$$

The following statements simulate an I(2) process and compute the rank test to test for cointegrated order 2:

```
proc iml;
   alpha  = { 1, 1};    * alphaOrthogonal = { 1, -1};
   beta   = { 1, -0.5}; * betaOrthogonal  = { 1, 2};
   * alphaOrthogonal' * phiStar * betaOrthogonal = 0;
   phiStar = { 1 0, 0 0.5};
   A1 = 2 * I(2) + alpha * beta` - phiStar;
```

```
   A2 = phiStar - I(2);
   phi = A1 // A2;
   sig = I(2);
   /* to simulate the vector time series */
   call varmasim(y,phi) sigma=sig n=200 seed=2;
   cn = {'y1' 'y2'};
   create simul4 from y[colname=cn];
   append from y;
   close;
quit;

proc varmax data=simul4;
   model y1 y2 /noint p=2 cointtest=(johansen=(iorder=2));
run;
```

The last two columns in Figure 42.80 explain the cointegration rank test with integrated order 1. For a specified significance level, such as 5%, the output indicates that the null hypothesis that the series are not cointegrated (H0: $r = 0$) is rejected, because the p-value for this test, shown in the column Pr > Trace of I(1), is less than 0.05. The results also indicate that the null hypothesis that there is a cointegrated relationship with cointegration rank 1 (H0: $r = 1$) cannot be rejected at the 5% significance level, because the p-value for the test statistic, 0.7961, is greater than 0.05. Because of this latter result, the rows in the table that are associated with $r = 1$ are further examined. The test statistic, 0.0257, tests the null hypothesis that the series are cointegrated order 2. The p-value that is associated with this test is 0.8955, which indicates that the null hypothesis cannot be rejected at the 5% significance level.

Figure 42.80 Cointegrated I(2) Test (IORDER= Option)

The VARMAX Procedure

Cointegration Rank Test for I(2)

r\k-r-s	2	1	Trace of I(1)	Pr > Trace of I(1)
0	575.3784	1.1833	215.3097	<.0001
Pr > Trace of I(2)	0.0000	0.3223		
1		0.0257	0.0986	0.7961
Pr > Trace of I(2)		0.8955		

Vector Error Correction Model in ARMA Form

The vector error correction model in ARMA form (the VEC-ARMA model) introduces MA terms and is defined as follows:

$$\Delta \mathbf{y}_t = \boldsymbol{\alpha}\boldsymbol{\beta}'\mathbf{y}_{t-1} + \sum_{i=1}^{p-1} \Phi_i^* \Delta \mathbf{y}_{t-i} + \boldsymbol{\epsilon}_t - \sum_{i=1}^{q} \Theta_i \boldsymbol{\epsilon}_{t-i}$$

The determined terms and the exogenous variables can also be introduced into the model. Similar to the VECM that has only AR terms, the constant term is constrained in the error correction term in Case 2 and the linear trend term is similarly constrained in Case 4.

The model is estimated through the maximum likelihood method. The log likelihood of the model is defined as

$$\ell = -\frac{T}{2}\log|\Sigma| - \frac{1}{2}\sum_{t=1}^{T}\mathbf{e}_t'\Sigma^{-1}\mathbf{e}_t$$

where

$$\mathbf{e}_t = \Delta\mathbf{y}_t - \boldsymbol{\alpha}\boldsymbol{\beta}'\mathbf{y}_{t-1} - \sum_{i=1}^{p-1}\Phi_i^*\Delta\mathbf{y}_{t-i} + \sum_{i=1}^{q}\Theta_i\mathbf{e}_{t-i}$$

conditional on the presample $\{y_0, \ldots, y_{1-p}\}$, and $e_s = 0, s \leq 0$.

You can specify a VEC-ARMA(2,1) model with cointegration rank 2 on the three-dimensional time series by the following statements:

```
model y1-y3 / p=2 q=1;
cointeg rank=2;
```

For more information about modeling the cointegrated VARMA processes, see Lütkepohl (2007, Chapter 14).

Multivariate GARCH Modeling

Stochastic volatility modeling is important in many areas, particularly in finance. To study the volatility of time series, GARCH models are widely used because they provide a good approach to conditional variance modeling.

BEKK Representation

Engle and Kroner (1995) propose a general multivariate GARCH model and call it a BEKK representation. Let $\mathcal{F}(t-1)$ be the sigma field generated by the past values of $\boldsymbol{\epsilon}_t$, and let H_t be the conditional covariance matrix of the k-dimensional random vector $\boldsymbol{\epsilon}_t$. Let H_t be measurable with respect to $\mathcal{F}(t-1)$; then the multivariate GARCH model can be written as

$$\begin{aligned}\boldsymbol{\epsilon}_t|\mathcal{F}(t-1) &\sim N(0, H_t)\\ H_t &= C + \sum_{i=1}^{q}A_i'\boldsymbol{\epsilon}_{t-i}\boldsymbol{\epsilon}_{t-i}'A_i + \sum_{i=1}^{p}G_i'H_{t-i}G_i\end{aligned}$$

where C, A_i, and G_i are $k \times k$ parameter matrices.

Consider the bivariate GARCH(1,1) model

$$\begin{aligned}H_t = &\begin{bmatrix} c_{11} & c_{12}\\ c_{12} & c_{22}\end{bmatrix} + \begin{bmatrix} a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}'\begin{bmatrix} \epsilon_{1,t-1}^2 & \epsilon_{1,t-1}\epsilon_{2,t-1}\\ \epsilon_{2,t-1}\epsilon_{1,t-1} & \epsilon_{2,t-1}^2\end{bmatrix}\begin{bmatrix} a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\\ &+\begin{bmatrix} g_{11} & g_{12}\\ g_{21} & g_{22}\end{bmatrix}'H_{t-1}\begin{bmatrix} g_{11} & g_{12}\\ g_{21} & g_{22}\end{bmatrix}\end{aligned}$$

or, representing the univariate model,

$$
\begin{aligned}
h_{11,t} &= c_{11} + a_{11}^2\epsilon_{1,t-1}^2 + 2a_{11}a_{21}\epsilon_{1,t-1}\epsilon_{2,t-1} + a_{21}^2\epsilon_{2,t-1}^2 \\
&\quad + g_{11}^2 h_{11,t-1} + 2g_{11}g_{21}h_{12,t-1} + g_{21}^2 h_{22,t-1} \\
h_{12,t} &= c_{12} + a_{11}a_{12}\epsilon_{1,t-1}^2 + (a_{21}a_{12} + a_{11}a_{22})\epsilon_{1,t-1}\epsilon_{2,t-1} + a_{21}a_{22}\epsilon_{2,t-1}^2 \\
&\quad + g_{11}g_{12}h_{11,t-1} + (g_{21}g_{12} + g_{11}g_{22})h_{12,t-1} + g_{21}g_{22}h_{22,t-1} \\
h_{22,t} &= c_{22} + a_{12}^2\epsilon_{1,t-1}^2 + 2a_{12}a_{22}\epsilon_{1,t-1}\epsilon_{2,t-1} + a_{22}^2\epsilon_{2,t-1}^2 \\
&\quad + g_{12}^2 h_{11,t-1} + 2g_{12}g_{22}h_{12,t-1} + g_{22}^2 h_{22,t-1}
\end{aligned}
$$

For the BEKK representation of the bivariate GARCH(1,1) model, the SAS statements are

```
model y1 y2;
garch q=1 p=1 form=bekk;
```

The multistep forecast of the conditional covariance matrix, $H_{t+h|t}, h = 1, 2, \ldots,$ is obtained recursively through the formula

$$
H_{t+h|t} = C + \sum_{i=1}^{h-1} A_i' H_{t+h-i|t} A_i + \sum_{i=h}^{q} A_i' \boldsymbol{\epsilon}_{t+h-i} \boldsymbol{\epsilon}_{t+h-i}' A_i + \sum_{i=1}^{p} G_i' H_{t+h-i|t} G_i
$$

where $H_{s|t} = H_s$ for $s \leq t$.

CCC Representation

Bollerslev (1990) proposes a multivariate GARCH model with time-varying conditional variances and covariances but constant conditional correlations.

The conditional covariance matrix H_t consists of

$$
H_t = D_t S D_t
$$

where D_t is a $k \times k$ stochastic diagonal matrix with element $\sigma_{i,t}$ and S is a $k \times k$ time-invariant correlation matrix with the typical element s_{ij}.

The element of H_t is

$$
h_{ij,t} = s_{ij}\sigma_{i,t}\sigma_{j,t} \quad i, j = 1, \ldots, k
$$

Note that $h_{ii,t} = \sigma_{i,t}^2, i = 1, \ldots, k$.

If you specify CORRCONSTANT=EXPECT, the element s_{ij} of the time-invariant correlation matrix S is

$$
s_{ij} = \frac{1}{T} \sum_{t=1}^{T} \frac{\boldsymbol{\epsilon}_{i,t}}{\sqrt{h_{ii,t}}} \frac{\boldsymbol{\epsilon}_{j,t}}{\sqrt{h_{jj,t}}}
$$

where T is the sample size.

By default, or when you specify SUBFORM=GARCH, $\sigma^2_{i,t}$ follows a univariate GARCH process,

$$\sigma^2_{i,t} = c_i + \sum_{l=1}^{q} a_{ii,l}\epsilon^2_{i,t-l} + \sum_{l=1}^{p} g_{ii,l}\sigma^2_{i,t-1} \quad i = 1,\ldots,k$$

As shown in many empirical studies, positive and negative innovations have different impacts on future volatility. There is a long list of variations of univariate GARCH models that consider the asymmetricity. Four typical variations follow:

- exponential GARCH (EGARCH) model (Nelson and Cao 1992)
- quadratic GARCH (QGARCH) model (Engle and Ng 1993)
- threshold GARCH (TGARCH) model (Glosten, Jaganathan, and Runkle 1993; Zakoian 1994)
- power GARCH (PGARCH) model (Ding, Granger, and Engle 1993)

For more information about the asymmetric GARCH models, see Engle and Ng (1993). You can choose the type of GARCH model of interest by specifying the SUBFORM= option.

The EGARCH model was proposed by Nelson (1991). Nelson and Cao (1992) argue that the nonnegativity constraints in the GARCH model are too restrictive. The GARCH model, implicitly or explicitly, imposes the nonnegative constraints on the parameters, whereas these parameters have no restrictions in the EGARCH model. In the EGARCH model, the conditional variance is an asymmetric function of lagged disturbances,

$$\ln(\sigma^2_{i,t}) = c_i + \sum_{l=1}^{q} a_{ii,l}\left(b_{ii,l}\frac{\epsilon_{i,t-l}}{\sigma_{i,t-l}} + |\frac{\epsilon_{i,t-l}}{\sigma_{i,t-l}}| - \sqrt{\frac{2}{\pi}}\right) + \sum_{l=1}^{p} g_{ii,l}\ln(\sigma^2_{i,t-l}) \quad i = 1,\ldots,k$$

In the QGARCH model, the lagged errors' centers are shifted from zero to some constant values,

$$\sigma^2_{i,t} = c_i + \sum_{l=1}^{q} a_{ii,l}(\epsilon_{i,t-l} - b_{ii,l})^2 + \sum_{l=1}^{p} g_{ii,l}\sigma^2_{i,t-1} \quad i = 1,\ldots,k$$

In the TGARCH model, each lagged squared error has an extra slope coefficient,

$$\sigma^2_{i,t} = c_i + \sum_{l=1}^{q}(a_{ii,l} + 1_{\epsilon_{i,t-l}<0}b_{ii,l})\epsilon^2_{i,t-l} + \sum_{l=1}^{p} g_{ii,l}\sigma^2_{i,t-1} \quad i = 1,\ldots,k$$

where the indicator function $1_{\epsilon_{i,t}<0}$ is one if $\epsilon_{i,t} < 0$ and zero otherwise.

The PGARCH model not only considers the asymmetric effect but also provides a way to model the long memory property in the volatility,

$$\sigma_{i,t}^{2\lambda_i} = c_i + \sum_{l=1}^{q} a_{ii,l}(|\epsilon_{i,t-l}| - b_{ii,l}\epsilon_{i,t-l})^{2\lambda_i} + \sum_{l=1}^{p} g_{ii,l}\sigma_{i,t-1}^{2\lambda_i} \quad i = 1, \ldots, k$$

where $\lambda_i > 0$ and $|b_{ii,l}| \leq 1, l = 1, \ldots, q, i = 1, \ldots, k$.

Note that the implemented TGARCH model is also well known as GJR-GARCH (Glosten, Jaganathan, and Runkle 1993), which is similar to the threshold GARCH model proposed by Zakoian (1994) but not exactly the same. In Zakoian's model, the conditional standard deviation is a linear function of the past values of the white noise. Zakoian's model can be regarded as a special case of the PGARCH model when $\lambda_i = 1/2$.

The following formulas are recursively implemented to obtain the multistep forecast of conditional error variance $\sigma_{i,t+h|t}^2, i = 1, \ldots, k$ and $h = 1, 2, \ldots$:

- for the GARCH(p, q) model:

$$\sigma_{i,t+h|t}^2 = c_i + \sum_{l=1}^{h-1} a_{ii,l}\sigma_{i,t+h-l|t}^2 + \sum_{l=h}^{q} a_{ii,l}\epsilon_{i,t+h-l}^2 + \sum_{l=1}^{p} g_{ii,l}\sigma_{i,t+h-1|t}^2$$

- for the EGARCH(p, q) model:

$$\ln(\sigma_{i,t+h|t}^2) = c_i + \sum_{l=h}^{q} a_{ii,l}\left(b_{ii,l}\frac{\epsilon_{i,t+h-l}}{\sigma_{i,t+h-l}} + |\frac{\epsilon_{i,t+h-l}}{\sigma_{i,t+h-l}}| - \sqrt{\frac{2}{\pi}}\right) + \sum_{l=1}^{p} g_{ii,l}\ln(\sigma_{i,t+h-l|t}^2)$$

- for the QGARCH(p, q) model:

$$\begin{aligned}\sigma_{i,t+h|t}^2 = {} & c_i + \sum_{l=1}^{h-1} a_{ii,l}(\sigma_{i,t+h-l|t}^2 + b_{ii,l}^2) + \sum_{l=h}^{q} a_{ii,l}(\epsilon_{i,t+h-l} - b_{ii,l})^2 \\ & + \sum_{l=1}^{p} g_{ii,l}\sigma_{i,t+h-1|t}^2\end{aligned}$$

- for the TGARCH(p, q) model:

$$\begin{aligned}\sigma_{i,t+h|t}^2 = {} & c_i + \sum_{l=1}^{h-1}(a_{ii,l} + b_{ii,l}/2)\sigma_{i,t+h-1|t}^2 + \sum_{l=h}^{q}(a_{ii,l} + 1_{\epsilon_{i,t-l}<0}b_{ii,l})\epsilon_{i,t-l}^2 \\ & + \sum_{l=1}^{p} g_{ii,l}\sigma_{i,t+h-1|t}^2\end{aligned}$$

- for the PGARCH(p, q) model:

$$\begin{aligned}\sigma_{i,t+h|t}^{2\lambda_i} &= c_i + \sum_{l=1}^{h-1} a_{ii,l}((1+b_{ii,l})^{2\lambda_i} + (1-b_{ii,l})^{2\lambda_i})\sigma_{i,t+h-l|t}^{2\lambda_i}/2 \\ &\quad + \sum_{l=h}^{q} a_{ii,l}(|\epsilon_{i,t-l}| - b_{ii,l}\epsilon_{i,t-l})^{2\lambda_i} + \sum_{l=1}^{p} g_{ii,l}\sigma_{i,t+h-1|t}^{2\lambda_i}\end{aligned}$$

In the preceding equations, $\sigma_{i,s|t} = \sigma_{i,s}$ for $s \leq t$. Then, the multistep forecast of conditional covariance matrix $H_{t+h|t}, h = 1, 2, \ldots$, is calculated by

$$H_{t+h|t} = D_{t+h|t} S D_{t+h|t}$$

where $D_{t+h|t}$ is the diagonal matrix with element $\sigma_{i,t+h|t}, i = 1, \ldots, k$.

DCC Representation

Engle (2002) proposes a parsimonious parametric multivariate GARCH model that has time-varying conditional covariances and correlations.

The conditional covariance matrix H_t consists of

$$H_t = D_t \Gamma_t D_t$$

where D_t is a $k \times k$ stochastic diagonal matrix with the element $\sigma_{i,t}$ and Γ_t is a $k \times k$ time-varying matrix with the typical element $\rho_{ij,t}$.

The element of H_t is

$$h_{ij,t} = \rho_{ij,t}\sigma_{i,t}\sigma_{j,t} \quad i, j = 1, \ldots, k$$

Note that $h_{ii,t} = \sigma_{i,t}^2, i = 1, \ldots, k$.

As in the CCC GARCH model, you can choose the type of GARCH model of interest by specifying the SUBFORM= option.

In the GARCH model,

$$\sigma_{i,t}^2 = c_i + \sum_{l=1}^{q} a_{ii,l}\epsilon_{i,t-l}^2 + \sum_{l=1}^{p} g_{ii,l}\sigma_{i,t-1}^2 \quad i = 1, \ldots, k$$

In the EGARCH model, the conditional variance is an asymmetric function of lagged disturbances,

$$\ln(\sigma_{i,t}^2) = c_i + \sum_{l=1}^{q} a_{ii,l}\left(b_{ii,l}\frac{\epsilon_{i,t-l}}{\sigma_{i,t-l}} + |\frac{\epsilon_{i,t-l}}{\sigma_{i,t-l}}| - \sqrt{\frac{2}{\pi}}\right) + \sum_{l=1}^{p} g_{ii,l}\ln(\sigma_{i,t-l}^2) \quad i = 1, \ldots, k$$

In the QGARCH model, the lagged errors' centers are shifted from zero to some constant values,

$$\sigma_{i,t}^2 = c_i + \sum_{l=1}^{q} a_{ii,l}(\epsilon_{i,t-l} - b_{ii,l})^2 + \sum_{l=1}^{p} g_{ii,l}\sigma_{i,t-1}^2 \quad i = 1, \ldots, k$$

In the TGARCH model, each lagged squared error has an extra slope coefficient,

$$\sigma_{i,t}^2 = c_i + \sum_{l=1}^{q} (a_{ii,l} + 1_{\epsilon_{i,t-l}<0} b_{ii,l})\epsilon_{i,t-l}^2 + \sum_{l=1}^{p} g_{ii,l}\sigma_{i,t-1}^2 \quad i = 1, \ldots, k$$

where the indicator function $1_{\epsilon_{i,t}<0}$ is one if $\epsilon_{i,t} < 0$ and zero otherwise.

The PGARCH model not only considers the asymmetric effect but also provides another way to model the long memory property in the volatility,

$$\sigma_{i,t}^{2\lambda_i} = c_i + \sum_{l=1}^{q} a_{ii,l}(|\epsilon_{i,t-l}| - b_{ii,l}\epsilon_{i,t-l})^{2\lambda_i} + \sum_{l=1}^{p} g_{ii,l}\sigma_{i,t-1}^{2\lambda_i} \quad i = 1, \ldots, k$$

where $\lambda_i > 0$ and $|b_{ii,l}| \leq 1, l = 1, \ldots, q; i = 1, \ldots, k$.

The conditional correlation estimator $\rho_{ij,t}$ is

$$\begin{aligned} \rho_{ij,t} &= \frac{q_{ij,t}}{\sqrt{q_{ii,t}q_{jj,t}}} \quad i, j = 1, \ldots, k \\ q_{ij,t} &= (1 - \alpha - \beta)s_{ij} + \alpha \frac{\epsilon_{i,t-1}}{\sigma_{i,t-1}} \frac{\epsilon_{j,t-1}}{\sigma_{j,t-1}} + \beta q_{ij,t-1} \end{aligned}$$

where s_{ij} is the element of S, the unconditional correlation matrix.

If you specify CORRCONSTANT=EXPECT, the element s_{ij} of the unconditional correlation matrix S is

$$s_{ij} = \frac{1}{T} \sum_{t=1}^{T} \frac{\epsilon_{i,t}}{\sigma_{i,t}} \frac{\epsilon_{j,t}}{\sigma_{j,t}}$$

where T is the sample size.

As shown in the CCC GARCH models, the following formulas are recursively implemented to obtain the multistep forecast of conditional error variance $\sigma_{i,t+h|t}^2, i = 1, \ldots, k$ and $h = 1, 2, \ldots$:

- for the GARCH(p, q) model:

$$\sigma_{i,t+h|t}^2 = c_i + \sum_{l=1}^{h-1} a_{ii,l}\sigma_{i,t+h-l|t}^2 + \sum_{l=h}^{q} a_{ii,l}\epsilon_{i,t+h-l}^2 + \sum_{l=1}^{p} g_{ii,l}\sigma_{i,t+h-1|t}^2$$

- for the EGARCH(p, q) model:

$$\ln(\sigma^2_{i,t+h|t}) = c_i + \sum_{l=h}^{q} a_{ii,l}\left(b_{ii,l}\frac{\epsilon_{i,t+h-l}}{\sigma_{i,t+h-l}} + |\frac{\epsilon_{i,t+h-l}}{\sigma_{i,t+h-l}}| - \sqrt{\frac{2}{\pi}}\right) + \sum_{l=1}^{p} g_{ii,l}\ln(\sigma^2_{i,t+h-l|t})$$

- for the QGARCH(p, q) model:

$$\begin{aligned}\sigma^2_{i,t+h|t} &= c_i + \sum_{l=1}^{h-1} a_{ii,l}(\sigma^2_{i,t+h-l|t} + b^2_{ii,l}) + \sum_{l=h}^{q} a_{ii,l}(\epsilon_{i,t+h-l} - b_{ii,l})^2 \\ &\quad + \sum_{l=1}^{p} g_{ii,l}\sigma^2_{i,t+h-1|t}\end{aligned}$$

- for the TGARCH(p, q) model:

$$\begin{aligned}\sigma^2_{i,t+h|t} &= c_i + \sum_{l=1}^{h-1}(a_{ii,l} + b_{ii,l}/2)\sigma^2_{i,t+h-1|t} + \sum_{l=h}^{q}(a_{ii,l} + 1_{\epsilon_{i,t-l}<0}b_{ii,l})\epsilon^2_{i,t-l} \\ &\quad + \sum_{l=1}^{p} g_{ii,l}\sigma^2_{i,t+h-1|t}\end{aligned}$$

- for the PGARCH(p, q) model:

$$\begin{aligned}\sigma^{2\lambda_i}_{i,t+h|t} &= c_i + \sum_{l=1}^{h-1} a_{ii,l}((1 + b_{ii,l})^{2\lambda_i} + (1 - b_{ii,l})^{2\lambda_i})\sigma^{2\lambda_i}_{i,t+h-l|t}/2 \\ &\quad + \sum_{l=h}^{q} a_{ii,l}(|\epsilon_{i,t-l}| - b_{ii,l}\epsilon_{i,t-l})^{2\lambda_i} + \sum_{l=1}^{p} g_{ii,l}\sigma^{2\lambda_i}_{i,t+h-1|t}\end{aligned}$$

In the preceding equations, $\sigma_{i,s|t} = \sigma_{i,s}$ for $s \le t$. Then, the multistep forecast of conditional covariance matrix $H_{t+h|t}, h = 1, 2, \ldots$, is calculated by

$$H_{t+h|t} = D_{t+h|t}\Gamma_{t+h|t}D_{t+h|t}$$

where $D_{t+h|t}$ is the diagonal matrix with element $\sigma_{i,t+h|t}, i = 1, \ldots, k$, and $\Gamma_{t+h|t}$ is the matrix with element $\rho_{ij,t+h|t}, i, j = 1, \ldots, k$,

$$\begin{aligned}\rho_{ij,t+h|t} &= \frac{q_{ij,t+h|t}}{\sqrt{q_{ii,t+h|t}q_{jj,t+h|t}}} \\ q_{ij,t+h|t} &= \begin{cases}(1-\alpha-\beta)s_{ij} + \alpha\frac{\epsilon_{i,t}}{\sigma_{i,t}}\frac{\epsilon_{j,t}}{\sigma_{j,t}} + \beta q_{ij,t} & h = 1 \\ (1-\alpha-\beta)s_{ij} + \alpha q_{ij,t+h-1|t} + \beta q_{ij,t+h-1|t} & h > 1\end{cases}\end{aligned}$$

Estimation of GARCH Model

The log-likelihood function of the multivariate GARCH model is written without a constant term as

$$\ell = -\frac{1}{2}\sum_{t=1}^{T}[\log|H_t| + \boldsymbol{\epsilon}_t' H_t^{-1}\boldsymbol{\epsilon}_t]$$

where $\boldsymbol{\epsilon}_t$ is calculated from the first-moment model (that is, the VARMAX model or VEC-ARMA model). The log-likelihood function is maximized by an iterative numerical method such as quasi-Newton optimization. The starting values for the regression parameters are obtained from the least squares estimates. The covariance of $\boldsymbol{\epsilon}_t$ is used as the starting value for the GARCH constant parameters, and the starting values for the other GARCH parameters are either 10^{-6} or 10^{-3}, depending on the GARCH model's representation.

Prediction of Endogenous (Dependent) Variables

In multivariate GARCH models, the optimal (minimum MSE) l-step-ahead forecast of endogenous variables $\mathbf{y}_{t+l|t}$ uses the same formula as shown in the section "Forecasting" on page 3057. However, the exogenous (independent) variables, if present, are always assumed to be nonstochastic (deterministic); that is, to predict the endogenous variables, you must specify the future values of the exogenous variables. The prediction error of the optimal l-step-ahead forecast is $\mathbf{e}_{t+l|t} = \mathbf{y}_{t+l} - \mathbf{y}_{t+l|t} = \sum_{j=0}^{l-1}\Psi_j\boldsymbol{\epsilon}_{t+l-j}$, with zero mean and covariance matrix,

$$\Sigma_t(l) = \mathrm{Cov}(\mathbf{e}_{t+l|t}) = \sum_{j=0}^{l-1}\Psi_j H_{t+l-j|t}\Psi_j'$$

where $H_{t+h|t}, h = 1,\ldots,l$, is the h-step-ahead forecast of the conditional covariance matrix. As emphasized by the subscript t, $\Sigma_t(l)$ is time-dependent. In the OUT= data set, the forecast standard errors and prediction intervals are constructed according to $\Sigma_t(l)$. If you specify the COVPE option, the prediction error covariances that are output in the CovPredictError and CovPredictErrorbyVar ODS tables are based on the time-independent formula

$$\Sigma(l) = \sum_{j=0}^{l-1}\Psi_j\Sigma\Psi_j'$$

where Σ is the unconditional covariance matrix of innovations. The decomposition of the prediction error covariances is also based on $\Sigma(l)$.

Covariance Stationarity

Define the multivariate GARCH process as

$$\mathbf{h}_t = \sum_{i=1}^{\infty} G(B)^{i-1}[\mathbf{c} + A(B)\boldsymbol{\eta}_t]$$

where $\mathbf{h}_t = \text{vec}(H_t)$, $\mathbf{c} = \text{vec}(C_0)$, and $\boldsymbol{\eta}_t = \text{vec}(\boldsymbol{\epsilon}_t \boldsymbol{\epsilon}_t')$. This representation is equivalent to a GARCH(p, q) model by the following algebra:

$$\begin{aligned}
\mathbf{h}_t &= \mathbf{c} + A(B)\boldsymbol{\eta}_t + \sum_{i=2}^{\infty} G(B)^{i-1}[\mathbf{c} + A(B)\boldsymbol{\eta}_t] \\
&= \mathbf{c} + A(B)\boldsymbol{\eta}_t + G(B)\sum_{i=1}^{\infty} G(B)^{i-1}[tmbc + A(B)\boldsymbol{\eta}_t] \\
&= \mathbf{c} + A(B)\boldsymbol{\eta}_t + G(B)\mathbf{h}_t
\end{aligned}$$

Defining $A(B) = \sum_{i=1}^{q}(A_i \otimes A_i)'B^i$ and $G(B) = \sum_{i=1}^{p}(G_i \otimes G_i)'B^i$ gives a BEKK representation.

The necessary and sufficient conditions for covariance stationarity of the multivariate GARCH process are that all the eigenvalues of $A(1) + G(1)$ are less than 1 in modulus.

An Example of a VAR(1)–ARCH(1) Model

The following DATA step simulates a bivariate vector time series to provide test data for the multivariate GARCH model:

```
data garch;
   retain seed 16587;
   esq1 = 0; esq2 = 0;
   ly1 = 0;  ly2 = 0;
   do i = 1 to 1000;
      ht = 6.25 + 0.5*esq1;
      call rannor(seed,ehat);
      e1 = sqrt(ht)*ehat;
      ht = 1.25 + 0.7*esq2;
      call rannor(seed,ehat);
      e2 = sqrt(ht)*ehat;
      y1 = 2 + 1.2*ly1 - 0.5*ly2 + e1;
      y2 = 4 + 0.6*ly1 + 0.3*ly2 + e2;
      if i>500 then output;
      esq1 = e1*e1; esq2 = e2*e2;
      ly1 = y1;  ly2 = y2;
   end;
   keep y1 y2;
run;
```

The following statements fit a VAR(1)–ARCH(1) model to the data. For a VAR-ARCH model, you specify the order of the autoregressive model with the P=1 option in the MODEL statement and the Q=1 option in the GARCH statement. In order to produce the initial and final values of parameters, the TECH=QN option is specified in the NLOPTIONS statement.

```
proc varmax data=garch;
   model y1 y2 / p=1
         print=(roots estimates diagnose);
   garch q=1;
   nloptions tech=qn;
run;
```

Figure 42.81 through Figure 42.85 show the details of this example. Figure 42.81 shows the initial values of parameters.

Figure 42.81 Start Parameter Estimates for the VAR(1)–ARCH(1) Model

The VARMAX Procedure

Optimization Start

Parameter Estimates

N	Parameter	Estimate	Gradient Objective Function
1	CONST1	2.249575	0.000082533
2	CONST2	3.902673	0.000401
3	AR1_1_1	1.231775	0.000105
4	AR1_2_1	0.576890	-0.004811
5	AR1_1_2	-0.528405	0.000617
6	AR1_2_2	0.343714	0.001811
7	GCHC1_1	9.929763	0.151293
8	GCHC1_2	0.193163	-0.014305
9	GCHC2_2	4.063245	0.370333
10	ACH1_1_1	0.001000	-0.667182
11	ACH1_2_1	0	-0.068905
12	ACH1_1_2	0	-0.734486
13	ACH1_2_2	0.001000	-3.127035

Figure 42.82 shows the final parameter estimates.

Figure 42.82 Results of Parameter Estimates for the VAR(1)–ARCH(1) Model

The VARMAX Procedure

Optimization Results

Parameter Estimates

N	Parameter	Estimate	Gradient Objective Function
1	CONST1	2.156865	0.000246
2	CONST2	4.048879	0.000105
3	AR1_1_1	1.224620	-0.001957
4	AR1_2_1	0.609651	0.000173
5	AR1_1_2	-0.534248	-0.000468
6	AR1_2_2	0.302599	-0.000375
7	GCHC1_1	8.238625	-0.000056090
8	GCHC1_2	-0.231183	-0.000021724
9	GCHC2_2	1.565459	0.000110
10	ACH1_1_1	0.374255	-0.000419
11	ACH1_2_1	0.035883	-0.000606
12	ACH1_1_2	0.057461	0.001636
13	ACH1_2_2	0.717897	-0.000149

Figure 42.83 shows the conditional variance by using the BEKK representation of the ARCH(1) model. The ARCH parameters are estimated as follows by the vectorized parameter matrices:

$$\begin{aligned}
\boldsymbol{\epsilon}_t|\mathcal{F}(t-1) &\sim N(0, H_t) \\
H_t &= \begin{bmatrix} 8.23863 & -0.23118 \\ -0.23118 & 1.56546 \end{bmatrix} \\
&+ \begin{bmatrix} 0.37426 & 0.05746 \\ 0.03588 & 0.71790 \end{bmatrix}' \boldsymbol{\epsilon}_{t-1}\boldsymbol{\epsilon}_{t-1}' \begin{bmatrix} 0.37426 & 0.05746 \\ 0.03588 & 0.71790 \end{bmatrix}
\end{aligned}$$

Figure 42.83 ARCH(1) Parameter Estimates for the VAR(1)–ARCH(1) Model

The VARMAX Procedure

Type of Model	VAR(1)-ARCH(1)
Estimation Method	Maximum Likelihood Estimation
Representation Type	BEKK

GARCH Model Parameter Estimates

Parameter	Estimate	Standard Error	t Value	Pr > |t|
GCHC1_1	8.23863	0.72663	11.34	0.0001
GCHC1_2	-0.23118	0.21434	-1.08	0.2813
GCHC2_2	1.56546	0.19407	8.07	0.0001
ACH1_1_1	0.37426	0.07502	4.99	0.0001
ACH1_2_1	0.03588	0.06974	0.51	0.6071
ACH1_1_2	0.05746	0.02597	2.21	0.0274
ACH1_2_2	0.71790	0.06895	10.41	0.0001

Figure 42.84 shows the AR parameter estimates and their significance.

The fitted VAR(1) model with the previous conditional covariance ARCH model is written as follows:

$$\mathbf{y}_t = \begin{bmatrix} 2.15687 \\ 4.04888 \end{bmatrix} + \begin{bmatrix} 1.22462 & -0.53425 \\ 0.60965 & 0.30260 \end{bmatrix} \mathbf{y}_{t-1} + \boldsymbol{\epsilon}_t$$

Figure 42.84 VAR(1) Parameter Estimates for the VAR(1)–ARCH(1) Model

Model Parameter Estimates

Equation	Parameter	Estimate	Standard Error	t Value	Pr > |t|	Variable
y1	**CONST1**	2.15687	0.21717	9.93	0.0001	1
	AR1_1_1	1.22462	0.02542	48.17	0.0001	y1(t-1)
	AR1_1_2	-0.53425	0.02807	-19.03	0.0001	y2(t-1)
y2	**CONST2**	4.04888	0.10663	37.97	0.0001	1
	AR1_2_1	0.60965	0.01216	50.13	0.0001	y1(t-1)
	AR1_2_2	0.30260	0.01491	20.30	0.0001	y2(t-1)

Figure 42.85 shows the roots of the AR and ARCH characteristic polynomials. The eigenvalues have a modulus less than one.

Figure 42.85 Roots for the VAR(1)–ARCH(1) Model

Roots of AR Characteristic Polynomial					
Index	Real	Imaginary	Modulus	Radian	Degree
1	0.76361	0.33641	0.8344	0.4150	23.7762
2	0.76361	-0.33641	0.8344	-0.4150	-23.7762

Roots of GARCH Characteristic Polynomial					
Index	Real	Imaginary	Modulus	Radian	Degree
1	0.52388	0.00000	0.5239	0.0000	0.0000
2	0.26661	0.00000	0.2666	0.0000	0.0000
3	0.26661	0.00000	0.2666	0.0000	0.0000
4	0.13569	0.00000	0.1357	0.0000	0.0000

VARFIMA and VARFIMAX Modeling

VAR and VARMA series are short-range dependent (SRD) in the sense that their autocovariance function dies out exponentially fast with the increasing lag. However, in many financial and macroeconomics applications, stationary yet persistent series arise, calling for models that have a slowly decaying autocovariance function and that are therefore more suitable to capture long-range dependence in the data.

The VARFIMA model captures both long-range and short-range dependence dynamics in a multivariate series. For a k-dimensional series $\mathbf{y}_t = (y_{1t}, \ldots, y_{kt})'$, $t = 1, \ldots, T$, the VARFIMA(p, D, q) model is defined as

$$\Phi(B)\mathbf{y}_t = (I - B)^{-D}\Theta(B)\boldsymbol{\epsilon}_t$$

where B and I are the backshift and identity operators; $D = \mathrm{diag}(d_j)$ $d_j \in (-1/2, 1/2)$, are the LRD parameters of the component series $\{y_{jt}\}_{t\in\mathrm{Z}}$, $j = 1, \ldots, k$; and $\{\boldsymbol{\epsilon}_t\}_{t\in\mathrm{Z}}$ is a k-dimensional white noise series with zero mean $\mathrm{E}\boldsymbol{\epsilon}_t = 0$ and covariance $\mathrm{E}\boldsymbol{\epsilon}_t\boldsymbol{\epsilon}_t' = \Sigma$.

The fractional integration operator $(I - B)^{-D}$ allows for long memory in the series. On the other hand, $\Phi(z)$ and $\Theta(z)$, which are the typical autoregressive and moving average matrix polynomials of orders p and q, respectively, capture the short-range dependence.

The VARFIMA(p, D, q) series satisfies the multivariate long-range dependence definitions given in Kechagias and Pipiras (2015). Moreover, each component series $\{y_{jt}\}_{t\in\mathrm{Z}}$, $j = 1, \ldots, k$, satisfies the univariate time and frequency domain LRD definitions given in Beran et al. (2013). The following sections briefly review these definitions and show how you can detect long-range dependence in the data before fitting a VARFIMA model.

Autocorrelation and Spectral Density of VARFIMA Series

The diagonal components of the autocorrelation matrix function of a VARFIMA(p, D, q) series satisfy the univariate LRD time domain definition

$$\rho_i(n) \sim c_1 n^{2d_i - 1},\ i = 1, \ldots, k, \quad \text{as} \quad n \to \infty$$

where $a_n \sim b_n$ implies that $\lim_{n\to\infty} a_n/b_n = 1$ and $c_1 > 0$. Similarly, the diagonal components of the spectral density matrix function of a VARFIMA(p, D, q) series satisfy

$$f_i(\lambda) \sim c_2\lambda^{-2d_i},\ i = 1, \ldots, k, \quad \text{as} \quad \lambda \to 0^+$$

for some $c_2 > 0$.

To obtain preliminary estimates of the LRD parameters, you can plot the logged periodogram values against the log of the Fourier frequencies $\lambda_j = 2\pi j/T$, $j = 1, \ldots, T/2$, and then fit a line for frequencies near 0. The slope of this line is expected to be equal to $-2d_i$ (the exponent in the right-hand side of the preceding relation). The following statements demonstrate this procedure for a synthetic VARFIMA$(1, D, 1)$ series with $T = 2{,}000$ and true parameters $d_1 = 0.4$, $d_2 = 0.3$, $\Phi_{11} = \Sigma_{11} = \Sigma_{22} = 3$, $\Sigma_{12} = 0.5$, $\Phi_{11} = 0.8$, $\Phi_{12} = 0.3$, $\Phi_{21} = -0.2$, $\Phi_{22} = 0.1$, $\Theta_{11} = 0.2$, $\Theta_{12} = 0.4$, $\Theta_{21} = 0$, and $\Theta_{22} = 0.3$:

```
data VARFIMA1D1;
   time = _N_;
   input y1 y2;
datalines;
1.495250048 2.694910375
4.503081454 1.42319642

   ... more lines ...

3.12049851 5.330308391
7.732287586 1.665071247
;

/* Compute the two periodograms */
proc spectra data = VARFIMA1D1 out = spectra;
   var y1 y2;
run;

/* Convert to log scale */
data logspectra;
   set spectra(firstobs=2);
   /* compute Fourier frequencies */
   j = _N_;
   pi = constant('pi');
   logfreq = log(2*pi*j/2000);

   logpdg1 = log(P_01);
   logpdg2 = log(P_02);

   /* Introduce weights where regression will be performed */
   wt = (1<= j <=100);
   keep wt logfreq logpdg1 logpdg2;
run;

/* Regression for log-periodogram of y1*/
proc autoreg data = logspectra(obs = 100);
   model logpdg1 = logfreq;
run;
```

```
/* Regression for log-periodogram of y1*/
proc autoreg data = logspectra(obs = 100);
         model logpdg2 = logfreq;
run;
```

The output from the two regressions is shown in Figure 42.86 and Figure 42.87.

Figure 42.86 Regression Estimates for y1

The AUTOREG Procedure

Parameter Estimates

Variable	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
Intercept	1	4.3279	0.2885	15.00	<.0001
logfreq	1	-0.9051	0.1245	-7.27	<.0001

Figure 42.87 Regression Estimates for y2

The AUTOREG Procedure

Parameter Estimates

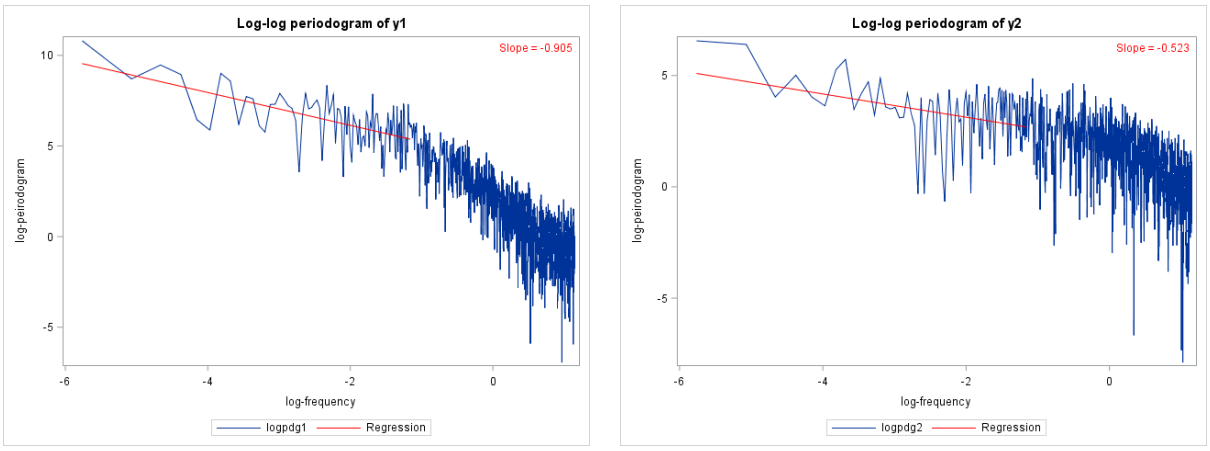
Variable	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
Intercept	1	2.0811	0.3172	6.56	<.0001
logfreq	1	-0.5227	0.1369	-3.82	0.0002

The following statements produce log-log plots of the two periodograms along with the regression lines:

```
/*Plot the periodograms in log-log scale*/
ods graphics on;

proc sgplot data = logspectra;
         series x = logfreq y = logpdg1 / lineattrs = (pattern = solid);
         reg y = logpdg1 x = logfreq / nomarkers weight = wt lineattrs =
                                       (thickness = 1 color = 'red' );
         inset "Slope = -0.905" / position = topright textattrs = (color = 'red');
         xaxis label = 'log-frequency';
         yaxis label = 'log-periodogram';
         title 'Log-periodogram of y1';
run;

proc sgplot data = logspectra;
         series x = logfreq y = logpdg2 / lineattrs = (pattern = solid);
         reg y = logpdg2 x = logfreq / nomarkers weight = wt lineattrs =
                                       (thickness = 1 color = 'red' );
         inset "Slope = -0.523" / position = topright textattrs = (color = 'red');
         xaxis label = 'log-frequency';
         yaxis label = 'log-periodogram';
         title 'Log-periodogram of y2';
run;
```

The final plots are shown in Figure 42.88.

Figure 42.88 Log-Log Periodogram Plots for the Two Series



Dividing the slopes by 2 and removing the negative signs yields preliminary estimates for the LRD parameters, $\widehat{d}_1 = 0.45$ and $\widehat{d}_2 = 0.26$.

Estimation

Estimation of all the parameters in the VARFIMA model is performed using the conditional likelihood Durbin-Levinson (CLDL) algorithm of Tsay (2010). This method uses the multivariate Durbin-Levinson algorithm, whose order of complexity is $O(T^2)$, making it computationally feasible for small or medium sample sizes.

The initial values of the LRD parameters are obtained by the semiparametric estimator of Geweke and Porter-Hudak (1983). The initial values of the AR and MA parameters are obtained from least squares estimation on the fractionally differenced series $(I - B)^D \mathbf{y}_t$. The LRD parameters are restricted in the range $(-1/2, 1/2)$. If an initial LRD parameter estimate is outside this range, then the chosen starting value is either $-1/2 + 10^{-6}$ or $1/2 - 10^{-6}$ for negative or positive initial semiparametric estimates, respectively.

Forecasting

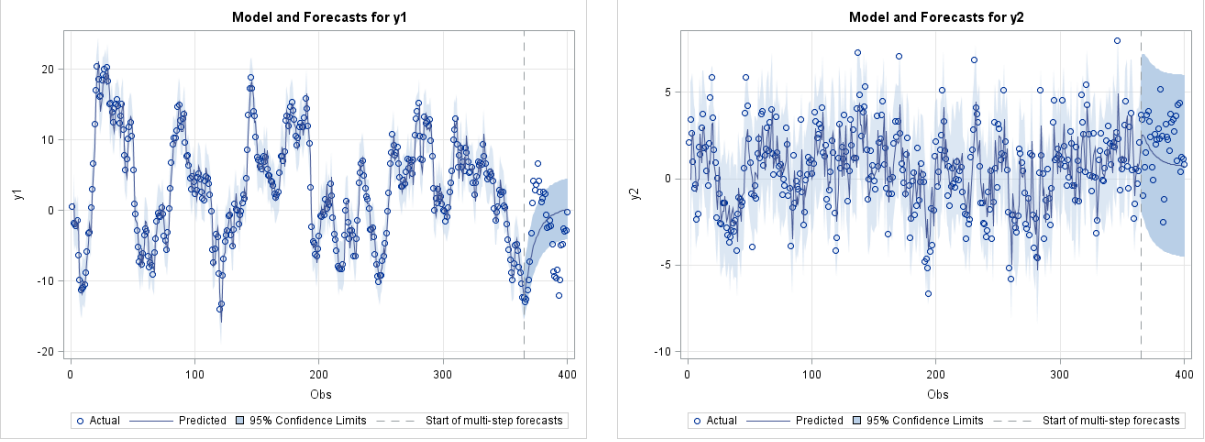
One-step-ahead and multi-step-ahead forecasts for the VARFIMA series are based on a finite past. However, the h-step-ahead forecast errors for $h > 1$ are based on the infinite past except for VARFIMA series that have only MA components. In the latter case, the forecast errors are also based on a finite past.

The following statements plot the h-step-ahead forecasts, $h = 1, \ldots, 36$, for a bivariate synthetic VARFIMA$(1, D, 1)$ series with $T = 400$ and true parameters $d_1 = 0.4$, $d_2 = 0.3$, $\Phi_{11} = \Sigma_{11} = \Sigma_{22} = 3$, $\Sigma_{12} = 0.5$, $\Phi_{11} = 0.8$, $\Phi_{12} = 0.3$, $\Phi_{21} = -0.2$, $\Phi_{22} = 0.1$, $\Theta_{11} = 0.2$, $\Theta_{12} = 0.4$, $\Theta_{21} = 0$, and $\Theta_{22} = 0.3$. The statements also specify initial values for d_1 and d_2 close to the true parameter values.

```
data VARFIMA1D1N4;
   time = _N_;
   input y1 y2;
datalines;
0.55596529 2.114409393
-1.842925215 3.415027987

   ... more lines ...
```

```
-2.86707489 1.147627529
-0.195787414 0.820107072
;

proc varmax data = VARFIMA1D1N4 plots = (forecasts);
         model y1 y2 / noint fi p=1 q=1;
         initial d(1) = 0.45, d(2) = 0.25;
         output out = forec back = 36 lead = 36;
run;
```

Figure 42.89 Plot of the Two Series and h-Step-Ahead Forecasts, $h = 1, \ldots, 36$



The BACK option in the preceding SAS statements is used to specify the point where the historical data ends and multi-step-ahead forecasting begins. Note that the BACK option does not affect estimation. The latter is performed using the whole data set, even when you specify the BACK option.

Impulse Response Functions

The impulse response functions of the VARFIMA series are calculated using the methodology of Chung (2001). The following statements produce the first 12 simple, accumulated and orthogonal impulse response functions and their corresponding standard errors for the VARFIMA$(1, D, 1)$ series of the preceding example.

```
proc varmax data = VARFIMA1D1N4 plots = (impulse);
         model y1 y2 / noint fi p=1 q=1 print = (impulse = (all));
run;
```

VARFIMAX Modeling

The VARFIMAX(p, D, q, s) series is defined as

$$\Phi(B)\mathbf{y}_t + \Theta^*(B)\mathbf{x}_t = (I - B)^{-D}\Theta(B)\boldsymbol{\epsilon}_t$$

where $\mathbf{x}_t = (x_{1t}, \ldots, x_{rt})'$, $t = 1, \ldots, T$, is an r-dimensional time series vector of exogenous variables and $\Theta^*(z)$ is the order s matrix polynomial defined as $\Theta^*(z) = \Theta_0^* + \Theta_1^* z + \cdots + \Theta_s^* z^s$ for some $k \times r$ real matrices Θ_i^*, $i = 1, \ldots, s$.

The following statements estimate a bivariate VARFIMAX$(1, D, 1, 0)$ model:

```
model y1 y2 = x1 / fi p=1 q=1;
```

Conditional Forecasts and Scenario Analysis

Conditional forecasts and scenario analysis have been widely applied in macroeconomics. If you have no knowledge of any future dependent variables or cannot use such information, you can perform only unconditional forecasts. In contrast, conditional forecasts are forecasts conditional on some future paths of dependent variables. Some typical examples of the usage of conditional forecasts and scenario analysis are the stress tests that are conducted by the US Federal Reserve Board in the Comprehensive Capital Analysis and Review (CCAR) and by the European Banking Authority (EBA) on euro area banks that are directly supervised by the European Central Bank (ECB). For more information about conditional forecasts and scenario analysis, see Waggoner and Zha (1999), Karlsson (2013), Bańbura, Giannone, and Lenza (2015), Clark and McCracken (2017), and references therein.

According to Waggoner and Zha (1999), the conditions can be classified into two groups: soft conditions and hard conditions. The soft conditions belong to the set of conditions in which the future values of some dependent variables are restricted within certain ranges. The hard conditions belong to the set of conditions in which the future values of some dependent variables are fixed to some single values.

In order to obtain the conditional forecasts under the soft conditions, you perform unconditional forecasts first, and then select the simulated forecasts that satisfy the soft conditions. For example, for a trivariate VAR model on y1, y2 and y3, two future y3 values are bounded—$y_{3,T+1} \leq 0.10$ and $y_{3,T+2} \geq 0.15$, where T is the in-sample sample size. The following statement performs the unconditional forecasts and outputs the simulated forecasts to the data set oucfsim:

```
condfore outsim=oucfsim;
```

The following statements select the forecasts that satisfy the soft conditions. The forecasts in the data set scForecasts are the conditional forecasts under the soft conditions. You can use the UNIVARIATE procedure or other procedures to get the mean, standard error, or quantiles of any future series of interest.

```
data scForecasts;
   set oucfsim;
   if (y3_1<=0.10 and y3_2>=0.15);
run;
```

You can define the hard conditions in a data set and then use the VARMAX procedure to pick up that data set by specifying the SDATA= option in the CONDFORE statement. For example, for a trivariate VAR model on y1, y2 and y3, two future y3 values are fixed—$y_{3,T+1} = 0.05$ and $y_{3,T+2} = 0.10$, where T is the in-sample sample size. The following statements define the hard conditions (that is, the scenario) in the data set scenario1:

```
data scenario1;
  y1=.; y2=.; y3 = 0.05; output;
  y1=.; y2=.; y3 = 0.10; output;
run;
```

The following statements use the scenario data set and output the (statistics of the simulated) forecasts to data set ocf:

```
condfore sdata=scenario1 out=ocf;
```

In fact, if all future values for a variable are missing, that variable can be omitted; that is, the following statements generate a scenario equivalent to the one in scenario1:

```
data scenario2;
  y3 = 0.05; output;
  y3 = 0.10; output;
run;
```

If there is more than one scenario, you can put the additional scenarios in one data set and distinguish them by using a numeric variable. The following statements define two scenarios and distinguish them with myScenario. In the first scenario (myscenario=1), the future values of y3 are available in two periods: $y_{3,T+1} = 0.05$ and $y_{3,T+2} = 0.10$. In the second scenario (myscenario=2), the future values of y3 are available in four periods: $y_{3,T+1} = 0.05$, $y_{3,T+2} = 0.10$, $y_{3,T+3} = 0.15$, and $y_{3,T+4} = 0.20$.

```
data scenario3;
  y3 = 0.05; myscenario=1; output;
  y3 = 0.10; myscenario=1; output;
  y3 = 0.05; myscenario=2; output;
  y3 = 0.10; myscenario=2; output;
  y3 = 0.15; myscenario=2; output;
  y3 = 0.20; myscenario=2; output;
run;
```

The following statements use the scenarios in the data set scenario3 and output the (statistics of the simulated) forecasts for two scenarios to data set ocf2:

```
condfore sdata=scenario3 sid=myscenario out=ocf2;
```

Future values of exogenous variables can be included in the scenario data set. The following list shows how PROC VARMAX treats various cases of how future values of exogenous variables are provided:

- If you do not include any future values of exogenous variables in the DATA= data set in the PROC VARMAX statement, you must include all future values of all exogenous variables for all forecast horizons for all scenarios in the SDATA= data set in the CONDFORE statement. These values are used in the conditional forecasts.
- If you include future values of exogenous variables for all forecast horizons in the DATA= data set in the PROC VARMAX statement and you do not include any future values of exogenous variables in the SDATA= data set in the CONDFORE statement, the future values of exogenous variables in the DATA= data set in the PROC VARMAX statement are used in the conditional forecasts.
- If you include future values of exogenous variables in both the DATA= data set in the PROC VARMAX statement and the SDATA= data set in the CONDFORE statement, the future values in both data sets are merged. During merging, nonmissing future values in the SDATA= data set in the CONDFORE statement override the corresponding future values in the DATA= data set in the PROC VARMAX statement. The merged future values of exogenous variables for all forecast horizons for each scenario should not contain any missing values because they are used in the conditional forecasts.

Regardless of whether you use the DIF option or the DIFX option on exogenous variables, the future values in the data set that is specified in the SDATA= option in the CONDFORE statement should be the future values of original exogenous variables. However, if you use DIF or DIFY option on a dependent variable, the future values of the correspondingly differenced dependent variable should be included in that data set.

Specifying the OUT= option in the CONDFORE statement creates a data set that contains the statistics of the simulated h-step-ahead forecasts for each dependent variable in each scenario. The following output variables can be created:

- the BY variables
- the ID variable
- STEP, a numeric variable that describes the forecast horizon
- *variable_name*_MEAN, a numeric variable that contains the mean of forecasts for the dependent variable *variable_name*
- *variable_name*_STDERR, a numeric variable that contains the standard error of forecasts for the dependent variable *variable_name*
- *variable_name*_MEDIAN, a numeric variable that contains the median of forecasts for the dependent variable *variable_name*
- *variable_name*_LB, a numeric variable that contains the lower bound of the credible interval of forecasts for the dependent variable *variable_name*
- *variable_name*_UB, a numeric variable that contains the upper bound of the credible interval of the dependent variable *variable_name*
- the SID variable

Specifying the OUTSIM= option in the CONDFORE statement creates a data set that contains the simulated forecasts for each Monte Carlo iteration. The following output variables can be created:

- the BY variables
- SIMID, a numeric variable that contains the index of Monte Carlo iterations
- *variable_name_h*, numeric variable that contains the h-step-ahead forecast for dependent variable *variable_name* in the Monte Carlo iteration The range of h is from 1 to H when you specify LEAD=H in the CONDFORE statement.
- the SID variable

An example that has more details is illustrated in the section "Example 42.5: Conditional Forecasts and Scenario Analysis" on page 3187.

Output Data Sets

The VARMAX procedure can create the OUT=, OUTEST=, OUTHT=, and OUTSTAT= data sets. In general, if processing fails, the output is not recorded or is set to missing in the relevant output data set, and appropriate error and/or warning messages are recorded in the log.

OUT= Data Set

The OUT= data set contains the forecast values that the OUTPUT statement produces. The following output variables can be created:

- the BY variables
- the ID variable
- dependent (endogenous) variables in the MODEL statement. These variables contain the actual values from the input data set.
- FOR*i*, numeric variables that contain the forecasts. The FOR*i* variables contain the forecasts for the *i*th endogenous variable in the MODEL statement list. Forecasts are one-step-ahead predictions until the end of the data or until the observation that is specified in the BACK= option. Multistep forecasts can be computed after that point according to the LEAD= option.
- RES*i*, numeric variables that contain the residual for the forecast of the *i*th endogenous variable in the MODEL statement list. For multistep forecast observations, the actual values are missing and the RES*i* variables contain missing values.
- STD*i*, numeric variables that contain the standard deviation for the forecast of the *i*th endogenous variable in the MODEL statement list. The values of the STD*i* variables can be used to construct univariate confidence limits for the corresponding forecasts.
- LCI*i*, numeric variables that contain the lower confidence limits for the corresponding forecasts of the *i*th endogenous variable in the MODEL statement list
- UCI*i*, numeric variables that contain the upper confidence limits for the corresponding forecasts of the *i*th endogenous variable in the MODEL statement list

The OUT= data set contains the values shown in Table 42.7 and Table 42.8 for a bivariate case.

Table 42.7 OUT= Data Set

Obs	ID Variable	y1	FOR1	RES1	STD1	LCI1	UCI1
1	date	y_{11}	f_{11}	r_{11}	σ_{11}	l_{11}	u_{11}
2	date	y_{12}	f_{12}	r_{12}	σ_{11}	l_{12}	u_{12}
⋮							

Table 42.8 OUT= Data Set Continued

Obs	y2	FOR2	RES2	STD2	LCI2	UCI2
1	y_{21}	f_{21}	r_{21}	σ_{22}	l_{21}	u_{21}
2	y_{22}	f_{22}	r_{22}	σ_{22}	l_{22}	u_{22}
⋮						

Consider the following example:

```
proc varmax data=simul1 noprint;
   id date interval=year;
   model y1 y2 / p=1 noint;
   output out=out lead=5;
run;

proc print data=out(firstobs=98);
run;
```

The output in Figure 42.90 shows part of the results of the OUT= data set for the preceding example.

Figure 42.90 OUT= Data Set

Log-periodogram of y2

Obs	date	y1	FOR1	RES1	STD1	LCI1	UCI1	y2	FOR2	RES2	STD2	LCI2	UCI2
98	1997	-0.58433	-0.13500	-0.44934	1.13523	-2.36001	2.09002	0.64397	-0.34932	0.99329	1.19096	-2.68357	1.98492
99	1998	-2.07170	-1.00649	-1.06522	1.13523	-3.23150	1.21853	0.35925	-0.07132	0.43057	1.19096	-2.40557	2.26292
100	1999	-3.38342	-2.58612	-0.79730	1.13523	-4.81113	-0.36111	-0.64999	-0.99354	0.34355	1.19096	-3.32779	1.34070
101	2000	.	-3.59212	.	1.13523	-5.81713	-1.36711	.	-2.09873	.	1.19096	-4.43298	0.23551
102	2001	.	-3.09448	.	1.70915	-6.44435	0.25539	.	-2.77050	.	1.47666	-5.66469	0.12369
103	2002	.	-2.17433	.	2.14472	-6.37792	2.02925	.	-2.75724	.	1.74212	-6.17173	0.65725
104	2003	.	-1.11395	.	2.43166	-5.87992	3.65203	.	-2.24943	.	2.01925	-6.20709	1.70823
105	2004	.	-0.14342	.	2.58740	-5.21463	4.92779	.	-1.47460	.	2.25169	-5.88782	2.93863

OUTEST= Data Set

The OUTEST= data set contains estimation results of the fitted model produced by the VARMAX statement. The following output variables can be created:

- BY variables
- NAME, a character variable that contains the name of the endogenous (dependent) variables or the name of the parameters for the covariance of the matrix of the parameter estimates if you specify the OUTCOV option
- TYPE, a character variable that contains the value EST for parameter estimates, the value STD for standard error of parameter estimates, and the value COV for the covariance of the matrix of the parameter estimates if you specify the OUTCOV option
- CONST, a numeric variable that contains the estimates of constant parameters and their standard errors
- SEASON_i, a numeric variable that contains the estimates of seasonal dummy parameters and their standard errors, where $i = 1, \ldots, (nseason - 1)$, and $nseason$ is based on the NSEASON= option
- LTREND, a numeric variable that contains the estimates of linear trend parameters and their standard errors
- QTREND, a numeric variable that contains the estimates of quadratic trend parameters and their standard errors

- XL*l*_*i*, numeric variables that contain the estimates of exogenous parameters and their standard errors, where *l* is the lag *l*th coefficient matrix and $i = 1, \ldots, r$, where *r* is the number of exogenous variables

- AR*l*_*i*, numeric variables that contain the estimates of autoregressive parameters and their standard errors, where *l* is the lag *l*th coefficient matrix and $i = 1, \ldots, k$, where *k* is the number of endogenous variables

- MA*l*_*i*, numeric variables that contain the estimates of moving-average parameters and their standard errors, where *l* is the lag *l*th coefficient matrix and $i = 1, \ldots, k$, where *k* is the number of endogenous variables

- COV_*i*, numeric variables that contain the estimates of the covariance of innovations parameters when the maximum likelihood method is applied, where $i = 1, \ldots, k$

- DCCAB, a numeric variable that contains the estimates of α or β in the correlation equation for DCC representation and their standard errors

- CCC_*i*, numeric variables that contain the estimates of the conditional constant correlation parameters for CCC representation, where $i = 2, \ldots, k$

- DCCS_*i*, numeric variables that contain the estimates of the unconditional correlation parameters for DCC representation, where $i = 2, \ldots, k$

- GCHC_*i*, numeric variables that contain the estimates of the constant parameters of the covariance matrix and their standard errors, where $i = 1, \ldots, k$ for BEKK representation, *k* is the number of endogenous variables, and $i = 1$ for CCC and DCC representations

- ACH*l*_*i*, numeric variables that contain the estimates of the ARCH parameters of the covariance matrix and their standard errors, where *l* is the lag *l*th coefficient matrix and $i = 1, \ldots, k$ for BEKK, CCC, and DCC representations, where *k* is the number of endogenous variables

- EACH*l*_*i*, numeric variables that contain the estimates of the exponential ARCH parameters of the covariance matrix and their standard errors, where *l* is the lag *l*th coefficient matrix and $i = 1, \ldots, k$ for CCC and DCC representations, where *k* is the number of endogenous variables

- PACH*l*_*i*, numeric variables that contain the estimates of the power ARCH parameters of the covariance matrix and their standard errors, where *l* is the lag *l*th coefficient matrix and $i = 1, \ldots, k$ for CCC and DCC representations, where *k* is the number of endogenous variables

- QACH*l*_*i*, numeric variables that contain the estimates of the quadratic ARCH parameters of the covariance matrix and their standard errors, where *l* is the lag *l*th coefficient matrix and $i = 1, \ldots, k$ for CCC and DCC representations, where *k* is the number of endogenous variables

- TACH*l*_*i*, numeric variables that contain the estimates of the threshold ARCH parameters of the covariance matrix and their standard errors, where *l* is the lag *l*th coefficient matrix and $i = 1, \ldots, k$ for CCC and DCC representations, where *k* is the number of endogenous variables

- GCH*l*_*i*, numeric variables that contain the estimates of the GARCH parameters of the covariance matrix and their standard errors, where *l* is the lag *l*th coefficient matrix and $i = 1, \ldots, k$ for BEKK, CCC, and DCC representations, where *k* is the number of endogenous variables

- LAMBDA, a numeric variable that contains the estimates of power parameters in the PGARCH model for CCC and DCC representations and their standard errors

The OUTEST= data set contains the values shown in Table 42.9 for a bivariate case.

Table 42.9 OUTEST= Data Set

Obs	NAME	TYPE	CONST	AR1_1	AR1_2	AR2_1	AR2_2
1	y1	EST	δ_1	$\phi_{1,11}$	$\phi_{1,12}$	$\phi_{2,11}$	$\phi_{2,12}$
2		STD	se(δ_1)	se($\phi_{1,11}$)	se($\phi_{1,12}$)	se($\phi_{2,11}$)	se($\phi_{2,12}$)
3	y2	EST	δ_2	$\phi_{1,21}$	$\phi_{1,22}$	$\phi_{2,21}$	$\phi_{2,22}$
4		STD	se(δ_2)	se($\phi_{1,21}$)	se($\phi_{1,22}$)	se($\phi_{2,21}$)	se($\phi_{2,22}$)

Consider the following example:

```
proc varmax data=simul2 outest=est;
   model y1 y2 / p=2 noint noprint;
   cointeg rank=1 normalize=y1;
run;

proc print data=est;
run;
```

The output in Figure 42.91 shows the results of the OUTEST= data set.

Figure 42.91 OUTEST= Data Set

Log-periodogram of y2

Obs	NAME	TYPE	AR1_1	AR1_2	AR2_1	AR2_2	COV_1	COV_2	ALPHA1	BETA1
1	y1	EST	-0.46680	0.91295	-0.74332	-0.74621	94.7557	4.527	-0.46680	1.00000
2		STD	0.04786	0.09359	0.04526	0.04769	13.5365	10.303	0.04786	.
3	y2	EST	0.10667	-0.20862	0.40493	-0.57157	4.5268	109.570	0.10667	-1.95575
4		STD	0.05146	0.10064	0.04867	0.05128	10.3030	15.653	0.05146	.

OUTHT= Data Set

The OUTHT= data set contains predictions of conditional covariance matrices of innovations of the fitted GARCH model that the GARCH statement produces. The following output variables can be created:

- the BY variables, if BY-group processing is performed
- the ID variable, if the ID statement is specified
- Hi_j, numeric variables that contain the prediction of covariance, where $1 \leq i \leq j \leq k$, where k is the number of dependent variables

The OUTHT= data set contains the values shown in Table 42.10 for a bivariate case.

Table 42.10 OUTHT= Data Set

Obs	H1_1	H1_2	H2_2
1	h111	h121	h221
2	h112	h122	h222
:	:	:	:

The OUTHT= data set has the same number of observations as the OUT= data set. Both the OUTHT= and OUT= data sets include any observations at the beginning of the data set that are skipped because of the DIF=, DIFY=, DIFX=, P=, or XLAG= option and include the predicted observations at the end of the data set, which correspond with the LEAD= specification. If you specify an ID statement together with the OUTHT= and OUT= options, then the values of the ID variable in the two data sets correspond with one another.

Consider the following example of the OUTHT= option:

```
data garch;
   set garch;
   date = intnx( 'month', '01may1972'd, _n_-1 );
   format date yymms.;
run;

proc varmax data=garch;
   id date interval=month;
   model y1 y2 / p=1;
   garch q=1 outht=ht;
   output out=og lead=6;
run;

proc print data=og(obs=8);
   var date y1 for1 std1 lci1 uci1 y2 for2 std2 lci2 uci2;
run;

proc print data=ht(obs=8);
run;

proc print data=og(firstobs=499);
   var date y1 for1 std1 lci1 uci1 y2 for2 std2 lci2 uci2;
run;

proc print data=ht(firstobs=499);
run;
```

The output in Figure 42.92 and Figure 42.93 shows the first eight observations in the OUT= and OUTHT= data sets, respectively. The first observation is skipped in the GARCH model estimation because of the P=1 option, resulting in the missing values in the first observations in the OUT= and OUTHT= data sets. The output in Figure 42.94 and Figure 42.95 shows the last eight observations in the OUT= and OUTHT= data sets, respectively. In the OUT= data set, the standard deviations of the forecast of dependent variables are time-variant. The last six observations in OUTHT= data set are the multistep forecast of conditional covariance matrices of innovations.

Figure 42.92 First Part of OUT= Data Set

Log-periodogram of y2

Obs	date	y1	FOR1	STD1	LCI1	UCI1	y2	FOR2	STD2	LCI2	UCI2
1	1972/05	-4.4005	.	.	.	.	1.83794	.	.	.	.
2	1972/06	-8.0533	-4.2140	3.10387	-10.2975	1.86947	1.59720	1.92227	1.92885	-1.85820	5.70274
3	1972/07	-10.8362	-8.5587	3.21511	-14.8602	-2.25720	1.51833	-0.37752	1.33100	-2.98623	2.23118
4	1972/08	-6.0179	-11.9245	2.97553	-17.7564	-6.09254	-1.57445	-2.09795	1.75464	-5.53697	1.34108
5	1972/09	-7.8272	-4.3716	3.63437	-11.4949	2.75160	-0.03774	-0.09637	1.44118	-2.92102	2.72829
6	1972/10	-8.4293	-7.4084	3.14734	-13.5770	-1.23969	-0.40424	-0.73442	1.26093	-3.20580	1.73695
7	1972/11	-7.8156	-7.9499	2.89408	-13.6222	-2.27757	0.20642	-1.21238	1.26383	-3.68944	1.26469
8	1972/12	-8.0182	-7.5245	2.87208	-13.1537	-1.89535	0.43513	-0.65343	1.61823	-3.82511	2.51825

Figure 42.93 First Part of OUTHT= Data Set

Log-periodogram of y2

Obs	date	h1_1	h1_2	h2_2
1	1972/05	.	.	.
2	1972/06	9.6340	0.14073	3.72045
3	1972/07	10.3369	0.42643	1.77155
4	1972/08	8.8538	-1.19603	3.07876
5	1972/09	13.2086	1.36328	2.07699
6	1972/10	9.9058	-0.02914	1.58995
7	1972/11	8.3757	-0.29722	1.59728
8	1972/12	8.2489	-0.12736	2.61868

Figure 42.94 Last Part of OUT= Data Set

Log-periodogram of y2

Obs	date	y1	FOR1	STD1	LCI1	UCI1	y2	FOR2	STD2	LCI2	UCI2
499	2013/11	-6.1917	-4.1545	2.88303	-9.8051	1.4962	6.09470	6.33899	1.43651	3.5235	9.1545
500	2013/12	-10.2133	-8.6817	2.97211	-14.5070	-2.8565	2.88544	2.11833	1.28490	-0.4000	4.6367
501	2014/01	.	-11.8921	2.92171	-17.6186	-6.1657	.	-1.30455	1.33400	-3.9191	1.3100
502	2014/02	.	-11.7095	4.83388	-21.1837	-2.2353	.	-3.59592	2.37237	-8.2457	1.0538
503	2014/03	.	-10.2617	6.20050	-22.4145	1.8910	.	-4.17796	3.77457	-11.5760	3.2201
504	2014/04	.	-8.1778	7.02293	-21.9425	5.5869	.	-3.47144	4.98630	-13.2444	6.3015
505	2014/05	.	-6.0032	7.41997	-20.5461	8.5396	.	-1.98718	5.81618	-13.3867	9.4123
506	2014/06	.	-4.1332	7.56318	-18.9567	10.6904	.	-0.21231	6.27549	-12.5120	12.0874

Figure 42.95 Last Part of OUTHT= Data Set

Log-periodogram of y2

Obs	date	h1_1	h1_2	h2_2
499	2013/11	8.31189	-0.42221	2.06356
500	2013/12	8.83341	-0.00565	1.65098
501	2014/01	8.53639	-0.48367	1.77955
502	2014/02	9.42359	-0.13271	2.47088
503	2014/03	9.55818	-0.00081	2.85906
504	2014/04	9.58107	0.04780	3.07044
505	2014/05	9.58585	0.06690	3.18347
506	2014/06	9.58718	0.07508	3.24331

OUTSTAT= Data Set

The OUTSTAT= data set contains estimation results of the fitted model produced by the VARMAX statement. The following output variables can be created. The subindex i is $1, \ldots, k$, where k is the number of endogenous variables.

- the BY variables
- NAME, a character variable that contains the name of endogenous (dependent) variables
- SIGMA_i, numeric variables that contain the estimate of the innovation covariance matrix
- AICC, a numeric variable that contains the corrected Akaike's information criterion value
- HQC, a numeric variable that contains the Hannan-Quinn's information criterion value
- AIC, a numeric variable that contains the Akaike's information criterion value
- SBC, a numeric variable that contains the Schwarz Bayesian's information criterion value
- FPEC, a numeric variable that contains the final prediction error criterion value
- LOGLIK, a numeric variable that contains the value of the log-likelihood function calculated at the parameter estimates
- RSquare, a numeric variable that contains the value of the coefficient of determination
- FValue, a numeric variable that contains the F statistics
- PValue, a numeric variable that contains p-value for the F statistics

If the JOHANSEN= option is specified, the following items are added:

- Eigenvalue, a numeric variable that contains eigenvalues for the cointegration rank test of integrated order 1
- RestrictedEigenvalue, a numeric variable that contains eigenvalues for the cointegration rank test of integrated order 1 when the NOINT option is not specified

- Beta_*i*, numeric variables that contain long-run effect parameter estimates, $\boldsymbol{\beta}$
- Alpha_*i*, numeric variables that contain adjustment parameter estimates, $\boldsymbol{\alpha}$

If the JOHANSEN=(IORDER=2) option is specified, the following items are added:

- EValueI2_*i*, numeric variables that contain eigenvalues for the cointegration rank test of integrated order 2
- EValueI1, a numeric variable that contains eigenvalues for the cointegration rank test of integrated order 1
- Eta_*i*, numeric variables that contain the parameter estimates in integrated order 2, $\boldsymbol{\eta}$
- Xi_*i*, numeric variables that contain the parameter estimates in integrated order 2, $\boldsymbol{\xi}$

The OUTSTAT= data set contains the values shown Table 42.11 for a bivariate case.

Table 42.11 OUTSTAT= Data Set

Obs	NAME	SIGMA_1	SIGMA_2	AICC	HQC	AIC	SBC
1	y1	σ_{11}	σ_{12}	aicc	hqc	aic	sbc
2	y2	σ_{21}	σ_{22}	.	.	.	.

Obs	FPEC	LOGLIK	RSquare	FValue	PValue
1	fpec	loglik	R_1^2	F_1	$prob_1$
2	.	.	R_2^2	F_2	$prob_2$

Obs	EValueI2_1	EValueI2_2	EValueI1	Beta_1	Beta_2
1	e_{11}	e_{12}	e_1	β_{11}	β_{12}
2	e_{21}	.	e_2	β_{21}	β_{21}

Obs	Alpha_1	Alpha_2	Eta_1	Eta_2	Xi_1	Xi_2
1	α_{11}	α_{12}	η_{11}	η_{12}	ξ_{11}	ξ_{12}
2	α_{21}	α_{22}	η_{21}	η_{22}	ξ_{21}	ξ_{22}

Consider the following example:

```
proc varmax data=simul2 outstat=stat;
   model y1 y2 / p=2 noint noprint
                 cointtest=(johansen=(iorder=2));
   cointeg rank=1 normalize=y1;
run;

proc print data=stat;
run;
```

The output in Figure 42.96 shows the results of the OUTSTAT= data set.

Figure 42.96 OUTSTAT= Data Set

Log-periodogram of y2

Obs	NAME	SIGMA_1	SIGMA_2	AICC	HQC	AIC	SBC	FPEC	LOGLIK	RSquare	FValue	PValue	EValuel2_1
1	y1	94.7557	4.527	0	0	0	0	0	-551.049	0.93900	482.308	6.1637E-57	0.98486
2	y2	4.5268	109.570	.	.	.	.	.	.	0.93912	483.334	5.6124E-57	0.81451

Obs	EValuel2_2	EValuel1	Beta_1	Beta_2	Alpha_1	Alpha_2	Eta_1	Eta_2	Xi_1	Xi_2
1	0.95079	0.50864	1.00000	1.00000	-0.46680	0.007937	-0.012307	0.027030	54.1606	-52.3144
2	.	0.01108	-1.95575	-1.33622	0.10667	0.033530	0.015555	0.023086	-79.4240	-18.3308

Printed Output

The default printed output produced by the VARMAX procedure is described in the following list:

- descriptive statistics, which include the number of observations used, the names of the variables, their means and standard deviations (STD), their minimums and maximums, the differencing operations used, and the labels of the variables
- a type of model to fit the data and an estimation method
- a table of parameter estimates that shows the following for each parameter: the variable name for the left-hand side of equation, the parameter name, the parameter estimate, the approximate standard error, t value, the approximate probability ($Pr > |t|$), and the variable name for the right-hand side of equations in terms of each parameter
- the innovation covariance matrix
- the information criteria

If PRINT=ESTIMATES is specified, the VARMAX procedure prints the following list with the default printed output:

- the estimates of the constant vector (or seasonal constant matrix), the trend vector, the coefficient matrices of the distributed lags, the AR coefficient matrices, and the MA coefficient matrices
- the ALPHA and BETA parameter estimates for the error correction model
- the schematic representation of parameter estimates

If PRINT=DIAGNOSE is specified, the VARMAX procedure prints the following list with the default printed output:

- the cross-covariance and cross-correlation matrices of the residuals
- the tables of test statistics for the hypothesis that the residuals of the model are white noise:
  - Durbin-Watson (DW) statistics

- F test for autoregressive conditional heteroscedastic (ARCH) disturbances
- F test for AR disturbance
- Jarque-Bera normality test
- portmanteau test

ODS Table Names

The VARMAX procedure assigns a name to each table that it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 42.12.

Table 42.12 ODS Tables Produced in the VARMAX Procedure

ODS Table Name	Description	Option
ODS Tables Created by the MODEL Statement		
AccumImpulse	Accumulated impulse response matrices	IMPULSE=(ACCUM) IMPULSE=(ALL)
AccumImpulsebyVar	Accumulated impulse response by variable	IMPULSE=(ACCUM) IMPULSE=(ALL)
AccumImpulseX	Accumulated transfer function matrices	IMPULSX=(ACCUM) IMPULSX=(ALL)
AccumImpulseXbyVar	Accumulated transfer function by variable	IMPULSX=(ACCUM) IMPULSX=(ALL)
Alpha	$\boldsymbol{\alpha}$ coefficients	JOHANSEN=
AlphaInECM	$\boldsymbol{\alpha}$ coefficients when RANK=r	PRINT=(ESTIMATES) with ECM=
AlphaOnDrift	$\boldsymbol{\alpha}$ coefficients under the restriction of a deterministic term	JOHANSEN=
AlphaBetaInECM	$\Pi = \boldsymbol{\alpha}\boldsymbol{\beta}'$ coefficients when RANK=r	PRINT=(ESTIMATES) with ECM=
ANOVA	Univariate model diagnostic checks for the residuals	PRINT=DIAGNOSE
ARCoef	AR coefficients	PRINT=(ESTIMATES) with P=
ARRoots	Roots of AR characteristic polynomial	ROOTS with P=
Beta	$\boldsymbol{\beta}$ coefficients	JOHANSEN=
BetaInECM	$b\beta$ coefficients when RANK=r	PRINT=(ESTIMATES) with ECM=
BetaOnDrift	$\boldsymbol{\beta}$ coefficients under the restriction of a deterministic term	JOHANSEN=
CCCCorrConstant	Constant correlation matrix in the CCC GARCH model	CORRCONSTANT=EXPECT with FORM=CCC
Constant	Constant estimates	Without NOINT
CorrB	Correlations of parameter estimates	CORRB
CorrResiduals	Correlations of residuals	PRINT=DIAGNOSE

Table 42.12 *continued*

ODS Table Name	Description	Option
CorrResidualsbyVar	Correlations of residuals by variable	PRINT=DIAGNOSE
CorrResidualsGraph	Schematic representation of correlations of residuals	PRINT=DIAGNOSE
CorrXGraph	Schematic representation of sample correlations of independent series	CORRX
CorrYGraph	Schematic representation of sample correlations of dependent series	CORRY
CorrXLags	Correlations of independent series	CORRX
CorrXbyVar	Correlations of independent series by variable	CORRX
CorrYLags	Correlations of dependent series	CORRY
CorrYbyVar	Correlations of dependent series by variable	CORRY
CovarianceParameter-Estimates	Covariance parameter estimates	METHOD=ML without the PRIOR= option, or GARCH statement
CovB	Covariances of parameter estimates	COVB
CovInnovation	Covariances of the innovations	Default
CovPredictError	Covariance matrices of the prediction error	COVPE
CovPredictErrorbyVar	Covariances of the prediction error by variable	COVPE
CovResiduals	Covariances of residuals	PRINT=DIAGNOSE
CovResidualsbyVar	Covariances of residuals by variable	PRINT=DIAGNOSE
CovXLags	Covariances of independent series	COVX
CovXbyVar	Covariances of independent series by variable	COVX
CovYLags	Covariances of dependent series	COVY
CovYbyVar	Covariances of dependent series by variable	COVY
DCCCorrConstant	Unconditional correlation matrix in the DCC GARCH model	CORRCONSTANT=EXPECT with FORM=DCC
DecomposeCovPre-dictError	Decomposition of the prediction error covariances	DECOMPOSE
DecomposeCovPre-dictErrorbyVar	Decomposition of the prediction error covariances by variable	DECOMPOSE
DFTest	Dickey-Fuller test	DFTEST
DiagnostAR	Test the AR disturbance for the residuals	PRINT=DIAGNOSE
DiagnostWN	Test the ARCH disturbance and normality for the residuals	PRINT=DIAGNOSE
DynamicARCoef	AR coefficients of the dynamic model	DYNAMIC
DynamicConstant	Constant estimates of the dynamic model	DYNAMIC
DynamicCovInno-vation	Covariances of the innovations of the dynamic model	DYNAMIC

Table 42.12 *continued*

ODS Table Name	Description	Option
DynamicLinearTrend	Linear trend estimates of the dynamic model	DYNAMIC
DynamicMACoef	MA coefficients of the dynamic model	DYNAMIC
DynamicSConstant	Seasonal constant estimates of the dynamic model	DYNAMIC
DynamicParameter-Estimates	Parameter estimates table of the dynamic model	DYNAMIC
DynamicParameter-Graph	Schematic representation of the parameters of the dynamic model	DYNAMIC
DynamicQuadTrend	Quadratic trend estimates of the dynamic model	DYNAMIC
DynamicSeasonGraph	Schematic representation of the seasonal dummies of the dynamic model	DYNAMIC
DynamicXLagCoef	Dependent coefficients of the dynamic model	DYNAMIC
Hypothesis	Hypothesis of different deterministic terms in cointegration rank test	JOHANSEN=
HypothesisTest	Test hypothesis of different deterministic terms in cointegration rank test	JOHANSEN=
EigenvalueI2	Eigenvalues in integrated order 2	JOHANSEN= (IORDER=2)
Eta	η coefficients	JOHANSEN= (IORDER=2)
InfiniteARRepresent	Infinite order ar representation	IARR
InfoCriteria	Information criteria	Default
LinearTrend	Linear trend estimates	TREND=
LogLikelihood	Log likelihood	Default
MACoef	MA coefficients	Q=
MARoots	Roots of MA characteristic polynomial	ROOTS with Q=
MaxTest	Cointegration rank test using the maximum eigenvalue	JOHANSEN= (TYPE=MAX)
Minic	Tentative order selection	MINIC or MINIC=
ModelType	Type of model	Default
NObs	Number of observations	Default
OrthoImpulse	Orthogonalized impulse response matrices	IMPULSE=(ORTH) IMPULSE=(ALL)
OrthoImpulsebyVar	Orthogonalized impulse response by variable	IMPULSE=(ORTH) IMPULSE=(ALL)
ParameterEstimates	Parameter estimates table	Default
ParameterGraph	Schematic representation of the parameters	PRINT=ESTIMATES
PartialAR	Partial autoregression matrices	PARCOEF
PartialARGraph	Schematic representation of partial autoregression	PARCOEF

Table 42.12 *continued*

ODS Table Name	Description	Option
PartialCanCorr	Partial canonical correlation analysis	PCANCORR
PartialCorr	Partial cross-correlation matrices	PCORR
PartialCorrbyVar	Partial cross-correlations by variable	PCORR
PartialCorrGraph	Schematic representation of partial cross-correlations	PCORR
PortmanteauTest	Chi-square test table for residual cross-correlations	PRINT=DIAGNOSE
ProportionCovPredictError	Proportions of prediction error covariance decomposition	DECOMPOSE
ProportionCovPredictErrorbyVar	Proportions of prediction error covariance decomposition by variable	DECOMPOSE
RankTestI2	Cointegration rank test in integrated order 2	JOHANSEN=(IORDER=2)
RestrictMaxTest	Cointegration rank test using the maximum eigenvalue under the restriction of a deterministic term	JOHANSEN=(TYPE=MAX) without NOINT
RestrictTraceTest	Cointegration rank test using the trace under the restriction of a deterministic term	JOHANSEN=(TYPE=TRACE) without NOINT
QuadTrend	Quadratic trend estimates	TREND=QUAD
SeasonGraph	Schematic representation of the seasonal dummies	PRINT=ESTIMATES with NSEASON=
SConstant	Seasonal constant estimates	NSEASON=
SimpleImpulse	Impulse response matrices	IMPULSE=(SIMPLE) IMPULSE=(ALL)
SimpleImpulsebyVar	Impulse response by variable	IMPULSE=(SIMPLE) IMPULSE=(ALL)
SimpleImpulseX	Impulse response matrices of transfer function	IMPULSX=(SIMPLE) IMPULSX=(ALL)
SimpleImpulseXbyVar	Impulse response of transfer function by variable	IMPULSX=(SIMPLE) IMPULSX=(ALL)
Summary	Simple summary statistics	Default
SWTest	Common trends test	SW=
TraceTest	Cointegration rank test using the trace	JOHANSEN=(TYPE=TRACE)
Xi	ξ coefficient matrix	JOHANSEN=(IORDER=2)
XLagCoef	Dependent coefficients	XLAG=
YWEstimates	Yule-Walker estimates	YW

Table 42.12 *continued*

ODS Table Name	Description	Option
ODS Tables Created by the GARCH Statement		
ARCHCoef	ARCH coefficients	Q=
GARCHCoef	GARCH coefficients	P=
GARCHConstant	GARCH constant estimates	PRINT=ESTIMATES
GARCHParameter-Estimates	GARCH parameter estimates table	Default
GARCHParameter-Graph	Schematic representation of the garch parameters	PRINT=ESTIMATES
GARCHRoots	Roots of GARCH characteristic polynomial	ROOTS
ODS Tables Created by the COINTEG Statement or the ECM Option in the MODEL Statement		
AlphaAndBetaPa-rameterEstimaters	Parameter estimates of $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\beta}_0$, and $\boldsymbol{\beta}_1$	Default
AlphaInECM	$\boldsymbol{\alpha}$ coefficients when RANK=r	PRINT=ESTIMATES
AlphaBetaInECM	$\Pi = \boldsymbol{\alpha}\boldsymbol{\beta}'$ coefficients when RANK=r	PRINT=ESTIMATES
AlphaOnAlpha	$\boldsymbol{\alpha}$ coefficients under the restriction of α	J=
AlphaOnBeta	$\boldsymbol{\alpha}$ coefficients under the restriction of β	H=
AlphaTestResults	Hypothesis testing of $\boldsymbol{\alpha}$	J=
BetaInECM	$\boldsymbol{\beta}$ coefficients when RANK=r	PRINT=ESTIMATES
BetaOnBeta	$\boldsymbol{\beta}$ coefficients under the restriction of $\boldsymbol{\beta}$	H=
BetaOnAlpha	$\boldsymbol{\beta}$ coefficients under the restriction of $\boldsymbol{\alpha}$	J=
BetaTestResults	Hypothesis testing of β	H=
GrangerRepresent	Coefficient of Granger representation	PRINT=ESTIMATES
HMatrix	Restriction matrix for $\boldsymbol{\beta}$	H=
JMatrix	Restriction matrix for $\boldsymbol{\alpha}$	J=
WeakExogeneity	Testing weak exogeneity of each dependent variable with respect to BETA	EXOGENEITY
ODS Tables Created by the CAUSAL Statement		
CausalityTest	Granger causality test	Default
GroupVars	Two groups of variables	Default
ODS Tables Created by the RESTRICT Statement		
Restrict	Restriction table	Default
ODS Tables Created by the TEST Statement		
Test	Wald test	Default
ODS Tables Created by the OUTPUT Statement		
Forecasts	Forecasts table	Without NOPRINT

Note that the ODS table names suffixed by "byVar" can be obtained with the PRINTFORM=UNIVARIATE option.

ODS Graphics

This section describes the use of ODS for creating statistical graphs with the VARMAX procedure.

When ODS GRAPHICS are in effect, the VARMAX procedure produces a variety of plots for each dependent variable.

The plots available are as follows:

- The procedure displays the following plots for each dependent variable in the MODEL statement with the PLOT= option in the VARMAX statement:
  - impulse response function
  - impulse response of the transfer function
  - time series and predicted series
  - prediction errors
  - distribution of the prediction errors
  - normal quantile of the prediction errors
  - ACF of the prediction errors
  - PACF of the prediction errors
  - IACF of the prediction errors
  - log scaled white noise test of the prediction errors
- The procedure displays forecast plots for each dependent variable in the OUTPUT statement with the PLOT= option in the VARMAX statement.

ODS Graph Names

The VARMAX procedure assigns a name to each graph it creates by using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 42.13.

Table 42.13 ODS Graphics Produced in the VARMAX Procedure

ODS Table Name	Plot Description	Statement
ErrorACFPlot	Autocorrelation function of prediction errors	MODEL
ErrorIACFPlot	Inverse autocorrelation function of prediction errors	MODEL
ErrorPACFPlot	Partial autocorrelation function of prediction errors	MODEL
ErrorDiagnosticsPanel	Diagnostics of prediction errors	MODEL
ErrorNormalityPanel	Histogram and Q-Q plot of prediction errors	MODEL
ErrorDistribution	Distribution of prediction errors	MODEL
ErrorQQPlot	Q-Q plot of prediction errors	MODEL
ErrorWhiteNoisePlot	White noise test of prediction errors	MODEL

Table 42.13 *continued*

ODS Table Name	Plot Description	Statement
ErrorPlot	Prediction errors	MODEL
ModelPlot	Time series and predicted series	MODEL
DCCPanel	Dynamic conditional covariances	PROC
AccumulatedIRFPanel	Accumulated impulse response function	MODEL
AccumulatedIRFXPanel	Accumulated impulse response of transfer function	MODEL
OrthogonalIRFPanel	Orthogonalized impulse response function	MODEL
SimpleIRFPanel	Simple impulse response function	MODEL
SimpleIRFXPanel	Simple impulse response of transfer function	MODEL
ModelForecastsPlot	Time series and forecasts	OUTPUT
ForecastsOnlyPlot	Forecasts	OUTPUT

Computational Issues

Computational Method

The VARMAX procedure uses numerous linear algebra routines and frequently uses the sweep operator (Goodnight 1979) and the Cholesky root (Golub and Van Loan 1983).

In addition, the VARMAX procedure uses the nonlinear optimization (NLO) subsystem to perform nonlinear optimization tasks for the maximum likelihood estimation. The optimization requires intensive computation.

Convergence Problems

For some data sets, the computation algorithm can fail to converge. Nonconvergence can result from a number of causes, including flat or ridged likelihood surfaces and ill-conditioned data.

If you experience convergence problems, the following points might be helpful:

- Data that contain extreme values can affect results in PROC VARMAX. Rescaling the data can improve stability.
- Changing the TECH=, MAXITER=, and MAXFUNC= options in the NLOPTIONS statement can improve the stability of the optimization process.
- Specifying a different model that might fit the data more closely and might improve convergence.

Memory

Let T be the length of each series, k be the number of dependent variables, p be the order of autoregressive terms, and q be the order of moving-average terms. The number of parameters to estimate for a VARMA(p, q)

model is

$$k + (p+q)k^2 + k * (k+1)/2$$

As k increases, the number of parameters to estimate increases very quickly. Furthermore, the memory requirement for VARMA(p,q) quadratically increases as k and T increase.

For a VARMAX(p,q,s) model and GARCH-type multivariate conditional heteroscedasticity models, the number of parameters to estimate and the memory requirements are considerable.

Computing Time

PROC VARMAX is computationally intensive, and execution times can be long. Extensive CPU time is often required to compute the maximum likelihood estimates.

Examples: VARMAX Procedure

Example 42.1: Analysis of United States Economic Variables

Consider the following four-dimensional system of US economic variables. Quarterly data for the years 1954 to 1987 are used (Lütkepohl 1993, Table E.3.).

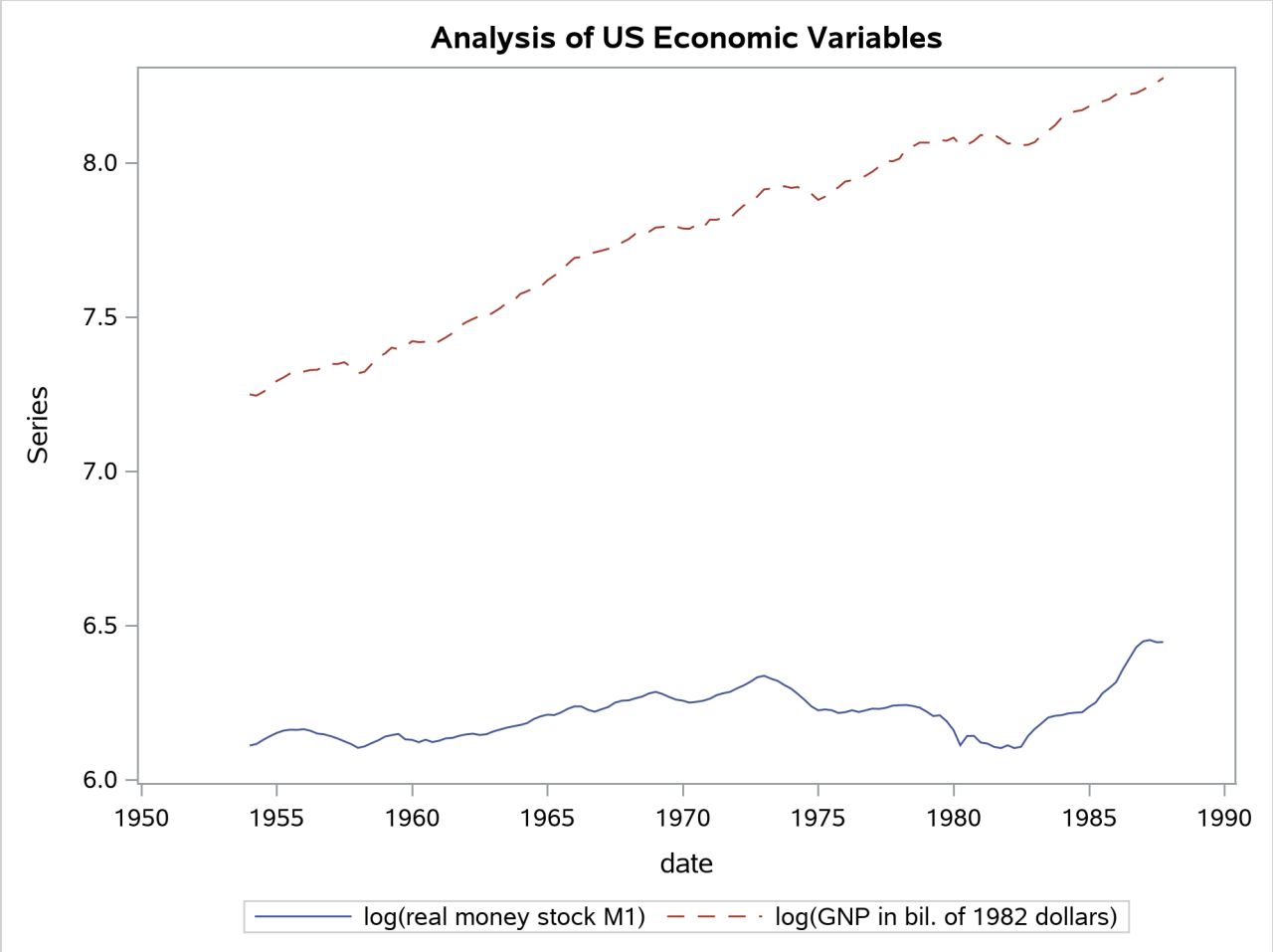
```
title 'Analysis of US Economic Variables';
data us_money;
   date=intnx( 'qtr', '01jan54'd, _n_-1 );
   format date yyq. ;
   input y1 y2 y3 y4 @@;
   y1=log(y1);
   y2=log(y2);
   label y1='log(real money stock M1)'
         y2='log(GNP in bil. of 1982 dollars)'
         y3='Discount rate on 91-day T-bills'
         y4='Yield on 20-year Treasury bonds';
datalines;
450.9 1406.8  0.010800000 0.026133333
453.0 1401.2 0.0081333333 0.025233333
459.1 1418.0 0.0087000000 0.024900000

   ... more lines ...
```

The following statements plot the series:

```
proc sgplot data=us_money;
   series x=date y=y1 / lineattrs=(pattern=solid);
   series x=date y=y2 / lineattrs=(pattern=dash);
   yaxis label="Series";
run;
```

Output 42.1.1 shows the plot of the variables $y1$ and $y2$.

Output 42.1.1 Plot of Data

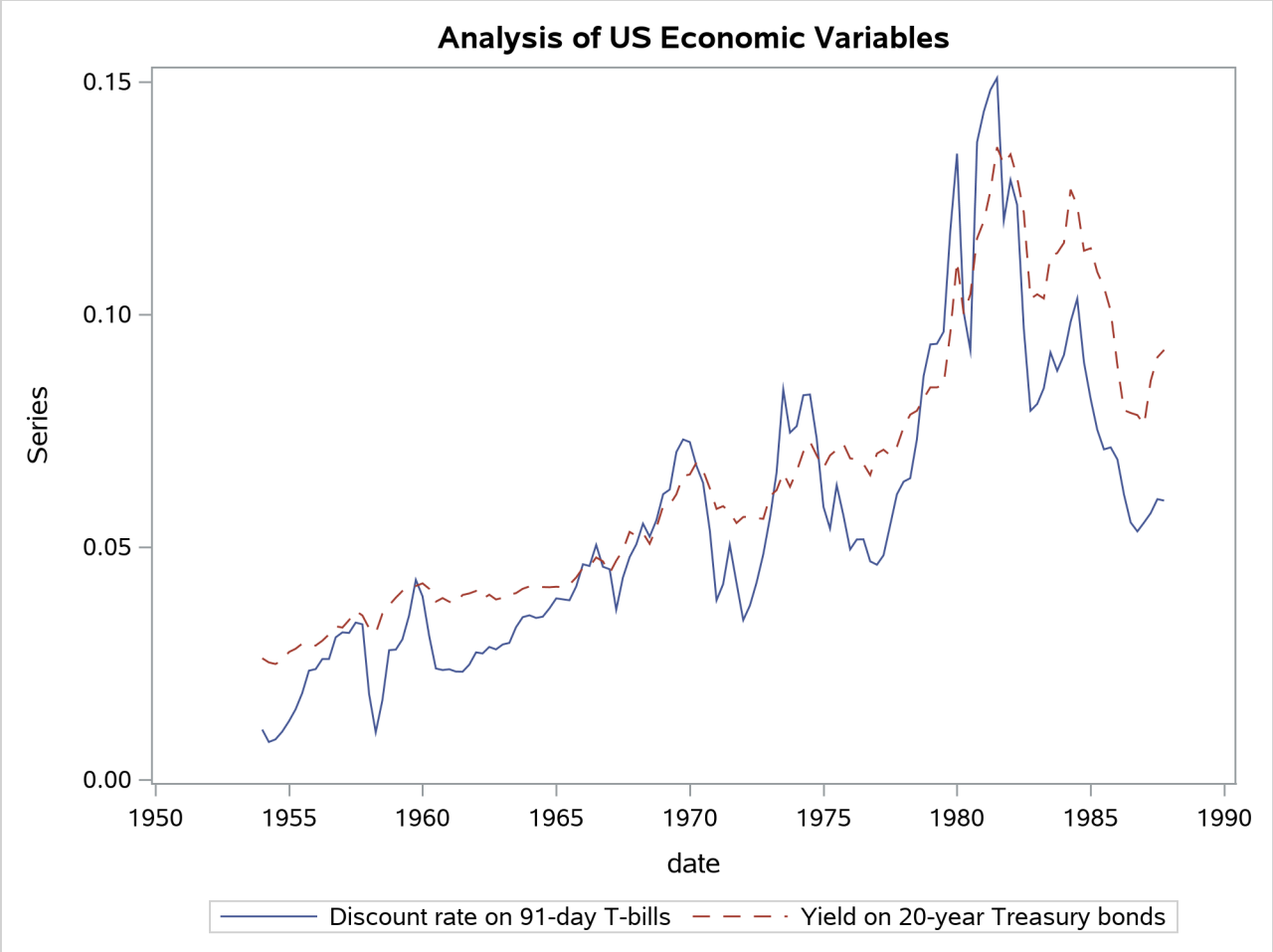


The following statements plot the variables $y3$ and $y4$:

```
proc sgplot data=us_money;
   series x=date y=y3 / lineattrs=(pattern=solid);
   series x=date y=y4 / lineattrs=(pattern=dash);
   yaxis label="Series";
run;
```

Output 42.1.2 shows the plot of the variables $y3$ and $y4$.

Output 42.1.2 Plot of Data



The following statements perform the Dickey-Fuller test for stationarity, the Johansen cointegrated test integrated order 2, and the exogeneity test. The VECM(2) is fit to the data.

```
proc varmax data=us_money;
   id date interval=qtr;
   model y1-y4 / p=2 lagmax=6 dftest
                 print=(iarr(3) estimates diagnose)
                 cointtest=(johansen=(iorder=2));
   cointeg rank=1 normalize=y1 exogeneity;
run;
```

From the outputs shown in Output 42.1.5, you can see that the series has unit roots and is cointegrated in

rank 1 with integrated order 1. The fitted VECM(2) is given as

$$\Delta \mathbf{y}_t = \begin{pmatrix} 0.0408 \\ 0.0860 \\ 0.0052 \\ -0.0144 \end{pmatrix} + \begin{pmatrix} -0.0140 & 0.0065 & -0.2026 & 0.1306 \\ -0.0281 & 0.0131 & -0.4080 & 0.2630 \\ -0.0022 & 0.0010 & -0.0312 & 0.0201 \\ 0.0051 & -0.0024 & 0.0741 & -0.0477 \end{pmatrix} \mathbf{y}_{t-1} + \begin{pmatrix} 0.3460 & 0.0913 & -0.3535 & -0.9690 \\ 0.0994 & 0.0379 & 0.2390 & 0.2866 \\ 0.1812 & 0.0786 & 0.0223 & 0.4051 \\ 0.0322 & 0.0496 & -0.0329 & 0.1857 \end{pmatrix} \Delta \mathbf{y}_{t-1} + \boldsymbol{\epsilon}_t$$

The Δ prefixed to a variable name implies differencing.

Output 42.1.3 through Output 42.1.16 show the details. Output 42.1.3 shows the descriptive statistics.

Output 42.1.3 Descriptive Statistics

Analysis of US Economic Variables

The VARMAX Procedure

Number of Observations	136
Number of Pairwise Missing	0

Simple Summary Statistics

Variable	Type	N	Mean	Standard Deviation	Min	Max	Label
y1	**Dependent**	136	6.21295	0.07924	6.10278	6.45331	log(real money stock M1)
y2	**Dependent**	136	7.77890	0.30110	7.24508	8.27461	log(GNP in bil. of 1982 dollars)
y3	**Dependent**	136	0.05608	0.03109	0.00813	0.15087	Discount rate on 91-day T-bills
y4	**Dependent**	136	0.06458	0.02927	0.02490	0.13600	Yield on 20-year Treasury bonds

Output 42.1.4 shows the output for Dickey-Fuller tests for the nonstationarity of each series. The null hypothesis is that there exists a unit root. All series have a unit root.

Output 42.1.4 Unit Root Tests

Unit Root Test

Variable	Type	Rho	Pr < Rho	Tau	Pr < Tau
y1	**Zero Mean**	0.05	0.6934	1.14	0.9343
	Single Mean	-2.97	0.6572	-0.76	0.8260
	Trend	-5.91	0.7454	-1.34	0.8725
y2	**Zero Mean**	0.13	0.7124	5.14	0.9999
	Single Mean	-0.43	0.9309	-0.79	0.8176
	Trend	-9.21	0.4787	-2.16	0.5063
y3	**Zero Mean**	-1.28	0.4255	-0.69	0.4182
	Single Mean	-8.86	0.1700	-2.27	0.1842
	Trend	-18.97	0.0742	-2.86	0.1803
y4	**Zero Mean**	0.40	0.7803	0.45	0.8100
	Single Mean	-2.79	0.6790	-1.29	0.6328
	Trend	-12.12	0.2923	-2.33	0.4170

The Johansen cointegration rank test shows whether the series is integrated order either 1 or 2 as shown in Output 42.1.5. The last two columns in Output 42.1.5 explain the cointegration rank test with integrated order 1. The results indicate that there is a cointegrated relationship with cointegration rank 1 with respect to the 0.05 significance level because the test statistic for the null hypothesis H0: $r = 0$ is 55.9633 and its corresponding p-value is 0.0072, less than 0.05 (indicating that H0: $r = 0$ should be rejected), and the test statistic for the null hypothesis H0: $r = 1$ is 20.6542 and its corresponding p-value is 0.3775, greater than 0.05 (indicating that H0: $r = 1$ cannot be rejected). Now, look at the row associated with $r = 1$. All p-values of the tests for the null hypothesis that the series are integrated order 2 are zeros, less than 0.05 significance level (indicating that the null hypothesis should be rejected).

Output 42.1.5 Cointegration Rank Test

Cointegration Rank Test for I(2)						
r\k-r-s	4	3	2	1	Trace of I(1)	Pr > Trace of I(1)
0	384.6090	214.3790	107.9378	37.0252	55.9633	0.0072
Pr > Trace of I(2)	0.0000	0.0000	0.0000	0.0000		
1		219.6239	89.2151	27.3261	20.6542	0.3775
Pr > Trace of I(2)		0.0000	0.0000	0.0000		
2			73.6178	22.1328	2.6477	0.9803
Pr > Trace of I(2)			0.0000	0.0000		
3				38.2943	0.0149	0.9031
Pr > Trace of I(2)				0.0000		

Output 42.1.6 shows the estimates of the long-run parameter, $\boldsymbol{\beta}$, and the adjustment coefficient, $\boldsymbol{\alpha}$.

Output 42.1.6 Cointegration Rank Test, Continued

Beta				
Variable	1	2	3	4
y1	1.00000	1.00000	1.00000	1.00000
y2	-0.46458	-0.63174	-0.69996	-0.16140
y3	14.51619	-1.29864	1.37007	-0.61806
y4	-9.35520	7.53672	2.47901	1.43731

Alpha				
Variable	1	2	3	4
y1	-0.01396	0.01396	-0.01119	0.00008
y2	-0.02811	-0.02739	-0.00032	0.00076
y3	-0.00215	-0.04967	-0.00183	-0.00072
y4	0.00510	-0.02514	-0.00220	0.00016

Output 42.1.7 shows the estimates η and ξ.

Output 42.1.7 Cointegration Rank Test, Continued

	Eta			
Variable	**1**	**2**	**3**	**4**
y1	52.74907	41.74502	-20.80403	55.77415
y2	-49.10609	-9.40081	98.87199	22.56416
y3	68.29674	-144.83173	-27.35953	15.51142
y4	121.25932	271.80496	85.85156	-130.11599

	Xi			
Variable	**1**	**2**	**3**	**4**
y1	-0.00842	-0.00052	-0.00208	-0.00250
y2	0.00141	0.00213	-0.00736	-0.00058
y3	-0.00445	0.00541	-0.00150	0.00310
y4	-0.00211	-0.00064	-0.00130	0.00197

Output 42.1.8 shows that the VECM(2) is fit to the data. The RANK=1 option in the COINTEG statement produces the estimates of the long-run parameter, β, and the adjustment coefficient, α.

Output 42.1.8 Parameter Estimates

Analysis of US Economic Variables

The VARMAX Procedure

Type of Model	VECM(2)
Estimation Method	Maximum Likelihood Estimation
Cointegrated Rank	1

	Beta
Variable	**1**
y1	1.00000
y2	-0.46458
y3	14.51619
y4	-9.35520

	Alpha
Variable	**1**
y1	-0.01396
y2	-0.02811
y3	-0.00215
y4	0.00510

Output 42.1.9 shows the parameter estimates in terms of the constant, the lag 1 coefficients ($\mathbf{y}_{t-1}$) that are contained in the $\alpha\beta'$ estimates, and the coefficients that are associated with the lag 1 first differences ($\Delta\mathbf{y}_{t-1}$).

Output 42.1.9 Parameter Estimates, Continued

Constant	
Variable	**Constant**
y1	0.04076
y2	0.08595
y3	0.00518
y4	-0.01438

Parameter Alpha * Beta' Estimates				
Variable	**y1**	**y2**	**y3**	**y4**
y1	-0.01396	0.00648	-0.20263	0.13059
y2	-0.02811	0.01306	-0.40799	0.26294
y3	-0.00215	0.00100	-0.03121	0.02011
y4	0.00510	-0.00237	0.07407	-0.04774

AR Coefficients of Differenced Lag					
DIF Lag	**Variable**	**y1**	**y2**	**y3**	**y4**
1	**y1**	0.34603	0.09131	-0.35351	-0.96895
	y2	0.09936	0.03791	0.23900	0.28661
	y3	0.18118	0.07859	0.02234	0.40508
	y4	0.03222	0.04961	-0.03292	0.18568

Output 42.1.10 through Output 42.1.12 show the parameter estimates and their significance.

Output 42.1.10 Parameter Estimates, Continued

Model Parameter Estimates

Equation	Parameter	Estimate	Standard Error	t Value	Pr > |t|	Variable
D_y1	CONST1	0.04076	0.01418	2.87	0.0048	1
	AR1_1_1	-0.01396	0.00495	-2.82	0.0056	y1(t-1)
	AR1_1_2	0.00648	0.00230	2.82	0.0056	y2(t-1)
	AR1_1_3	-0.20263	0.07191	-2.82	0.0056	y3(t-1)
	AR1_1_4	0.13059	0.04634	2.82	0.0056	y4(t-1)
	AR2_1_1	0.34603	0.06414	5.39	<.0001	D_y1(t-1)
	AR2_1_2	0.09131	0.07334	1.25	0.2154	D_y2(t-1)
	AR2_1_3	-0.35351	0.11024	-3.21	0.0017	D_y3(t-1)
	AR2_1_4	-0.96895	0.20737	-4.67	<.0001	D_y4(t-1)
D_y2	CONST2	0.08595	0.01679	5.12	<.0001	1
	AR1_2_1	-0.02811	0.00586	-4.79	<.0001	y1(t-1)
	AR1_2_2	0.01306	0.00272	4.79	<.0001	y2(t-1)
	AR1_2_3	-0.40799	0.08514	-4.79	<.0001	y3(t-1)
	AR1_2_4	0.26294	0.05487	4.79	<.0001	y4(t-1)
	AR2_2_1	0.09936	0.07594	1.31	0.1932	D_y1(t-1)
	AR2_2_2	0.03791	0.08683	0.44	0.6632	D_y2(t-1)
	AR2_2_3	0.23900	0.13052	1.83	0.0695	D_y3(t-1)
	AR2_2_4	0.28661	0.24552	1.17	0.2453	D_y4(t-1)
D_y3	CONST3	0.00518	0.01608	0.32	0.7476	1
	AR1_3_1	-0.00215	0.00562	-0.38	0.7024	y1(t-1)
	AR1_3_2	0.00100	0.00261	0.38	0.7024	y2(t-1)
	AR1_3_3	-0.03121	0.08151	-0.38	0.7024	y3(t-1)
	AR1_3_4	0.02011	0.05253	0.38	0.7024	y4(t-1)
	AR2_3_1	0.18118	0.07271	2.49	0.0140	D_y1(t-1)
	AR2_3_2	0.07859	0.08313	0.95	0.3463	D_y2(t-1)
	AR2_3_3	0.02234	0.12496	0.18	0.8584	D_y3(t-1)
	AR2_3_4	0.40508	0.23506	1.72	0.0873	D_y4(t-1)
D_y4	CONST4	-0.01438	0.00803	-1.79	0.0758	1
	AR1_4_1	0.00510	0.00281	1.82	0.0713	y1(t-1)
	AR1_4_2	-0.00237	0.00130	-1.82	0.0713	y2(t-1)
	AR1_4_3	0.07407	0.04072	1.82	0.0713	y3(t-1)
	AR1_4_4	-0.04774	0.02624	-1.82	0.0713	y4(t-1)
	AR2_4_1	0.03222	0.03632	0.89	0.3768	D_y1(t-1)
	AR2_4_2	0.04961	0.04153	1.19	0.2345	D_y2(t-1)
	AR2_4_3	-0.03292	0.06243	-0.53	0.5990	D_y3(t-1)
	AR2_4_4	0.18568	0.11744	1.58	0.1164	D_y4(t-1)

Output 42.1.11 Parameter Estimates, Continued

Alpha and Beta Parameter Estimates

Equation	Parameter	Estimate	Standard Error	t Value	Pr > |t|	Variable
D_y1	ALPHA1_1	-0.01396	0.00495	-2.82	0.0056	Beta[,1]'*_DEP_(t-1)
	BETA1_1	1.00000				y1(t-1)
D_y2	ALPHA2_1	-0.02811	0.00586	-4.79	<.0001	Beta[,1]'*_DEP_(t-1)
	BETA2_1	-0.46458				y2(t-1)
D_y3	ALPHA3_1	-0.00215	0.00562	-0.38	0.7024	Beta[,1]'*_DEP_(t-1)
	BETA3_1	14.51619				y3(t-1)
D_y4	ALPHA4_1	0.00510	0.00281	1.82	0.0713	Beta[,1]'*_DEP_(t-1)
	BETA4_1	-9.35520				y4(t-1)

Output 42.1.12 Parameter Estimates, Continued

Covariance Parameter Estimates

Parameter	Estimate	Standard Error	t Value	Pr > |t|
COV1_1	0.00005	0.00001	8.19	<.0001
COV1_2	0.00001	0.00001	2.78	0.0062
COV2_2	0.00007	0.00001	8.19	<.0001
COV1_3	-0.00001	0.00001	-1.60	0.1118
COV2_3	0.00002	0.00001	2.71	0.0077
COV3_3	0.00007	0.00001	8.19	<.0001
COV1_4	-0.00000	0.00000	-1.31	0.1936
COV2_4	0.00001	0.00000	3.29	0.0013
COV3_4	0.00002	0.00000	6.67	<.0001
COV4_4	0.00002	0.00000	8.19	<.0001

Output 42.1.13 shows the innovation covariance matrix estimates, the log-likelihood, the various information criteria results, and the tests for white noise residuals. According to the portmanteau test results, the residuals have significant correlations at lag 2 and 3, indicating that a VECM(3) model might be a better fit than the VECM(2) model.

Output 42.1.13 Diagnostic Checks

Covariances of Innovations

Variable	y1	y2	y3	y4
y1	0.00005	0.00001	-0.00001	-0.00000
y2	0.00001	0.00007	0.00002	0.00001
y3	-0.00001	0.00002	0.00007	0.00002
y4	-0.00000	0.00001	0.00002	0.00002

Log-likelihood	2479.23

Information Criteria	
AICC	-4859
HQC	-4844.07
AIC	-4886.46
SBC	-4782.14
FPEC	2.23E-18

Schematic Representation of Cross Correlations of Residuals

Variable/Lag	0	1	2	3	4	5	6
y1	++..	....	++..	....	+...	..−−	....
y2	++++	....	....	....	....	....	....
y3	.+++	....	+.−.	..++	−...	....	....
y4	.+++	....	....	..+.	....	....	....

+ is > 2*std error, - is < -2*std error, . is between

Portmanteau Test for Cross Correlations of Residuals

Up To Lag	DF	Chi-Square	Pr > ChiSq
3	16	53.90	<.0001
4	32	74.03	<.0001
5	48	103.08	<.0001
6	64	116.94	<.0001

Output 42.1.14 describes how well each univariate equation fits the data. The residuals for $y3$ and $y4$ differ from normality. Except for the residuals for $y3$, there are no AR effects on other residuals. Except for the residuals for $y4$, there are no ARCH effects on other residuals.

Output 42.1.14 Diagnostic Checks, Continued

Univariate Model ANOVA Diagnostics

Variable	R-Square	Standard Deviation	F Value	Pr > F
y1	0.6754	0.00712	32.51	<.0001
y2	0.3070	0.00843	6.92	<.0001
y3	0.1328	0.00807	2.39	0.0196
y4	0.0831	0.00403	1.42	0.1963

Univariate Model White Noise Diagnostics

		Normality		ARCH	
Variable	Durbin Watson	Chi-Square	Pr > ChiSq	F Value	Pr > F
y1	2.13418	7.19	0.0275	1.62	0.2053
y2	2.04003	1.20	0.5483	1.23	0.2697
y3	1.86892	253.76	<.0001	1.78	0.1847
y4	1.98440	105.21	<.0001	21.01	<.0001

Univariate Model AR Diagnostics

	AR1		AR2		AR3		AR4	
Variable	F Value	Pr > F	F Value	Pr > F	F Value	Pr > F	F Value	Pr > F
y1	0.68	0.4126	2.98	0.0542	2.01	0.1154	2.48	0.0473
y2	0.05	0.8185	0.12	0.8842	0.41	0.7453	0.30	0.8762
y3	0.56	0.4547	2.86	0.0610	4.83	0.0032	3.71	0.0069
y4	0.01	0.9340	0.16	0.8559	1.21	0.3103	0.95	0.4358

The PRINT=(IARR) option provides the VAR(2) representation in Output 42.1.15.

Output 42.1.15 Infinite Order AR Representation

Infinite Order AR Representation

Lag	Variable	y1	y2	y3	y4
1	**y1**	1.33208	0.09780	-0.55614	-0.83836
	y2	0.07125	1.05096	-0.16899	0.54955
	y3	0.17903	0.07959	0.99113	0.42520
	y4	0.03732	0.04724	0.04116	1.13795
2	**y1**	-0.34603	-0.09131	0.35351	0.96895
	y2	-0.09936	-0.03791	-0.23900	-0.28661
	y3	-0.18118	-0.07859	-0.02234	-0.40508
	y4	-0.03222	-0.04961	0.03292	-0.18568
3	**y1**	0.00000	0.00000	0.00000	0.00000
	y2	0.00000	0.00000	0.00000	0.00000
	y3	0.00000	0.00000	0.00000	0.00000
	y4	0.00000	0.00000	0.00000	0.00000

Output 42.1.16 shows whether each variable is the weak exogeneity of other variables. The variable $y1$ is not the weak exogeneity of other variables, $y2$, $y3$, and $y4$; the variable $y2$ is not the weak exogeneity of other variables, $y1$, $y3$, and $y4$; the variables $y3$ and $y4$ are the weak exogeneity of other variables.

Output 42.1.16 Weak Exogeneity Test

Testing Weak Exogeneity of Each Variable			
Variable	**DF**	**Chi-Square**	**Pr > ChiSq**
y1	1	6.55	0.0105
y2	1	12.54	0.0004
y3	1	0.09	0.7695
y4	1	1.81	0.1786

Example 42.2: Analysis of German Economic Variables

This example considers a three-dimensional VAR(2) model. The model contains the logarithms of a quarterly, seasonally adjusted West German fixed investment, disposable income, and consumption expenditures. The data used are in Lütkepohl (1993, Table E.1).

```
title 'Analysis of German Economic Variables';
data west;
   date = intnx( 'qtr', '01jan60'd, _n_-1 );
   format date yyq. ;
   input y1 y2 y3 @@;
   y1 = log(y1);
   y2 = log(y2);
   y3 = log(y3);
   label y1 = 'logarithm of investment'
         y2 = 'logarithm of income'
         y3 = 'logarithm of consumption';
datalines;
180  451  415 179  465  421 185  485  434 192  493  448
211  509  459 202  520  458 207  521  479 214  540  487

   ... more lines ...


data use;
   set west;
   where  date < '01jan79'd;
   keep date y1 y2 y3;
run;

proc varmax data=use;
   id date interval=qtr;
   model y1-y3 / p=2 dify=(1)
                 print=(decompose(6) impulse=(stderr) estimates diagnose)
                 printform=both lagmax=3;
   causal group1=(y1) group2=(y2 y3);
   output lead=5;
run;
```

First, the differenced data are modeled as a VAR(2) with the following result:

$$\begin{aligned}\Delta \mathbf{y}_t &= \begin{pmatrix} -0.01672 \\ 0.01577 \\ 0.01293 \end{pmatrix} + \begin{pmatrix} -0.31963 & 0.14599 & 0.96122 \\ 0.04393 & -0.15273 & 0.28850 \\ -0.00242 & 0.22481 & -0.26397 \end{pmatrix} \Delta \mathbf{y}_{t-1} \\ &\quad + \begin{pmatrix} -0.16055 & 0.11460 & 0.93439 \\ 0.05003 & 0.01917 & -0.01020 \\ 0.03388 & 0.35491 & -0.02223 \end{pmatrix} \Delta \mathbf{y}_{t-2} + \boldsymbol{\epsilon}_t\end{aligned}$$

The parameter estimates AR1_1_2, AR1_1_3, AR2_1_2, and AR2_1_3 are insignificant, and the VARX model is fitted in the next step.

The detailed output is shown in Output 42.2.1 through Output 42.2.8.

Output 42.2.1 shows the descriptive statistics.

Output 42.2.1 Descriptive Statistics

Analysis of German Economic Variables

The VARMAX Procedure

Number of Observations	75
Number of Pairwise Missing	0
Observation(s) eliminated by differencing	1

Simple Summary Statistics

Variable	Type	N	Mean	Standard Deviation	Min	Max	Difference	Label
y1	**Dependent**	75	0.01811	0.04680	-0.14018	0.19358	1	logarithm of investment
y2	**Dependent**	75	0.02071	0.01208	-0.02888	0.05023	1	logarithm of income
y3	**Dependent**	75	0.01987	0.01040	-0.01300	0.04483	1	logarithm of consumption

Output 42.2.2 shows that a VAR(2) model is fit to the data.

Output 42.2.2 Parameter Estimates

Analysis of German Economic Variables

The VARMAX Procedure

Type of Model	VAR(2)
Estimation Method	Least Squares Estimation

	Constant
Variable	**Constant**
y1	-0.01672
y2	0.01577
y3	0.01293

		AR		
Lag	**Variable**	**y1**	**y2**	**y3**
1	**y1**	-0.31963	0.14599	0.96122
	y2	0.04393	-0.15273	0.28850
	y3	-0.00242	0.22481	-0.26397
2	**y1**	-0.16055	0.11460	0.93439
	y2	0.05003	0.01917	-0.01020
	y3	0.03388	0.35491	-0.02223

Output 42.2.3 shows the parameter estimates and their significance.

Output 42.2.3 Parameter Estimates, Continued

Schematic Representation

Variable/Lag	C	AR1	AR2
y1	.	-..	...
y2	+	...	...
y3	+	.+.	.+.

+ is > 2*std error, - is < -2*std error, . is between, * is N/A

Model Parameter Estimates

Equation	Parameter	Estimate	Standard Error	t Value	Pr > |t|	Variable
y1	CONST1	-0.01672	0.01723	-0.97	0.3352	1
	AR1_1_1	-0.31963	0.12546	-2.55	0.0132	y1(t-1)
	AR1_1_2	0.14599	0.54567	0.27	0.7899	y2(t-1)
	AR1_1_3	0.96122	0.66431	1.45	0.1526	y3(t-1)
	AR2_1_1	-0.16055	0.12491	-1.29	0.2032	y1(t-2)
	AR2_1_2	0.11460	0.53457	0.21	0.8309	y2(t-2)
	AR2_1_3	0.93439	0.66510	1.40	0.1647	y3(t-2)
y2	CONST2	0.01577	0.00437	3.60	0.0006	1
	AR1_2_1	0.04393	0.03186	1.38	0.1726	y1(t-1)
	AR1_2_2	-0.15273	0.13857	-1.10	0.2744	y2(t-1)
	AR1_2_3	0.28850	0.16870	1.71	0.0919	y3(t-1)
	AR2_2_1	0.05003	0.03172	1.58	0.1195	y1(t-2)
	AR2_2_2	0.01917	0.13575	0.14	0.8882	y2(t-2)
	AR2_2_3	-0.01020	0.16890	-0.06	0.9520	y3(t-2)
y3	CONST3	0.01293	0.00353	3.67	0.0005	1
	AR1_3_1	-0.00242	0.02568	-0.09	0.9251	y1(t-1)
	AR1_3_2	0.22481	0.11168	2.01	0.0482	y2(t-1)
	AR1_3_3	-0.26397	0.13596	-1.94	0.0565	y3(t-1)
	AR2_3_1	0.03388	0.02556	1.33	0.1896	y1(t-2)
	AR2_3_2	0.35491	0.10941	3.24	0.0019	y2(t-2)
	AR2_3_3	-0.02223	0.13612	-0.16	0.8708	y3(t-2)

Output 42.2.4 shows the innovation covariance matrix estimates, the various information criteria results, and the tests for white noise residuals. The residuals are uncorrelated except at lag 3 for $y2$ variable.

Output 42.2.4 Diagnostic Checks

Covariances of Innovations

Variable	y1	y2	y3
y1	0.00213	0.00007	0.00012
y2	0.00007	0.00014	0.00006
y3	0.00012	0.00006	0.00009

Information Criteria	
AICC	-1527.51
HQC	-1536.46
AIC	-1561.11
SBC	-1499.27
FPEC	2.18E-11

Cross Correlations of Residuals

Lag	Variable	y1	y2	y3
0	y1	1.00000	0.13242	0.28275
	y2	0.13242	1.00000	0.55526
	y3	0.28275	0.55526	1.00000
1	y1	0.01461	-0.00666	-0.02394
	y2	-0.01125	-0.00167	-0.04515
	y3	-0.00993	-0.06780	-0.09593
2	y1	0.07253	-0.00226	-0.01621
	y2	-0.08096	-0.01066	-0.02047
	y3	-0.02660	-0.01392	-0.02263
3	y1	0.09915	0.04484	0.05243
	y2	-0.00289	0.14059	0.25984
	y3	-0.03364	0.05374	0.05644

Schematic Representation of Cross Correlations of Residuals

Variable/Lag	0	1	2	3
y1	+.+	...	...	...
y2	.++	...	...	..+
y3	+++	...	...	...

+ is > 2*std error, - is < -2*std error, . is between

Portmanteau Test for Cross Correlations of Residuals

Up To Lag	DF	Chi-Square	Pr > ChiSq
3	9	9.69	0.3766

Output 42.2.5 describes how well each univariate equation fits the data. The residuals are off from the normality, but have no AR effects. The residuals for $y1$ variable have the ARCH effect.

Output 42.2.5 Diagnostic Checks Continued

Univariate Model ANOVA Diagnostics

Variable	R-Square	Standard Deviation	F Value	Pr > F
y1	0.1286	0.04615	1.62	0.1547
y2	0.1142	0.01172	1.42	0.2210
y3	0.2513	0.00944	3.69	0.0032

Univariate Model White Noise Diagnostics

		Normality		ARCH	
Variable	Durbin Watson	Chi-Square	Pr > ChiSq	F Value	Pr > F
y1	1.96269	10.22	0.0060	12.39	0.0008
y2	1.98145	11.98	0.0025	0.38	0.5386
y3	2.14583	34.25	<.0001	0.10	0.7480

Univariate Model AR Diagnostics

	AR1		AR2		AR3		AR4	
Variable	F Value	Pr > F	F Value	Pr > F	F Value	Pr > F	F Value	Pr > F
y1	0.01	0.9029	0.19	0.8291	0.39	0.7624	1.39	0.2481
y2	0.00	0.9883	0.00	0.9961	0.46	0.7097	0.34	0.8486
y3	0.68	0.4129	0.38	0.6861	0.30	0.8245	0.21	0.9320

Output 42.2.6 is the output in a matrix format associated with the PRINT=(IMPULSE=) option for the impulse response function and standard errors. The $y3$ variable in the first row is an impulse variable. The $y1$ variable in the first column is a response variable. The numbers, 0.96122, 0.41555, –0.40789 at lag 1 to 3 are decreasing.

Output 42.2.6 Impulse Response Function

Simple Impulse Response by Variable

Variable Response\Impulse	Lag	y1	y2	y3
y1	**1**	-0.31963	0.14599	0.96122
	STD	0.12546	0.54567	0.66431
	2	-0.05430	0.26174	0.41555
	STD	0.12919	0.54728	0.66311
	3	0.11904	0.35283	-0.40789
	STD	0.08362	0.38489	0.47867
y2	**1**	0.04393	-0.15273	0.28850
	STD	0.03186	0.13857	0.16870
	2	0.02858	0.11377	-0.08820
	STD	0.03184	0.13425	0.16250
	3	-0.00884	0.07147	0.11977
	STD	0.01583	0.07914	0.09462
y3	**1**	-0.00242	0.22481	-0.26397
	STD	0.02568	0.11168	0.13596
	2	0.04517	0.26088	0.10998
	STD	0.02563	0.10820	0.13101
	3	-0.00055	-0.09818	0.09096
	STD	0.01646	0.07823	0.10280

The proportions of decomposition of the prediction error covariances of three variables are given in Output 42.2.7. If you see the $y3$ variable in the first column, then the output explains that about 64.713% of the one-step-ahead prediction error covariances of the variable y_{3t} is accounted for by its own innovations, about 7.995% is accounted for by y_{1t} innovations, and about 27.292% is accounted for by y_{2t} innovations.

Output 42.2.7 Proportions of Prediction Error Covariance Decomposition

Proportions of Prediction Error Covariances by Variable

Variable	Lead	y1	y2	y3
y1	**1**	1.00000	0.00000	0.00000
	2	0.95996	0.01751	0.02253
	3	0.94565	0.02802	0.02633
	4	0.94079	0.02936	0.02985
	5	0.93846	0.03018	0.03136
	6	0.93831	0.03025	0.03145
y2	**1**	0.01754	0.98246	0.00000
	2	0.06025	0.90747	0.03228
	3	0.06959	0.89576	0.03465
	4	0.06831	0.89232	0.03937
	5	0.06850	0.89212	0.03938
	6	0.06924	0.89141	0.03935
y3	**1**	0.07995	0.27292	0.64713
	2	0.07725	0.27385	0.64890
	3	0.12973	0.33364	0.53663
	4	0.12870	0.33499	0.53631
	5	0.12859	0.33924	0.53217
	6	0.12852	0.33963	0.53185

The table in Output 42.2.8 gives forecasts and their prediction error covariances.

Output 42.2.8 Forecasts

Forecasts

Variable	Obs	Time	Forecast	Standard Error	95% Confidence Limits	
y1	**77**	**1979:1**	6.54027	0.04615	6.44982	6.63072
	78	**1979:2**	6.55105	0.05825	6.43688	6.66522
	79	**1979:3**	6.57217	0.06883	6.43725	6.70708
	80	**1979:4**	6.58452	0.08021	6.42732	6.74173
	81	**1980:1**	6.60193	0.09117	6.42324	6.78063
y2	**77**	**1979:1**	7.68473	0.01172	7.66176	7.70770
	78	**1979:2**	7.70508	0.01691	7.67193	7.73822
	79	**1979:3**	7.72206	0.02156	7.67980	7.76431
	80	**1979:4**	7.74266	0.02615	7.69140	7.79392
	81	**1980:1**	7.76240	0.03005	7.70350	7.82130
y3	**77**	**1979:1**	7.54024	0.00944	7.52172	7.55875
	78	**1979:2**	7.55489	0.01282	7.52977	7.58001
	79	**1979:3**	7.57472	0.01808	7.53928	7.61015
	80	**1979:4**	7.59344	0.02205	7.55022	7.63666
	81	**1980:1**	7.61232	0.02578	7.56179	7.66286

Output 42.2.9 shows that you cannot reject Granger noncausality from $(y2, y3)$ to $y1$ using the 0.05 significance level.

Output 42.2.9 Granger Causality Tests

Granger-Causality Wald Test			
Test	DF	Chi-Square	Pr > ChiSq
1	4	6.37	0.1734

Test 1:	Group 1 Variables:	y1
	Group 2 Variables:	y2 y3

The following SAS statements show that the variable $y1$ is the exogenous variable and fit the VARX(2,1) model to the data:

```
proc varmax data=use;
   id date interval=qtr;
   model y2 y3 = y1 / p=2 dify=(1) difx=(1) xlag=1 lagmax=3
                      print=(estimates diagnose);
run;
```

The fitted VARX(2,1) model is written as

$$\begin{pmatrix} \Delta y_{2t} \\ \Delta y_{3t} \end{pmatrix} = \begin{pmatrix} 0.01542 \\ 0.01319 \end{pmatrix} + \begin{pmatrix} 0.02520 \\ 0.05130 \end{pmatrix} \Delta y_{1t} + \begin{pmatrix} 0.03870 \\ 0.00363 \end{pmatrix} \Delta y_{1,t-1} + \begin{pmatrix} -0.12258 & 0.25811 \\ 0.24367 & -0.31809 \end{pmatrix} \begin{pmatrix} \Delta y_{2,t-1} \\ \Delta y_{3,t-1} \end{pmatrix} + \begin{pmatrix} 0.01651 & 0.03498 \\ 0.34921 & -0.01664 \end{pmatrix} \begin{pmatrix} \Delta y_{2,t-2} \\ \Delta y_{3,t-2} \end{pmatrix} + \begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \end{pmatrix}$$

The detailed output is shown in Output 42.2.10 through Output 42.2.13.

Output 42.2.10 shows the parameter estimates in terms of the constant, the current and the lag one coefficients of the exogenous variable, and the lag two coefficients of the dependent variables.

Output 42.2.10 Parameter Estimates

Analysis of German Economic Variables

The VARMAX Procedure

Type of Model	VARX(2,1)
Estimation Method	Least Squares Estimation

Constant

Variable	Constant
y2	0.01542
y3	0.01319

XLag

Lag	Variable	y1
0	**y2**	0.02520
	y3	0.05130
1	**y2**	0.03870
	y3	0.00363

AR

Lag	Variable	y2	y3
1	**y2**	-0.12258	0.25811
	y3	0.24367	-0.31809
2	**y2**	0.01651	0.03498
	y3	0.34921	-0.01664

Output 42.2.11 shows the parameter estimates and their significance.

Output 42.2.11 Parameter Estimates, Continued

Model Parameter Estimates

Equation	Parameter	Estimate	Standard Error	t Value	Pr > |t|	Variable
y2	**CONST1**	0.01542	0.00443	3.48	0.0009	1
	XL0_1_1	0.02520	0.03130	0.81	0.4237	y1(t)
	XL1_1_1	0.03870	0.03252	1.19	0.2383	y1(t-1)
	AR1_1_1	-0.12258	0.13903	-0.88	0.3811	y2(t-1)
	AR1_1_2	0.25811	0.17370	1.49	0.1421	y3(t-1)
	AR2_1_1	0.01651	0.13766	0.12	0.9049	y2(t-2)
	AR2_1_2	0.03498	0.16783	0.21	0.8356	y3(t-2)
y3	**CONST2**	0.01319	0.00346	3.81	0.0003	1
	XL0_2_1	0.05130	0.02441	2.10	0.0394	y1(t)
	XL1_2_1	0.00363	0.02536	0.14	0.8868	y1(t-1)
	AR1_2_1	0.24367	0.10842	2.25	0.0280	y2(t-1)
	AR1_2_2	-0.31809	0.13546	-2.35	0.0219	y3(t-1)
	AR2_2_1	0.34921	0.10736	3.25	0.0018	y2(t-2)
	AR2_2_2	-0.01664	0.13088	-0.13	0.8992	y3(t-2)

Output 42.2.12 shows the innovation covariance matrix estimates, the various information criteria results, and the tests for white noise residuals. The residuals is uncorrelated except at lag 3 for $y2$ variable.

Output 42.2.12 Diagnostic Checks

Covariances of Innovations		
Variable	**y2**	**y3**
y2	0.00014	0.00006
y3	0.00006	0.00009

Information Criteria	
AICC	-1182.33
HQC	-1177.94
AIC	-1193.46
SBC	-1154.52
FPEC	9.91E-9

Cross Correlations of Residuals			
Lag	**Variable**	**y2**	**y3**
0	**y2**	1.00000	0.56462
	y3	0.56462	1.00000
1	**y2**	-0.02312	-0.05927
	y3	-0.07056	-0.09145
2	**y2**	-0.02849	-0.05262
	y3	-0.05804	-0.08567
3	**y2**	0.16071	0.29588
	y3	0.10882	0.13002

Schematic Representation of Cross Correlations of Residuals				
Variable/Lag	**0**	**1**	**2**	**3**
y2	++	. .	. .	. +
y3	++	. .	. .	. .

+ is > 2*std error, - is < -2*std error, . is between

Portmanteau Test for Cross Correlations of Residuals			
Up To Lag	**DF**	**Chi-Square**	**Pr > ChiSq**
3	4	8.38	0.0787

Output 42.2.13 describes how well each univariate equation fits the data. The residuals are off from the normality, but have no ARCH and AR effects.

Output 42.2.13 Diagnostic Checks Continued

Univariate Model ANOVA Diagnostics

Variable	R-Square	Standard Deviation	F Value	Pr > F
y2	0.0897	0.01188	1.08	0.3809
y3	0.2796	0.00926	4.27	0.0011

Univariate Model White Noise Diagnostics

Variable	Durbin Watson	Normality Chi-Square	Normality Pr > ChiSq	ARCH F Value	ARCH Pr > F
y2	2.02413	14.54	0.0007	0.49	0.4842
y3	2.13414	32.27	<.0001	0.08	0.7782

Univariate Model AR Diagnostics

Variable	AR1 F Value	AR1 Pr > F	AR2 F Value	AR2 Pr > F	AR3 F Value	AR3 Pr > F	AR4 F Value	AR4 Pr > F
y2	0.04	0.8448	0.04	0.9570	0.62	0.6029	0.42	0.7914
y3	0.62	0.4343	0.62	0.5383	0.72	0.5452	0.36	0.8379

Example 42.3: Analysis of Restricted Cointegrated Systems

The structural relationships between economic time series have been of interest for decades. Because of the cointegration, the vector error correction model (VECM), introduced by Engle and Granger (1987), is one of the most important tools for performing such analysis. Although there exist analytical solutions for a nonrestricted VECM and some restricted VECMs in special forms, the estimation of a generally restricted VECM relies on numerical methods. This section illustrates how to use the RESTRICT (or BOUND) and TEST statements, together with the COINTEG statement, to estimate the restricted VECM and perform the statistical tests. For more information about this topic, see Boswijk and Doornik (2004) and references therein.

The data are simulated based on the VECM,

$$
\begin{aligned}
\Delta \mathbf{y}_t &= \boldsymbol{\alpha}\boldsymbol{\beta}'\mathbf{y}_{t-1} + \Phi_1^*\Delta\mathbf{y}_{t-1} + \Theta_0^*\mathbf{x}_t + \boldsymbol{\epsilon}_t \\
&= \begin{bmatrix} 0.01 & -0.02 \\ -0.03 & 0.04 \\ 0.05 & -0.06 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \mathbf{y}_{t-1} \\
&\quad + \begin{bmatrix} -0.01 & 0.03 & 0.05 & -0.02 \\ 0.02 & -0.04 & 0.06 & 0.03 \\ 0 & 0 & 0.10 & 0 \\ 0 & 0 & 0 & 0.04 \end{bmatrix} \Delta\mathbf{y}_{t-1} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \mathbf{x}_t + \boldsymbol{\epsilon}_t, \\
\boldsymbol{\epsilon}_t &\sim \text{iid } N(0, \Sigma), \Sigma = I_4
\end{aligned}
$$

where I_4 is the 4×4 identity matrix.

The following statements implement the simulation:

```
title 'Analysis of Restricted Cointegrated Systems';
proc iml;
   alpha = {0.01 -0.02, -0.03 0.04, 0.05 -0.06, 0 0};
   beta = {1 0, 0 1, -1 0, 0 -1};
   phiStar = {-0.01  0.03 0.05 -0.02,
               0.02 -0.04 0.06  0.03,
                  0     0 0.10     0,
                  0     0    0  0.04};
   Pi = alpha * beta` ;
   A1 = I(4) + Pi+ phiStar;
   A2 = -phiStar;
   phi = A1 // A2;
   sig = I(4);

   /* to simulate the vector time series */
   T = 600;
   myseed = 2;
   call varmasim(y,phi) sigma=sig n=T seed=myseed;
   x = J(T,1,0);
   do i = 1 to T;
      x[i] = normal(myseed);
   end;
   y = y || x;

   cn = {'y1' 'y2' 'y3' 'y4' 'x'};
   create simul5 from y[colname=cn];
   append from y;
   close;
quit;
```

Weak Exogeneity Tests

This example shows different methods for checking weak exogeneity.

The first method uses the EXOGENEITY option in the following statements, and the test results are shown in Output 42.3.1:

```
/* Method 1 -- To use the EXOGENEITY option */
ods output LogLikelihood = tbl_ll_g;
proc varmax data=simul5;
   model y1 y2 y3 y4 = x / noint p=2;
   cointeg rank=2 exogeneity;
run;
```

Output 42.3.1 Test Weak Exogeneity with the EXOGENEITY Option

Analysis of Restricted Cointegrated Systems

The VARMAX Procedure

Testing Weak Exogeneity of Each Variable

Variable	DF	Chi-Square	Pr > ChiSq
y1	2	102.96	<.0001
y2	2	116.12	<.0001
y3	2	200.80	<.0001
y4	2	3.99	0.1357

The second method uses the RESTRICT statement and then the likelihood ratio (LR) test in the following statements. The results are shown in Output 42.3.2. In theory, the first and second methods should have exactly same statistics and p-values because they implement the same LR tests. However, because of the difference between the analytical solution and the numerical solution for the restricted VECM, the statistics are a little different, although for the 0.05 significance level they lead to the same correct conclusion: the variable $y1$ is not the weak exogeneity of variables $y2$, $y3$, and $y4$; the variable $y2$ is not the weak exogeneity of variables $y1$, $y3$, and $y4$; the variable $y3$ is not the weak exogeneity of variables $y1$, $y2$, and $y4$; the variable $y4$ is the weak exogeneity of variables $y1$, $y2$, and $y3$.

```
/* Method 2 -- Use the RESTRICT statement and implement LR test */
%macro LRTestForVECM();
   %do i = 1 %to 4;
      ods output LogLikelihood = tbl_ll_r1_&i.;
      proc varmax data=simul5;
         model y1 y2 y3 y4 = x / noint p=2;
         cointeg rank=2;
         restrict alpha(&i.,1:2) = 0;
      run;
   %end;
   proc iml;
      use tbl_ll_g;
      read all var {nValue1} into ll_g;
      close;
      %do i = 1 %to 4;
         use tbl_ll_r1_&i.;
         read all var {nValue1} into ll_r_&i.;
         close;
      %end;
      DF = J(4,1,2);
      ll_r = ll_r_1 // ll_r_2 // ll_r_3 // ll_r_4;
      Stat = -2*(ll_r - ll_g);
      pValue = 1-cdf("CHISQUARE", Stat, DF);
      Test =  {"H0: Alpha(1,)=0"} // {"H0: Alpha(2,)=0"}
           // {"H0: Alpha(3,)=0"} // {"H0: Alpha(4,)=0"};
      print  Test DF Stat pValue;
   quit;
%mend;
%LRTestForVECM();
```

Output 42.3.2 Test Weak Exogeneity with the RESTRICT Statement and LR Tests

Analysis of Restricted Cointegrated Systems

Test	DF	Stat	pValue
H0: Alpha(1,)=0	2	109.05157	0
H0: Alpha(2,)=0	2	124.56535	0
H0: Alpha(3,)=0	2	238.35505	0
H0: Alpha(4,)=0	2	5.0877698	0.0785606

The third method uses the TEST statement, which implements the Wald tests. Asymptotically, the Wald test has the same distribution as the LR test.

```
/* Method 3 -- To use the TEST statement and the Wald test */
proc varmax data=simul5;
   model y1 y2 y3 y4 = x / noint p=2;
   cointeg rank=2;
   test alpha(1,1:2) = 0;
   test alpha(2,1:2) = 0;
   test alpha(3,1:2) = 0;
   test alpha(4,1:2) = 0;
run;
```

Based on the test results shown in Output 42.3.3, the same correct conclusion can be obtained at the 0.05 significance level: the variable $y1$ is not the weak exogeneity of variables $y2$, $y3$, and $y4$; the variable $y2$ is not the weak exogeneity of variables $y1$, $y3$, and $y4$; the variable $y3$ is not the weak exogeneity of variables $y1$, $y2$, and $y4$; the variable $y4$ is the weak exogeneity of variables $y1$, $y2$, and $y3$.

Output 42.3.3 Test Weak Exogeneity with the TEST Statement, Wald Tests

Analysis of Restricted Cointegrated Systems

The VARMAX Procedure

Testing of the Parameters			
Test	DF	Chi-Square	Pr > ChiSq
1	2	113.27	<.0001
2	2	129.15	<.0001
3	2	245.21	<.0001
4	2	4.81	0.0903

Identification

This example shows how important it is to identify $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ when applying the Wald test on $\boldsymbol{\alpha}$. First, in the following statements, there are no constraints on $\boldsymbol{\beta}$:

```
proc varmax data=simul5;
   model y1 y2 y3 y4 = x / noint p=2;
   cointeg rank=2;
   test alpha(1,2) = alpha(2,2) + alpha(3,2);
run;
```

As shown in Output 42.3.4, based on the test results, the null hypothesis H0: $\boldsymbol{\alpha}[1,2] = \boldsymbol{\alpha}[2,2] + \boldsymbol{\alpha}[3,2]$ should be rejected at the 0.05 significance level, although the true parameter values for the data generating process indicate that H0 is correct.

Output 42.3.4 Importance of Identifying $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in the Wald Test

Analysis of Restricted Cointegrated Systems

The VARMAX Procedure

Testing of the Parameters			
Test	DF	Chi-Square	Pr > ChiSq
1	1	21.44	<.0001

In the following statements, r^2 constraints are now imposed on $\boldsymbol{\beta}$, where r is the cointegration rank:

```
proc varmax data=simul5;
   model y1 y2 y3 y4 = x / noint p=2;
   cointeg rank=2;
   restrict beta(3:4,1:2) = -I(2);
   test alpha(1,2) = alpha(2,2) + alpha(3,2);
run;
```

As shown in Output 42.3.5, the null hypothesis cannot be rejected at 0.05 significance level; that is to say, the correct conclusion is achieved.

Output 42.3.5 Importance of Identifying $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in the Wald Test, Continued

Analysis of Restricted Cointegrated Systems

The VARMAX Procedure

Testing of the Parameters			
Test	DF	Chi-Square	Pr > ChiSq
1	1	0.16	0.6869

Besides $\boldsymbol{\alpha}$, other short-run parameters in a VECM can also be tested by using the TEST statement. Because short-run parameters other than $\boldsymbol{\alpha}$ are identified in a VECM, it is not necessary to impose additional constraints on $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The following statements test the null hypothesis H0: $\Phi_1^* = 0$:

```
proc varmax data=simul5;
   model y1 y2 y3 y4 = x / noint p=2;
   cointeg rank=2;
   test ar(2);
run;
```

According to the results shown in Output 42.3.6, the null hypothesis should be rejected at the 0.05 significance level.

Output 42.3.6 Wald Tests for Short-Run Parameters

Analysis of Restricted Cointegrated Systems

The VARMAX Procedure

Testing of the Parameters			
Test	DF	Chi-Square	Pr > ChiSq
1	16	32.79	0.0079

The following statements test the null hypothesis H0: $\Theta_0^* = 0$:

```
proc varmax data=simul5;
   model y1 y2 y3 y4 = x / noint p=2;
   cointeg rank=2;
   test xl;
run;
```

According to the results shown in Output 42.3.7, the null hypothesis cannot be rejected at the 0.05 significance level.

Output 42.3.7 Wald Tests for Short-Run Parameters, Continued

Analysis of Restricted Cointegrated Systems

The VARMAX Procedure

Testing of the Parameters			
Test	DF	Chi-Square	Pr > ChiSq
1	4	6.01	0.1982

Besides the parameters that are estimated in a VECM, you can also use the TEST statement on $\Pi(= \boldsymbol{\alpha}\boldsymbol{\beta}')$, and $\boldsymbol{\delta}_0$ or $\boldsymbol{\delta}_1$ for Case 2 or 4 when the constant or linear trend, respectively, is restricted in the error correction term. However, keep in mind that the covariance matrix for these parameter estimates is singular when the cointegration rank is less than the number of dependent variables; hence, you might not get the results for some tests.

```
proc varmax data=simul5;
   model y1 y2 y3 y4 = x / noint p=2;
   cointeg rank=2;
   test ar(1,4,1:4);
   test ar(1,4,{1 3});
run;
```

As shown in Output 42.3.8, the first test on H0: $\Pi[4,] = 0$ cannot be calculated, whereas the second test on H0: $\Pi[4,1] = \Pi[4,3] = 0$ can be.

Output 42.3.8 Wald Tests for Π

Analysis of Restricted Cointegrated Systems

The VARMAX Procedure

Testing of the Parameters			
Test	DF	Chi-Square	Pr > ChiSq
1	4		
2	2	4.81	0.0903

Tests for Long-Run Parameter

This example focuses on testing the relationships on the long-run parameter $\boldsymbol{\beta}$. Here, only the following specific form of hypothesis is discussed,

$$\text{H0: } \boldsymbol{\beta} = (\mathbf{H}, \boldsymbol{\phi})$$

where $\mathbf{H}$ is a known $k \times r_1$ matrix, $\boldsymbol{\phi}$ is a freely varying $k \times (r - r_1)$ parameter matrix, k is the number of dependent variables, r is the cointegration rank, and $0 \leq r_1 \leq r$. Other forms of hypothesis—for example, H0: $\boldsymbol{\beta} = (\mathbf{H}_1\boldsymbol{\phi}_1, \ldots, \mathbf{H}_r\boldsymbol{\phi}_r)$ or H0: $\mathbf{H}\text{vec}(\boldsymbol{\beta}) = \mathbf{h}$—are omitted, although they can also be implemented in the same logic. The following statements test the null hypothesis that $(1\ 0\ -1\ 0)'$ is in the cointegrating space that is spanned by $\boldsymbol{\beta}$:

```
/* Use the RESTRICT statement and LR test for restrictions on Beta. */
/* H0: Beta = [ H, phi ] where H is known and phi is free */
ods output LogLikelihood = tbl_ll_r2;
proc varmax data=simul5;
   model y1 y2 y3 y4 = x / noint p=2;
   cointeg rank=2;
   restrict beta(,1) = {1, 0, -1, 0};
   nloptions tech=qn maxit=5000;
run;

proc iml;
   use tbl_ll_g;
   read all var {nValue1} into ll_g;
   close;
   use tbl_ll_r2;
   read all var {nValue1} into ll_r;
   close;
   DF = (4-2)*1; /* DF = (k-r)*r_1 */
   Stat = -2*(ll_r - ll_g);
   pValue = 1-cdf("CHISQUARE", stat, df);
   Test = "H0: Beta[1,1:4] = {1 0 -1 0}'";
   print  Test DF Stat pValue;
quit;
```

According to the result shown in Output 42.3.9, the null hypothesis cannot be rejected at the 0.05 significance level.

Output 42.3.9 LR Tests on Long-Run Parameter β

Analysis of Restricted Cointegrated Systems

Test	DF	Stat	pValue
H0: Beta[1,1:4] = {1 0 -1 0}'	2	1.6194924	0.444971

The following statements test the null hypothesis that the cointegrating space is spanned by $(1\ 0\ -1\ 0, 0\ 1\ 0\ -1)'$:

```
/* H0: Beta = H, where H is the true Beta for DGP */
ods output LogLikelihood = tbl_ll_r3;
proc varmax data=simul5;
   model y1 y2 y3 y4 = x / noint p=2;
   cointeg rank=2;
   restrict beta = I(2) // (-I(2));
   nloptions tech=qn maxit=5000;
run;

proc iml;
   use tbl_ll_g;
   read all var {nValue1} into ll_g;
   close;
   use tbl_ll_r3;
   read all var {nValue1} into ll_r;
   close;
   DF = (4-2)*2; /* DF = (k-r)*r_1 */
   Stat = -2*(ll_r - ll_g);
   pValue = 1-cdf("CHISQUARE", stat, df);
   Test = "H0: Beta = {1 0, 0 1, -1 0, 0 -1}";
   print  Test DF Stat pValue;
quit;
```

According to the result shown in Output 42.3.10, the null hypothesis cannot be rejected at the 0.05 significance level.

Output 42.3.10 LR Tests on Long-Run Parameter β, Continued

Analysis of Restricted Cointegrated Systems

Test	DF	Stat	pValue
H0: Beta = {1 0, 0 1, -1 0, 0 -1}	4	1.5815435	0.8121055

The following statements test the null hypothesis that the cointegrating space is spanned by $(1\ 0\ 1\ 0, 0\ 1\ 0\ 1)'$, the orthogonal matrix to the true β for the data generating process:

```
/* H0: Beta = H, where H is the matrix orthogonal
       to the true Beta for DGP */
ods output LogLikelihood = tbl_ll_r4;
proc varmax data=simul5;
   model y1 y2 y3 y4 = x / noint p=2;
   cointeg rank=2;
```

```
   restrict beta = {1 0, 0 1, 1 0, 0 1};
   nloptions tech=qn maxit=5000;
run;

proc iml;
   use tbl_ll_g;
   read all var {nValue1} into ll_g;
   close;
   use tbl_ll_r4;
   read all var {nValue1} into ll_r;
   close;
   DF = (4-2)*2; /* DF = (k-r)*r_1 */
   Stat = -2*(ll_r - ll_g);
   pValue = 1-cdf("CHISQUARE", stat, df);
   Test = "H0: Beta = {1 0, 0 1, 1 0, 0 1}";
   print  Test DF Stat pValue;
quit;
```

According to the result shown in Output 42.3.11, the null hypothesis should be rejected at the 0.05 significance level.

Output 42.3.11 LR Tests on Long-Run Parameter β, Continued

Analysis of Restricted Cointegrated Systems

Test	DF	Stat	pValue
H0: Beta = {1 0, 0 1, 1 0, 0 1}	4	227.68902	0

For the VECM, the BOUND statement can be regarded as an alias of the RESTRICT statement; that is, you can directly replace any RESTRICT statement with a BOUND statement and get the same result. The linear inequality constraints in the restricted cointegrated systems are not discussed in this section, although they are also supported in the BOUND and RESTRICT statements. For more information, see the sections "BOUND Statement" on page 2996 and "RESTRICT Statement" on page 3025.

Obtaining the numerical solution for the restricted VECM is not an easy task in most cases. You might need to use the INITIAL and NLOPTIONS statements to tune the process. For more information, see the sections "INITIAL Statement" on page 3006 and "NLOPTIONS Statement" on page 3024.

Example 42.4: Analysis of Euro Foreign Exchange Reference Rates

This example illustrates how to use and select the VARMA-GARCH model for exchange rates, a general type of financial data. As shown in much of the literature, the financial variables are cross-correlated and autocorrelated not only on first moments, but also on second moments. The VARMA-GARCH model and the vector error correction GARCH model are often used to catch the stylized fact.

The data, downloaded from European Central Bank website (https://www.ecb.europa.eu), consist of four pairs of daily foreign exchange reference rates: the euro and the Australian dollar (AUD), the euro and the British pound sterling (GBP), the euro and the Japanese yen (JPY), and the euro and the US dollar (USD). The full sample covers the period from January 4, 1999, to February 12, 2015 (4,127 days). In the following statements, the series are logarithmically transformed, and the returns (in percentage) are calculated:

```
title 'Analysis of Euro Foreign Exchange Reference Rates';
data eurofxrr;
   input date : MMDDYY10. aud gbp jpy usd;
   label aud='The euro and the Australian dollar'
         usd='The euro and the U.S. dollar'
         jpy='The euro and the Japanese yen'
         gbp='The euro and the British pound sterling';
   logAUD = log(AUD); logGBP = log(GBP);
   logJPY = log(JPY); logUSD = log(USD);
   rAUD = (logAUD - lag(logAUD))*100;
   rGBP = (logGBP - lag(logGBP))*100;
   rJPY = (logJPY - lag(logJPY))*100;
   rUSD = (logUSD - lag(logUSD))*100;
datalines;
01/04/1999    1.9100    0.71110    133.73    1.1789
01/05/1999    1.8944    0.71220    130.96    1.1790
01/06/1999    1.8820    0.70760    131.42    1.1743
01/07/1999    1.8474    0.70585    129.43    1.1632

   ... more lines ...

02/10/2015    1.4522    0.74200    134.67    1.1297
02/11/2015    1.4606    0.73960    135.50    1.1314
02/12/2015    1.4761    0.73760    135.72    1.1328
;
```

Although it is well known that unit roots exist in the exchange rate series and they are not cointegrated, you can use the following statements to verify:

```
/*--- Unit Roots and Cointegration in Log Exchange Rates ---*/

proc varmax data=eurofxrr;
   model logAUD logGBP logJPY logUSD / p=2 dftest cointtest;
run;
```

According to the results of the Dickey-Fuller unit root tests shown in Output 42.4.1, the null hypothesis that there is a unit root in each series cannot be rejected at the 5% significance level. The results of the Johansen cointegration rank trace tests shown in Output 42.4.2 confirm that there is no cointegration between series because the null hypothesis that the cointegration rank is 0, in both unrestricted and restricted cases, cannot be rejected at the 5% significance level. Because there is no cointegration, you do not need to consider vector error correction models; otherwise, the final selected model might be a vector error correction GARCH model, instead of a VARMA-GARCH model.

Output 42.4.1 Dickey-Fuller Unit Root Tests

Analysis of Euro Foreign Exchange Reference Rates

The VARMAX Procedure

Unit Root Test					
Variable	Type	Rho	Pr < Rho	Tau	Pr < Tau
logAUD	Zero Mean	-1.05	0.4644	-1.08	0.2549
	Single Mean	-9.44	0.1549	-2.31	0.1683
	Trend	-13.85	0.2287	-2.63	0.2657
logGBP	Zero Mean	-0.57	0.5554	-0.59	0.4630
	Single Mean	-3.23	0.6297	-1.27	0.6445
	Trend	-11.11	0.3666	-2.27	0.4502
logJPY	Zero Mean	0.00	0.6836	0.02	0.6894
	Single Mean	-6.11	0.3394	-1.73	0.4140
	Trend	-6.56	0.7000	-1.83	0.6901
logUSD	Zero Mean	-1.46	0.4014	-0.88	0.3346
	Single Mean	-3.29	0.6216	-1.27	0.6471
	Trend	-5.76	0.7638	-1.47	0.8394

Output 42.4.2 Johansen Cointegration Rank Trace Tests

Cointegration Rank Test Using Trace						
H0: Rank=r	H1: Rank>r	Eigenvalue	Trace	Pr > Trace	Drift in ECM	Drift in Process
0	0	0.0059	36.6836	0.3601	Constant	Linear
1	1	0.0018	12.1427	0.9269		
2	2	0.0008	4.7724	0.8319		
3	3	0.0003	1.3036	0.2532		

Cointegration Rank Test Using Trace Under Restriction						
H0: Rank=r	H1: Rank>r	Eigenvalue	Trace	Pr > Trace	Drift in ECM	Drift in Process
0	0	0.0060	37.1246	0.6151	Constant	Constant
1	1	0.0018	12.1792	0.9921		
2	2	0.0008	4.7941	0.9855		
3	3	0.0003	1.3041	0.9066		

Before modeling returns, you can test whether unit roots still exist in the differenced data with the following statement:

```
/*--- Unit Roots in Returns and Model Specification ---*/
proc varmax data=eurofxrr;
   model rAUD rGBP rJPY rUSD / p=2 dftest;
   test const; test ar(1); test ar(2);
run;
```

Output 42.4.3 shows that there is no unit root in each differenced series.

Output 42.4.3 Dickey-Fuller Unit Root Tests

Analysis of Euro Foreign Exchange Reference Rates

The VARMAX Procedure

Unit Root Test					
Variable	Type	Rho	Pr < Rho	Tau	Pr < Tau
rAUD	Zero Mean	-4242.7	0.0001	-46.04	<.0001
	Single Mean	-4243.7	0.0001	-46.04	<.0001
	Trend	-4244.2	0.0001	-46.04	<.0001
rGBP	Zero Mean	-4358.4	0.0001	-46.67	<.0001
	Single Mean	-4358.4	0.0001	-46.67	<.0001
	Trend	-4358.5	0.0001	-46.66	<.0001
rJPY	Zero Mean	-4181.4	0.0001	-45.72	<.0001
	Single Mean	-4181.4	0.0001	-45.72	<.0001
	Trend	-4181.9	0.0001	-45.72	<.0001
rUSD	Zero Mean	-4306.8	0.0001	-46.40	<.0001
	Single Mean	-4306.8	0.0001	-46.39	<.0001
	Trend	-4307.4	0.0001	-46.39	<.0001

The preceding statements also test whether the constant and each of two lags of AR terms are 0. The test results are shown in Output 42.4.4.

Output 42.4.4 Tests on Constant and AR Terms

Testing of the Parameters			
Test	DF	Chi-Square	Pr > ChiSq
1	4	0.46	0.9776
2	16	59.42	<.0001
3	16	15.67	0.4759

The null hypothesis that the constant term is 0 and the null hypothesis that the second lag AR term is 0 are both accepted at the 5% significance level. However, the null hypothesis that the first lag AR term is 0 is rejected at the 5% significance level. In the remaining model selection process, only the first lag AR term is considered.

The following statements estimate a zero-mean VAR(1) model and also print some diagnostic results:

```
/*--- VAR Model ---*/

proc varmax data=eurofxrr;
   model rAUD rGBP rJPY rUSD / noint p=1 print=(diagnose);
run;
```

Output 42.4.5 shows the information criteria for the estimated zero-mean VAR(1) model. In this example, AICC is used as the criterion for model selection: the smaller the AICC, the better the model.

Output 42.4.5 Information Criteria for the VAR Model

Analysis of Euro Foreign Exchange Reference Rates

The VARMAX Procedure

Information Criteria	
AICC	-1745.29
HQC	-1687.44
AIC	-1745.64
SBC	-1581.19
FPEC	0.011938

Diagnostics are printed because the PRINT=(DIAGNOSE) option is specified. As shown in Output 42.4.6, the null hypotheses that there is no ARCH effect in each series are all rejected at the 5% significance level.

Output 42.4.6 Tests on ARCH Effects

Univariate Model White Noise Diagnostics

		Normality		ARCH	
Variable	Durbin Watson	Chi-Square	Pr > ChiSq	F Value	Pr > F
rAUD	1.99811	8277.31	<.0001	217.35	<.0001
rGBP	1.99601	2537.71	<.0001	315.25	<.0001
rJPY	2.00007	2456.22	<.0001	149.75	<.0001
rUSD	1.99959	1398.54	<.0001	157.85	<.0001

To find the right GARCH model, you can start with the VAR(1)-CCC-GARCH(1,1) model (which is usually the fastest one to be estimated) as in the following statement:

```
/*--- VAR CCC GARCH Model ---*/

proc varmax data=eurofxrr;
   model rAUD rGBP rJPY rUSD / noint p=1;
   garch p=1 q=1 form=ccc;
run;
```

Compared to the AICC for the zero-mean VAR(1) model (shown in Output 42.4.5), the AICC for VAR(1)-CCC-GARCH(1,1) model, as shown in Output 42.4.7, dramatically decreases, which means that the ARCH effects do play an important role and should be modeled.

Output 42.4.7 Information Criteria for VAR CCC GARCH Model

Analysis of Euro Foreign Exchange Reference Rates

The VARMAX Procedure

Information Criteria	
AICC	-4646.77
HQC	-4571.24
AIC	-4647.35
SBC	-4432.31
FPEC	0.011966

As indicated by its name, a basic assumption of the CCC GARCH model is that the conditional correlation is time-invariant, which might not be true. The following statements estimate a BEKK GARCH model to see whether modeling the conditional correlation could improve the model performance:

```
/*---  VAR BEKK GARCH Model ---*/

proc varmax data=eurofxrr outest=oediagbekk;
   model rAUD rGBP rJPY rUSD / noint p=1;
   garch p=1 q=1 form=bekk;
run;
```

As shown in Output 42.4.8, the AICC for the VAR BEKK GARCH model does get smaller than the AICC for the CCC GARCH model (shown in Output 42.4.7). The smaller AICC implies that the assumption of the CCC GARCH model might be inaccurate.

Output 42.4.8 Information Criteria for VAR BEKK GARCH Model

Analysis of Euro Foreign Exchange Reference Rates

The VARMAX Procedure

Information Criteria	
AICC	-5667.7
HQC	-5539.55
AIC	-5669.38
SBC	-5302.54
FPEC	0.011979

One shortcoming of the BEKK GARCH model is that it has too many parameters. In practice, especially for a large number of dependent variables, the scalar BEKK GARCH model and the diagonal BEKK GARCH model are often applied, as shown in the following statements. In the RESTRICT statement, matrix operations are used; using matrix operations is much more concise than restricting tens of ARCH and GARCH parameters one by one.

```
/*--- VAR Scalar BEKK GARCH Model ---*/

proc varmax data=eurofxrr outest=oediagbekk;
   model rAUD rGBP rJPY rUSD / noint p=1;
   garch p=1 q=1 form=bekk;
   restrict ach(1)=ach(1,1,1)*I(4), gch(1)=gch(1,1,1)*I(4);
run;

/*--- VAR Diagonal BEKK GARCH Model ---*/

proc varmax data=eurofxrr outest=oediagbekk;
   model rAUD rGBP rJPY rUSD / noint p=1;
   garch p=1 q=1 form=bekk;
   restrict ach(1)=ach(1)#I(4), gch(1)=gch(1)#I(4);
run;
```

The AICCs for the scalar and diagonal BEKK GARCH models are shown in Output 42.4.9 and Output 42.4.10, respectively, and both of them are larger than the AICC for the BEKK GARCH model (shown in Output 42.4.8). Hence, so far, the VAR BEKK GARCH model is the best.

Output 42.4.9 Information Criteria for VAR Scalar BEKK GARCH Model

Analysis of Euro Foreign Exchange Reference Rates

The VARMAX Procedure

Information Criteria	
AICC	-5615.11
HQC	-5552.83
AIC	-5615.51
SBC	-5438.41
FPEC	0.011974

Output 42.4.10 Information Criteria for VAR Diagonal BEKK GARCH Model

Analysis of Euro Foreign Exchange Reference Rates

The VARMAX Procedure

Information Criteria	
AICC	-5630.31
HQC	-5554.78
AIC	-5630.89
SBC	-5415.85
FPEC	0.011978

Another type of multivariate GARCH model that is suitable for modeling the time-varying conditional correlation is the dynamic conditional correlation (DCC) GARCH model, as indicated by its name. The following statements estimate the DCC GARCH model:

```
/*--- VAR DCC GARCH Model ---*/

proc varmax data=eurofxrr;
   model rAUD rGBP rJPY rUSD / noint p=1;
   garch p=1 q=1 form=dcc;
run;
```

As shown in Output 42.4.11, the AICC for the VAR DCC GARCH model is smaller than the AICC for the VAR BEKK GARCH model (shown in Output 42.4.8), implying that the best model should be in the class of DCC GARCH models.

Output 42.4.11 Information Criteria for VAR DCC GARCH Model

Analysis of Euro Foreign Exchange Reference Rates

The VARMAX Procedure

Information Criteria	
AICC	-5689.43
HQC	-5609.5
AIC	-5690.08
SBC	-5462.39
FPEC	0.011973

Could the DCC GARCH model be more parsimonious? The following statements use the sample correlation matrix of the standardized residuals (saving six parameters) to calculate the unconditional correlation matrix in the DCC GARCH model:

```
/*--- Parsimonious VAR DCC GARCH Model ---*/

proc varmax data=eurofxrr;
   model rAUD rGBP rJPY rUSD / noint p=1;
   garch p=1 q=1 form=dcc corrconst=expect;
run;
```

The AICC of the parsimonious VAR DCC GARCH model, as shown in Output 42.4.12, becomes a little smaller. Hence, the best model so far is the parsimonious VAR DCC GARCH model.

Output 42.4.12 Information Criteria for the Parsimonious VAR DCC GARCH Model

Analysis of Euro Foreign Exchange Reference Rates

The VARMAX Procedure

Information Criteria	
AICC	-5694.89
HQC	-5628.19
AIC	-5695.35
SBC	-5505.6
FPEC	0.011973

Another way to refine the model is to try different subforms of GARCH models for each series. The following statements estimate the VAR DCC EGARCH model and produce Output 42.4.13:

```
/*--- VAR DCC EGARCH Model ---*/

proc varmax data=eurofxrr;
   model rAUD rGBP rJPY rUSD / noint p=1;
   garch p=1 q=1 form=dcc subform=egarch corrconst=expect;
   nloptions maxit=5000 pall;
run;
```

The following statements estimate the VAR DCC PGARCH model and produce Output 42.4.14:

```
/*--- VAR DCC PGARCH Model ---*/

proc varmax data=eurofxrr;
   model rAUD rGBP rJPY rUSD / noint p=1;
   garch p=1 q=1 form=dcc subform=pgarch corrconst=expect;
   nloptions maxit=5000 pall;
run;
```

The following statements estimate the VAR DCC QGARCH model and produce Output 42.4.15:

```
/*--- VAR DCC QGARCH Model ---*/

proc varmax data=eurofxrr;
   model rAUD rGBP rJPY rUSD / noint p=1;
   garch p=1 q=1 form=dcc subform=qgarch corrconst=expect;
   nloptions maxit=5000 pall;
run;
```

The following statements estimate the VAR DCC TGARCH model and produce Output 42.4.16:

```
/*--- VAR DCC TGARCH Model ---*/

proc varmax data=eurofxrr;
   model rAUD rGBP rJPY rUSD / noint p=1;
   garch p=1 q=1 form=dcc subform=tgarch corrconst=expect;
   nloptions maxit=5000 pall;
run;
```

Comparing the AICCs shown in Output 42.4.13 through Output 42.4.16, you find that the AICC for the VAR DCC PGARCH model is the smallest. Hence, the best model becomes the zero-mean VAR(1)-DCC-PGARCH(1,1) model, whose unconditional correlation matrix is estimated by the sample correlation matrix of the standardized residuals.

Output 42.4.13 Information Criteria for the Parsimonious VAR DCC EGARCH Model

Analysis of Euro Foreign Exchange Reference Rates

The VARMAX Procedure

Information Criteria	
AICC	-5704.33
HQC	-5628.81
AIC	-5704.92
SBC	-5489.87
FPEC	0.011982

Output 42.4.14 Information Criteria for the Parsimonious VAR DCC PGARCH Model

Analysis of Euro Foreign Exchange Reference Rates

The VARMAX Procedure

Information Criteria	
AICC	-5724.44
HQC	-5640.1
AIC	-5725.16
SBC	-5484.82
FPEC	0.011974

Output 42.4.15 Information Criteria for the Parsimonious VAR DCC QGARCH Model

Analysis of Euro Foreign Exchange Reference Rates

The VARMAX Procedure

Information Criteria	
AICC	-5696.97
HQC	-5621.44
AIC	-5697.55
SBC	-5482.51
FPEC	0.011972

Output 42.4.16 Information Criteria for the Parsimonious VAR DCC TGARCH Model

Analysis of Euro Foreign Exchange Reference Rates

The VARMAX Procedure

Information Criteria	
AICC	-5705.59
HQC	-5630.06
AIC	-5706.17
SBC	-5491.13
FPEC	0.011973

Output 42.4.17 shows that most of the AR parameter estimates in the VAR DCC PGARCH model are not significant.

Output 42.4.17 AR Parameter Estimates for the Parsimonious VAR DCC PGARCH Model

Model Parameter Estimates

Equation	Parameter	Estimate	Standard Error	t Value	Pr > |t|	Variable
rAUD	AR1_1_1	0.05718	0.01790	3.19	0.0014	rAUD(t-1)
	AR1_1_2	0.00042	0.02396	0.02	0.9859	rGBP(t-1)
	AR1_1_3	-0.02305	0.01619	-1.42	0.1546	rJPY(t-1)
	AR1_1_4	0.02005	0.02020	0.99	0.3211	rUSD(t-1)
rGBP	AR1_2_1	0.02686	0.01147	2.34	0.0193	rAUD(t-1)
	AR1_2_2	0.04512	0.01880	2.40	0.0164	rGBP(t-1)
	AR1_2_3	-0.00462	0.01138	-0.41	0.6845	rJPY(t-1)
	AR1_2_4	-0.04651	0.01475	-3.15	0.0016	rUSD(t-1)
rJPY	AR1_3_1	0.05602	0.01845	3.04	0.0024	rAUD(t-1)
	AR1_3_2	-0.05011	0.02697	-1.86	0.0632	rGBP(t-1)
	AR1_3_3	-0.00181	0.01893	-0.10	0.9240	rJPY(t-1)
	AR1_3_4	-0.00839	0.02226	-0.38	0.7061	rUSD(t-1)
rUSD	AR1_4_1	0.03852	0.01513	2.55	0.0109	rAUD(t-1)
	AR1_4_2	0.00566	0.02290	0.25	0.8048	rGBP(t-1)
	AR1_4_3	0.00084	0.01477	0.06	0.9548	rJPY(t-1)
	AR1_4_4	-0.03202	0.02011	-1.59	0.1115	rUSD(t-1)

The following statements test the significance of some parameter estimates:

```
/*--- Significance Of Some Parameter Estimates ---*/

proc varmax data=eurofxrr;
   model rAUD rGBP rJPY rUSD / noint p=1;
   garch p=1 q=1 form=dcc subform=pgarch corrconst=expect;
   nloptions maxit=5000 pall;
   test ar(1, 1, 2:4), ar(1, 2, 3), ar(1, 3, 3:4), ar(1, 4, 2:4);
run;
```

As shown in Output 42.4.18, the null hypothesis that all nine of the parameters in the TEST statement are 0 cannot be rejected at the 5% significance level.

Output 42.4.18 Test on Significance of Some Parameter Estimates

Analysis of Euro Foreign Exchange Reference Rates

The VARMAX Procedure

Testing of the Parameters			
Test	DF	Chi-Square	Pr > ChiSq
1	9	7.36	0.6002

The following statements estimate the VAR DCC PGARCH model without those insignificant parameters:

```
/*--- VAR DCC PGARCH Model w/o Insignificant Parameters ---*/

proc varmax data=eurofxrr;
   model rAUD rGBP rJPY rUSD / noint p=1;
   garch p=1 q=1 form=dcc subform=pgarch corrconst=expect;
   nloptions maxit=5000 pall;
   restrict ar(1, 1, 2:4), ar(1, 2, 3), ar(1, 3, 3:4), ar(1, 4, 2:4);
run;
```

As shown in Output 42.4.19, the AICC does improve and decrease. Further refining the model is possible but beyond the scope of this example. Hence, the best model, according to the AICC, is the zero-mean VAR(1)-DCC-PGARCH(1,1) model without insignificant AR parameters, and its unconditional correlation matrix is estimated by the sample correlation matrix of the standardized residuals.

Output 42.4.19 Information Criteria for the VAR DCC PGARCH Model without Insignificant Parameters

Analysis of Euro Foreign Exchange Reference Rates

The VARMAX Procedure

Information Criteria	
AICC	-5735.05
HQC	-5670.56
AIC	-5735.48
SBC	-5552.06
FPEC	0.011996

This example focuses only on using the information criterion to distinguish models. In practice, the forecast performance of the model might be more important. The VARMAX procedure supports multistep forecasting in both VARMAX-GARCH models and vector error correction GARCH models. Hence, although it is not covered in this example, you can also use the VARMAX procedure and a criterion based on out-of-sample forecast to perform model selection.

Example 42.5: Conditional Forecasts and Scenario Analysis

Conditional forecasts incorporate future information and the uncertainty of parameters in the forecasts, and they often provide more accurate forecasts than unconditional forecasts do. Clark and McCracken (2017) evaluate conditional forecasts and focus on tests of bias, efficiency, and equal accuracy applied to conditional forecasts from VAR models. In this example, a Monte Carlo experiment is created in order to compare different types of forecasts; that is, 1,000 data sets are generated and the following forecasts are compared: equation-based unconditional forecasts, simulation-based unconditional forecasts, simulation-based conditional forecasts under hard conditions, and simulation-based conditional forecasts under soft conditions.

Consider the following trivariate VAR(2) model:

$$\mathbf{y}_t = \mathbf{c} + \mathbf{A}_1\mathbf{y}_{t-1} + \mathbf{A}_2\mathbf{y}_{t-2} + \boldsymbol{\epsilon}_t, \boldsymbol{\epsilon}_t \sim N(0, \Sigma)$$

where

$$\mathbf{c} = \begin{pmatrix} 2.425 \\ 0.054 \\ -0.110 \end{pmatrix}, \quad \mathbf{A}_1 = \begin{pmatrix} 0.234 & -0.134 & -0.057 \\ 0.029 & 0.575 & 0.200 \\ 0.059 & 0.038 & 1.006 \end{pmatrix}$$

$$\mathbf{A}_2 = \begin{pmatrix} 0.164 & -0.150 & -0.165 \\ -0.039 & 0.138 & -0.184 \\ 0.031 & 0.019 & -0.087 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 9.265 & 0.296 & 0.553 \\ 0.296 & 1.746 & 0.184 \\ 0.553 & 0.184 & 0.752 \end{pmatrix}$$

As indicated in Clark and McCracken (2017), the parameter values are equal to the OLS (ordinary least squares) estimates of a VAR in GDP (gross domestic product) growth, inflation less a survey-based measure of trend inflation, and the federal funds rate less a survey-based measure of trend, over a sample from 1961 to 2007. Many central banks require forecasts conditional on particular paths of policy instruments. This example analyzes different scenarios of some known future information on the third variable that is related to federal funds rate.

Conditional Forecasts and Scenario Analysis under Hard Conditions

This section considers the hard conditions, under which some future dependent variables are fixed to some single values.

The following macro generates the data for analysis:

```
title 'Conditional Forecasts and Scenario Analysis';
%macro cfSimulateData(dgpi,T,lead,tblDGP,tblSample);
   * dgpi: index of DGP;
   * T: in-sample sample size;
   * lead: future horizons;
   * tblDGP: output table name for full-sample data;
   * tblSample: output table name for in-sample data;
   proc iml;
      * simulate the data;
      * trivariate VAR(2) model;
      seed = 12345 + &dgpi.; * random seed;
      n = 3;                 * dim of dependent variable;
```

```
      T = &T.;               * in-sample sample size;
      lead = &lead.;         * future horizons;
      p = 2;                 * AR order;
      * parameter values;
      const = {2.425, 0.054, -0.110};
      phi   = {0.234 -0.134 -0.057,
               0.029  0.575  0.200,
               0.059  0.038  1.006,
               0.164 -0.150 -0.165,
              -0.039  0.138 -0.184,
               0.031  0.019 -0.087};
      sigma = {9.265  0.296  0.553,
               0.296  1.746  0.184,
               0.553  0.184  0.752};
      call varmasim(y,phi) sigma = sigma n = T+lead seed = seed;
      mu = (inv(I(3)-phi[1:3,]-phi[4:6,])*const)`;
      y = y + mu;
      name={y1 y2 y3};
      create &tblDGP. from y[colname=name];
      append from y;
      close;
   quit;
   data &tblSample.; set &tblDGP.(obs=&T.); run;
%mend;
```

The following macro constructs four scenarios that contain hard conditions. In scenario $i, i = 1, \ldots, 4$, the first i future values of y3 are fixed (to their true values). In the real world, the true future values cannot be known. Here, the true future values are used so that you can check later whether using more information results in any advantage in conditional forecasts.

```
%macro hcConstructScenarios(T,tblDGP,tblScenarios);
   * T: in-sample sample size;
   * tblDGP: input table name for full-sample data;
   * tblScenarios: output table name for scenarios;
   data &tblScenarios.;
      set &tblDGP.(firstobs=%eval(&T.+1) obs=%eval(&T.+1))
          &tblDGP.(firstobs=%eval(&T.+1) obs=%eval(&T.+2))
          &tblDGP.(firstobs=%eval(&T.+1) obs=%eval(&T.+3))
          &tblDGP.(firstobs=%eval(&T.+1) obs=%eval(&T.+4))
          &tblDGP.(firstobs=%eval(&T.+1) obs=%eval(&T.+1));
      * scenario 1: y3(1) is known;
      if(_N_=1) then do;
         y1=.; y2=.; myscenario=1;
      end;
      * scenario 2: y3(1:2) is known;
      if(_N_>1 and _N_<=3) then do;
         y1=.; y2=.; myscenario=2;
      end;
      * scenario 3: y3(1:3) is known;
      if(_N_>3 and _N_<=6) then do;
         y1=.; y2=.; myscenario=3;
      end;
      * scenario 4: y3(1:4) is known;
      if(_N_>6 and _N_<=10) then do;
```

```
         y1=.; y2=.; myscenario=4;
      end;
      * scenario 5: nothing is known (unconditional forecast);
      if(_N_>10) then do;
         y1=.; y2=.; y3=.; myscenario=5;
      end;
   run;
%mend;
```

The following macro estimates and performs several types of forecasts. The equation-based forecasts are output to the OUT= data set that is specified in the OUTPUT statement. The conditional forecasts for scenarios 1 to 4 and the unconditional forecasts (for scenario 5) are output to the OUT= data set that is specified in the CONDFORE statement.

```
%macro hcEstimateAndForecast(tblSample,tblScenarios,alpha,lead,nmc,
   tblF,tblCf,tblCfSim);
   * tblSample: input table name for in-sample data;
   * tblScenarios: input table name for scenarios;
   * alpha: size of the confidence interval or credible interval;
   * lead: future horizons;
   * nmc: number of Monte Carlo iterations;
   * tblF: output table name of equation-based point and interval forecasts;
   * tblCf: output table name of conditionial point and interval forecasts;
   * tblCfSim: output table name of simulated conditional forecasts;
   proc varmax data=&tblSample.;
      model y1 - y3 / p=2 prior noprint;
      output alpha=&alpha. lead=&lead. out=&tblF. noprint;
      condfore alpha=&alpha. lead=&lead. out=&tblCf. outsim=&tblCfSim.
         sdata=&tblScenarios. sid=myscenario
         parm=sampling(scenario) nbi=1000 nmc=&nmc.;
   run;
%mend;
```

The following macro saves all types of forecasts for one simulated data set to one data set for evaluation:

```
%macro hcSaveForecasts(dgpi,T,lead,nScenarios,tblDGP,tblF,tblCf,tblAll);
   * dgpi: index of DGP;
   * T: in-sample sample size;
   * lead: future horizons;
   * nScenarios: number of scenarios;
   * tblDGP: input table name for full-sample data;
   * tblF: input table name of equation-based point and interval forecasts;
   * tblCf: input table name of conditionial point and interval forecasts;
   * tblAll: output table name of point and interval forecasts for all DGPs;
   data forecasts;
      set &tblDGP.(firstobs=%eval(&T.+1) obs=%eval(&T.+&lead.) keep=Y1 Y2);
      * for notation convenience, name equation-based forecasts as S0;
      set &tblF.(firstobs=%eval(&T.+1) obs=%eval(&T.+&lead.)
         rename=( for1=Y1_S0 for2=Y2_S0
                  lci1=Y1_LB_S0 uci1=Y1_UB_S0
                  lci2=Y2_LB_S0 uci2=Y2_UB_S0)
         keep=FOR1 LCI1 UCI1 FOR2 LCI2 UCI2);
      %do i = 1 %to &nScenarios.;
         set &tblCf.(where=(myscenario_S&i.=&i.)
```

```
            rename=( Y1_MEDIAN=Y1_S&i. Y2_MEDIAN=Y2_S&i.
                     Y1_LB=Y1_LB_S&i. Y1_UB=Y1_UB_S&i.
                     Y2_LB=Y2_LB_S&i. Y2_UB=Y2_UB_S&i.
                     myscenario=myscenario_S&i.)
            keep=myscenario Y1_MEDIAN Y2_MEDIAN Y1_LB Y1_UB Y2_LB Y2_UB);
         %end;
         dgpIndex = &dgpi.;
         h = _N_;
         drop myscenario_S1 - myscenario_S&nScenarios.;
      run;
      proc append base=&tblAll. data=forecasts; run;
   %mend;
```

The following macro evaluates the forecasts from different methods and under different conditions. The accuracy of point forecasts is measured through the symmetric mean absolute percentage error (sMAPE), which is used in the M4 Forecasting Competition. The sMAPE is defined as

$$\text{sMAPE} = \frac{1}{h}\sum_{i=1}^{h}\frac{2|\mathrm{y}_{T+i} - \hat{\mathrm{y}}_{T+i}|}{|\mathrm{y}_{T+i}| + |\hat{\mathrm{y}}_{T+i}|}$$

where T is the in-sample sample size, y_{T+i} is the future value at $T+i$, $\hat{\mathrm{y}}_{T+i}$ is the ith-step-ahead forecast, and h is the forecasting horizon. The smaller the sMAPE, the better the forecasting method. In order to easily compare the sMAPEs, the relative sMAPEs are calculated. The simulation-based unconditional forecasts are used as the benchmark. As for the interval forecasts (that is, the confidence interval for equation-based forecasts and the credible interval for simulation-based conditional and unconditional forecasts), first the size is checked, and then the lengths of intervals are compared: if the size is correct, the smaller the interval length, the more accurate and better the forecasting method.

```
%macro cfEvaluate(tblAll,lead,nSim,nScenarios,qScenario0,scenarioBM,tblEval);
   * tblAll: input table name of point and interval forecasts for all DGPs;
   * lead: future horizons;
   * nSim: number of DGPs;
   * nScenarios: number of scenarios;
   * qScenario0: whether there is S0 for equation-based forecasts,
                 1 for yes and 0 for no;
   * scenarioBM: the index of the benchmark scenario;
   * tblEval: output table name for evaluation results;
   proc iml;
      use &tblAll.;
      read all into d;
      close;
      lead = &lead.;
      nSim = &nSim.;
      nScenarios = &nScenarios. + &qScenario0.;
      scenarioBM = &scenarioBM.;
      nVars = 2;
      sMAPE = J(lead,nScenarios*nVars,0);
      rsMAPE = J(lead,nScenarios*nVars,0);
      sizeCI = J(lead,nScenarios*nVars,0);
      rLenCI = J(lead,nScenarios*nVars,0);
      do iSim = 1 to nSim;
         do h = 1 to lead;
            do iScenario = 1 to nScenarios;
```

```
         do iVar = 1 to nVars;
            yF=d[(iSim-1)*lead+h,
                 nVars+((iScenario-1)*nVars+(iVar-1))*3+1];
            yFlb=d[(iSim-1)*lead+h,
                   nVars+((iScenario-1)*nVars+(iVar-1))*3+2];
            yFub=d[(iSim-1)*lead+h,
                   nVars+((iScenario-1)*nVars+(iVar-1))*3+3];
            yFlbBM=d[(iSim-1)*lead+h,
                     nVars+((scenarioBM-1)*nVars+(iVar-1))*3+2];
            yFubBM=d[(iSim-1)*lead+h,
                     nVars+((scenarioBM-1)*nVars+(iVar-1))*3+3];
            y =d[(iSim-1)*lead+h,iVar];
            * symmetric Mean Absolute Percentage Error (sMAPE);
            if(abs(yF)+abs(y)>0) then do;
               sMAPE[h,(iVar-1)*nScenarios+iScenario] =
                  sMAPE[h,(iVar-1)*nScenarios+iScenario]
                     + 2*abs(yF-y)/(abs(yF)+abs(y))/nSim;
            end;
            * size;
            if(y>=yFlb & y<=yFub) then do;
               sizeCI[h,(iVar-1)*nScenarios+iScenario] =
                  sizeCI[h,(iVar-1)*nScenarios+iScenario] + 1/nSim;
            end;
            * relative length;
            if(yFubBM-yFlbBM>0) then do;
               rLenCI[h,(iVar-1)*nScenarios+iScenario] =
                  rLenCI[h,(iVar-1)*nScenarios+iScenario]
                     + (yFub-yFlb)/(yFubBM-yFlbBM)/nSim;
            end;
         end;
      end;
   end;
end;
do h = 2 to lead;
   do iScenario = 1 to nScenarios;
      do iVar = 1 to nVars;
         sMAPE[h,(iVar-1)*nScenarios+iScenario] =
            (sMAPE[h,(iVar-1)*nScenarios+iScenario]
               + sMAPE[h-1,(iVar-1)*nScenarios+iScenario]*(h-1))/h;
      end;
   end;
end;
do h = 1 to lead;
   do iScenario = 1 to nScenarios;
      do iVar = 1 to nVars;
         * relative symmetric Mean Absolute Percentage Error;
         rsMAPE[h,(iVar-1)*nScenarios+iScenario] =
            sMAPE[h,(iVar-1)*nScenarios+iScenario]
               / sMAPE[h,(iVar-1)*nScenarios+scenarioBM];
      end;
   end;
end;

* rearrange results;
```

```
      n = ncol(sMAPE)/nVars;
      evalResult = sMAPE[,1:n];
      do iVar = 2 to nVars;
         evalResult = evalResult // sMAPE[,(iVar-1)*n+1:iVar*n];
      end;
      do iVar = 1 to nVars;
         evalResult = evalResult // rsMAPE[,(iVar-1)*n+1:iVar*n];
      end;
      do iVar = 1 to nVars;
         evalResult = evalResult // sizeCI[,(iVar-1)*n+1:iVar*n];
      end;
      do iVar = 1 to nVars;
         evalResult = evalResult // rLenCI[,(iVar-1)*n+1:iVar*n];
      end;
      evalResult = ((1:4)`@J(lead*nVars,1,1))
                || (J(4,1,1)@((1:nVars)`@J(lead,1,1)))
                || (J(4*nVars,1,1)@(1:lead)`)
                || evalResult;
      create &tblEval. from evalResult;
      append from evalResult;
      close;
   quit;
%mend;
```

The following macro incorporates all the previous macros to test the forecasts from different methods (equation-based versus simulation-based) for different scenarios (unconditional versus four types of hard conditions). All point and interval forecasts are saved in the data set that is specified by the tblAll argument. All evaluation results are saved in the data set that is specified by the tblEval argument.

```
%macro hcTest(nSim,T,lead,alpha,nmc,nScenarios,qScenario0,scenarioBM,
   tblAll,tblEval);
   * nSim: number of DGPs;
   * T: in-sample sample size;
   * lead: future horizons;
   * alpha: size of the confidence interval or credible interval;
   * nmc: number of Monte Carlo iterations;
   * nScenarios: number of scenarios;
   * qScenario0: whether there is scenario 0 for equation-based forecasts,
                 1 for yes and 0 for no;
   * scenarioBM: the index of the benchmark scenario;
   * tblAll: output table name of point and interval forecasts for all DGPs;
   * tblEval: output table name for evaluation results;
   %do iSim = 1 %to &nSim.;
      %cfSimulateData(&iSim.,&T.,&lead.,t1,t2);
      %hcConstructScenarios(&T.,t1,t3);
      %hcEstimateAndForecast(t2,t3,&alpha.,&lead.,&nmc.,of,ocf,ocfsim);
      %hcSaveForecasts(&iSim.,&T.,&lead.,&nScenarios.,t1,of,ocf,&tblAll.);
   %end;
   %cfEvaluate(&tblAll.,&lead.,&nSim.,
               &nScenarios.,&qScenario0.,&scenarioBM.,
               &tblEval.);
%mend;
```

The following macro variables and macro set up and perform the test:

```
%let nSim = 1000;
%let T = 200;
%let lead = 4;
%let alpha = 0.05;
%let nmc = 10000;
%let nScenarios = 5;
%let qScenario0 = 1;
%let scenarioBM = 6;

%hcTest(&nSim.,&T.,&lead.,&alpha.,&nmc.,
        &nScenarios.,&qScenario0.,&scenarioBM.,
        hcForecasts,hcEval);
```

In order to show the result in a good style, the following template is created and the macro applies the template to the data set:

```
proc template;
   define table hcEvalTemplate;
      column col3 col4 col5 col6 col7 col8 col9;
      define header hc;
         text "Conditional Forecasts, Hard Conditions";
         start=col5; end=col8;
      end;
      define column col3;
         header="Horizon";
      end;
      define column col4;
         header="Equation Based"; format=7.5;
      end;
      define column col5;
         header="Scenario 1"; format=12.5;
      end;
      define column col6;
         header="Scenario 2"; format=12.5;
      end;
      define column col7;
         header="Scenario 3"; format=12.5;
      end;
      define column col8;
         header="Scenario 4"; format=12.5;
      end;
      define column col9;
         header="Unconditional"; format=12.5;
      end;
   end;
run;
%macro cfPrint(template,data);
   data _NULL_;
      set &data.;
      file print ods=(template="&template.");
      put _ODS_;
   run;
%mend;
```

The following macro calls print the sMAPEs for y1 and y2. Output 42.5.1 and Output 42.5.2 show that as more future information from scenario 1 to 4 is used in the conditional forecasts, the sMAPEs get smaller, which means that the accuracy of forecasts gets better.

```
%cfPrint(hcEvalTemplate, hcEval(where=(col1=1 and col2=1)));

%cfPrint(hcEvalTemplate, hcEval(where=(col1=1 and col2=2)));
```

Output 42.5.1 The sMAPEs for y1

Conditional Forecasts and Scenario Analysis

Conditional Forecasts, Hard Conditions						
Horizon	Equation Based	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Unconditional
1	0.90307	0.90101	0.89794	0.89874	0.89419	0.90146
2	0.88474	0.88055	0.87737	0.87471	0.87234	0.88451
3	0.87674	0.87243	0.87309	0.86822	0.86664	0.87632
4	0.86908	0.86553	0.86553	0.86266	0.86129	0.86954

Output 42.5.2 The sMAPEs for y2

Conditional Forecasts and Scenario Analysis

Conditional Forecasts, Hard Conditions						
Horizon	Equation Based	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Unconditional
1	1.07895	1.06867	1.07027	1.07053	1.06977	1.07941
2	1.16152	1.14704	1.15181	1.14594	1.14441	1.16143
3	1.22260	1.21487	1.21397	1.20050	1.19548	1.22251
4	1.27106	1.26842	1.26392	1.25277	1.24037	1.27112

The following macro calls print the relative sMAPEs for y1 and y2. Output 42.5.3 and Output 42.5.4 show that the relative sMAPE for all conditional point forecasts is less than 1, which means that all of them have better accuracy than the benchmark unconditional forecasts. The relative sMAPE for equation-based point forecasts is very close to 1, which means that the accuracy of the equation-based forecasts is similar to that of unconditional forecasts.

```
%cfPrint(hcEvalTemplate, hcEval(where=(col1=2 and col2=1)));

%cfPrint(hcEvalTemplate, hcEval(where=(col1=2 and col2=2)));
```

Output 42.5.3 The Relative sMAPEs for y1

Conditional Forecasts and Scenario Analysis

Conditional Forecasts, Hard Conditions						
Horizon	Equation Based	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Unconditional
1	1.00179	0.99951	0.99610	0.99698	0.99194	1.00000
2	1.00026	0.99552	0.99193	0.98892	0.98624	1.00000
3	1.00048	0.99556	0.99632	0.99076	0.98896	1.00000
4	0.99948	0.99540	0.99539	0.99209	0.99052	1.00000

Output 42.5.4 The Relative sMAPEs for y2

Conditional Forecasts and Scenario Analysis

Conditional Forecasts, Hard Conditions						
Horizon	Equation Based	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Unconditional
1	0.99958	0.99005	0.99153	0.99177	0.99107	1.00000
2	1.00008	0.98760	0.99172	0.98666	0.98534	1.00000
3	1.00008	0.99375	0.99302	0.98200	0.97789	1.00000
4	0.99995	0.99788	0.99434	0.98556	0.97581	1.00000

The following macro calls print the sizes for y1 and y2. Output 42.5.5 and Output 42.5.6 show that the sizes for all forecasts are very close to 0.95, which means all forecasts have the correct sizes.

```
%cfPrint(hcEvalTemplate, hcEval(where=(col1=3 and col2=1)));

%cfPrint(hcEvalTemplate, hcEval(where=(col1=3 and col2=2)));
```

Output 42.5.5 The Sizes for y1

Conditional Forecasts and Scenario Analysis

Conditional Forecasts, Hard Conditions						
Horizon	Equation Based	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Unconditional
1	0.94700	0.94300	0.93600	0.94500	0.94800	0.94400
2	0.94900	0.94700	0.94100	0.94600	0.94700	0.95000
3	0.95100	0.94600	0.94300	0.95000	0.95300	0.95200
4	0.94400	0.94700	0.94500	0.94800	0.94200	0.93800

Output 42.5.6 The Sizes for y2

Conditional Forecasts and Scenario Analysis

Conditional Forecasts, Hard Conditions

Horizon	Equation Based	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Unconditional
1	0.95400	0.95100	0.94700	0.94400	0.94700	0.94600
2	0.95000	0.94300	0.94100	0.94200	0.93200	0.94900
3	0.94900	0.94900	0.94600	0.94400	0.94300	0.94200
4	0.94100	0.94100	0.93700	0.94200	0.94400	0.94500

The following macro calls print the relative interval lengths for y1 and y2. Output 42.5.7 and Output 42.5.8 show that almost all conditional forecasts show smaller relative interval length than the benchmark unconditional forecasts show, which means that the conditional forecasts have better interval forecasts than unconditional forecasts. The relative interval lengths for equation-based forecasts are all greater than 1, which means that equation-based forecasts have worse interval forecasting ability than unconditional forecasts have. The main reason might be that the unconditional forecasts account for the uncertainty of parameters but equation-based forecasts do not.

```
%cfPrint(hcEvalTemplate, hcEval(where=(col1=4 and col2=1)));

%cfPrint(hcEvalTemplate, hcEval(where=(col1=4 and col2=2)));
```

Output 42.5.7 The Relative Interval Lengths for y1

Conditional Forecasts and Scenario Analysis

Conditional Forecasts, Hard Conditions

Horizon	Equation Based	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Unconditional
1	1.02290	0.99210	0.95616	0.95064	0.96477	1.00000
2	1.00814	1.00050	0.95801	0.94982	0.94094	1.00000
3	1.01348	0.98599	0.99613	0.97029	0.95520	1.00000
4	1.01754	1.00582	0.99147	1.00897	0.96589	1.00000

Output 42.5.8 The Relative Interval Lengths for y2

Conditional Forecasts and Scenario Analysis

Conditional Forecasts, Hard Conditions

Horizon	Equation Based	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Unconditional
1	1.02990	1.00010	0.99527	0.97468	0.98016	1.00000
2	1.02772	0.98791	0.96935	0.97000	0.95431	1.00000
3	1.03613	1.01415	0.97397	0.97472	0.95715	1.00000
4	1.01918	0.97980	0.96314	0.96461	0.95298	1.00000

Conditional Forecasts and Scenario Analysis under Soft Conditions

This section considers the soft conditions, under which some future dependent variables are bounded within certain ranges instead of fixed to some single values.

The following macro estimates the model and outputs the unconditional forecasts. In the case of soft conditions, only the unconditional forecasts are needed. When the SDATA= option in the CONDFORE statement is not specified, unconditional forecasts are generated. The number of Monte Carlo iterations needs to be large because later the simulated forecasts that satisfy the soft conditions are selected from the pool of all unconditional forecasts.

```
%macro scEstimateAndForecast(tblSample,alpha,lead,nmc,tblUcf,tblUcfSim);
   * tblSample: input table name for in-sample data;
   * alpha: size of the credible interval;
   * lead: future horizons;
   * nmc: number of Monte Carlo iterations;
   * tblUcf: output table name of unconditional point and interval
            forecasts;
   * tblUcfSim: output table name of simulated unconditional forecasts;
   proc varmax data=&tblSample.;
      model y1 - y3 / p=2 prior noprint;
      condfore alpha=&alpha. lead=&lead. out=&tblUcf. outsim=&tblUcfSim.
         parm=sampling nbi=1000 nmc=&nmc.;
   run;
%mend;
```

The scenarios for four types of soft conditions are constructed from the following macro. To set up the correct bounds, both the true DGP (data-generating process) and unconditional forecasts are used. In the real world, the true DGP is not available, and those bound values reflect the scenarios of interest.

```
%macro scConstructScenarios(T,lead,tblDGP,tblUcf);
   * T: in-sample sample size;
   * lead: future horizons;
   * tblDGP: input table name for full-sample data;
   * tblUcf: input table name of unconditional point and interval
            forecasts;
   * four scenarios are implicitly output:
        scenarios i, i=1 to 4: future y3 is known for
           lb_j<=y3_j<=ub_j, j=1 to i, where lb_j and ub_j are lower and
           upper bounds whose values are saved in the corresponding macro
           variables, and y3_j is the j-step-ahead furture value of y3;
   data _NULL_;
      set &tblDGP.(firstobs=%eval(&T.+1) obs=%eval(&T.+&lead.) keep=Y3);
      set &tblUcf.(keep=step Y3_MEDIAN Y3_LB Y3_UB);
      array q[&lead.] q1 - q&lead.;
      array lb[&lead.] lb1 - lb&lead.;
      array ub[&lead.] ub1 - ub&lead.;
      retain q lb ub;
      if(Y3<=Y3_LB) then do;
         q[step] = 1; ub[step] = Y3_LB;
      end;
      if (Y3>Y3_LB and Y3<=Y3_MEDIAN) then do;
         q[step] = 2; lb[step] = Y3_LB; ub[step] = Y3_MEDIAN;
      end;
```

```
      if (Y3>Y3_MEDIAN and Y3<=Y3_UB) then do;
         q[step] = 3; lb[step] = Y3_MEDIAN; ub[step] =Y3_UB;
      end;
      if (Y3>Y3_UB) then do;
         q[step] = 4; lb[step] = Y3_UB;
      end;
      if(_N_=&lead.) then do;
         %do i = 1 %to &lead.;
            call symputx("lb&i.",lb&i.,'G');
            call symputx("ub&i.",ub&i.,'G');
         %end;
      end;
   run;
%mend;
```

The simulated forecasts that satisfy each type of soft condition in each scenario are selected in the following macro:

```
%macro scClassifySimulatedForecasts(lead,tblUcfSim,tblSCSim);
   * lead: future horizons;
   * tblUcfSim: output table name of simulated unconditional forecasts;
   * tblSCSim: output table name of simulated conditional forecasts
               under soft conditioins specified in the scenarios;
   data &tblSCSim.;
      set &tblUcfSim.;
      array lb[&lead.] lb1 - lb&lead.;
      array ub[&lead.] ub1 - ub&lead.;
      array y3f[&lead.] y3_1 - y3_&lead.;
      %do j = 1 %to &lead.;
         lb[&j.]=&&lb&j.; ub[&j.]=&&ub&j.;
      %end;
      do myScenario=1 to 4;
         outputCond = 1;
         do i = 1 to myScenario;
            if(outputCond=1) then do;
               if(lb[i]=.) then do;
                  if(y3f[i]<=ub[i]) then outputCond=1;
                  else outputCond = 0;
               end;
               else do;
                  if(ub[i]=.) then do;
                     if(y3f[i]>lb[i]) then outputCond=1;
                     else outputCond = 0;
                  end;
                  else do;
                     if(y3f[i]>lb[i] and y3f[i]<=ub[i]) then outputCond=1;
                     else outputCond = 0;
                  end;
               end;
            end;
         end;
         if(outputCond=1) then output;
      end;
   run;
```

```
   proc sort data=&tblSCSim.; by myscenario; run;
%mend;
```

The point and interval forecasts for each scenario of soft conditions are generated from the following macro:

```
%macro scGetForecastStats(alpha,lead,tblSCSim,tblSCForecasts);
   * alpha: size of the credible interval;
   * lead: future horizons;
   * tblSCSim: input table name of simulated conditional forecasts
               under soft conditioins specified in the scenarios;
   * tblSCForecasts: output table name of conditional point and interval
               forecasts under soft conditioins specified in the scenarios;
   data _NULL_;
      lbPctl = &alpha./2*100;
      ubPctl = 100-lbPctl;
      call symputx("lbPctl",lbPctl,'G');
      call symputx("ubPctl",ubPctl,'G');
   run;
   proc univariate data=&tblSCSim. noprint;
      var y1_1 - y1_&lead. y2_1 - y2_&lead.;
      output out=oucfx pctlpts=&lbPctl. &ubPctl. 50
         pctlpre=%do j=1 %to &lead.; y1_&j. %end;
                 %do j=1 %to &lead.; y2_&j. %end;
         pctlname=_lb _ub _median;
      by myscenario;
   run;
   data &tblSCForecasts.;
      set oucfx;
      array y1f[&lead.]  %do j=1 %to &lead.; Y1_&j._median %end; ;
      array y1lb[&lead.] %do j=1 %to &lead.; Y1_&j._lb     %end; ;
      array y1ub[&lead.] %do j=1 %to &lead.; Y1_&j._ub     %end; ;
      array y2f[&lead.]  %do j=1 %to &lead.; Y2_&j._median %end; ;
      array y2lb[&lead.] %do j=1 %to &lead.; Y2_&j._lb     %end; ;
      array y2ub[&lead.] %do j=1 %to &lead.; Y2_&j._ub     %end; ;
      do i = 1 to &lead.;
         Y1_MEDIAN = y1f[i];
         Y1_LB = y1lb[i];
         Y1_UB = y1ub[i];
         Y2_MEDIAN = y2f[i];
         Y2_LB = y2lb[i];
         Y2_UB = y2ub[i];
         step = i;
         output;
      end;
      keep myscenario step y1_median y1_lb y1_ub y2_median y2_lb y2_ub;
   run;
%mend;
```

The following macro saves all types of forecasts for one simulated data set to one data set for evaluation:

```
%macro scSaveForecasts(dgpi,T,lead,tblDGP,tblUcf,tblSCForecasts,tblAll);
   * dgpi: index of DGP;
   * T: in-sample sample size;
   * lead: future horizons;
   * tblDGP: input table name for full-sample data;
```

```
   * tblUcf: input table name of unconditional point and interval
             forecasts;
   * tblSCForecasts: input table name of conditional point and interval
                     forecasts under soft conditioins specified in the scenarios;
   * tblAll: output table name of point and interval forecasts for all DGPs;
   data forecasts;
      set &tblDGP.(firstobs=%eval(&T.+1) obs=%eval(&T.+&lead.) keep=Y1 Y2);
      %do i = 1 %to 4;
         set &tblSCForecasts.( where=(myscenario_S&i.=&i.)
            rename=( Y1_MEDIAN=Y1_S&i. Y2_MEDIAN=Y2_S&i.
                     Y1_LB=Y1_LB_S&i. Y1_UB=Y1_UB_S&i.
                     Y2_LB=Y2_LB_S&i. Y2_UB=Y2_UB_S&i.
                     myscenario=myscenario_S&i.)
            keep=myscenario Y1_MEDIAN Y2_MEDIAN Y1_LB Y1_UB Y2_LB Y2_UB);
      %end;
      set &tblUcf.(
            rename=( Y1_MEDIAN=Y1_S5 Y2_MEDIAN=Y2_S5
                     Y1_LB=Y1_LB_S5 Y1_UB=Y1_UB_S5
                     Y2_LB=Y2_LB_S5 Y2_UB=Y2_UB_S5 )
            keep=Y1_MEDIAN Y2_MEDIAN Y1_LB Y1_UB Y2_LB Y2_UB);
      dgpIndex = &dgpi.;
      h = _N_;
      drop myscenario_S1 - myscenario_S4;
   run;

   proc append base=&tblAll. data=forecasts; run;
%mend;
```

The following macro incorporates all previous macros to test the forecasts for different scenarios (unconditional versus four types of soft conditions). All point and interval forecasts are saved in the data set that is specified in the tblAll argument. All evaluation results are saved in the data set that is specified in the tblEval argument.

```
%macro scTest(nSim,T,lead,alpha,nmc,nScenarios,qScenario0,scenarioBM,
   tblAll,tblEval);
   * nSim: number of DGPs;
   * T: in-sample sample size;
   * lead: future horizons;
   * alpha: size of the confidence interval or credible interval;
   * nmc: number of Monte Carlo iterations;
   * nScenarios: number of scenarios;
   * qScenario0: whether there is S0 for equation-based forecasts,
                 1 for yes and 0 for no;
   * scenarioBM: the index of the benchmark scenario;
   * tblAll: output table name of point and interval forecasts for all DGPs;
   * tblEval: output table name for evaluation results;
   %do iSim = 1 %to &nSim.;
      %cfSimulateData(&iSim.,&T.,&lead.,t1,t2);
      %scEstimateAndForecast(t2,&alpha.,&lead.,&nmc.,oucf,oucfsim);
      %scConstructScenarios(&T.,&lead.,t1,oucf);
      %scClassifySimulatedForecasts(&lead.,oucfsim,oucfsimx);
      %scGetForecastStats(&alpha.,&lead.,oucfsimx,oscf);
      %scSaveForecasts(&iSim.,&T.,&lead.,t1,oucf,oscf,&tblAll.);
   %end;
```

```
   %cfEvaluate(&tblAll.,&lead.,&nSim.,
               &nScenarios.,&qScenario0.,&scenarioBM.,
               &tblEval.);
%mend;
```

The following macro variables and macro set up and perform the test:

```
%let nSim = 1000;
%let T = 200;
%let lead = 4;
%let alpha = 0.50;
%let nmc = 100000;
%let nScenarios = 5;
%let qScenario0 = 0;
%let scenarioBM = 5;

%scTest(&nSim., &T.,&lead.,&alpha.,&nmc.,
        &nScenarios.,&qScenario0.,&scenarioBM.,
        scForecasts,scEval);
```

In order to show the result in a good style, the following template is created for the evaluation data set:

```
proc template;
   define table scEvalTemplate;
      column col3 col4 col5 col6 col7 col8;
      define header sc;
         text "Conditional Forecasts, Soft Conditions";
         start=col4; end=col7;
      end;
      define column col3;
         header="Horizon";
      end;
      define column col4;
         header="Scenario 1"; format=12.5;
      end;
      define column col5;
         header="Scenario 2"; format=12.5;
      end;
      define column col6;
         header="Scenario 3"; format=12.5;
      end;
      define column col7;
         header="Scenario 4"; format=12.5;
      end;
      define column col8;
         header="Unconditional"; format=12.5;
      end;
   end;
run;
```

The following macro calls print the sMAPEs for y1 and y2 in Output 42.5.9 and Output 42.5.10, respectively. An interesting fact is that as the horizon increases, the accuracy of forecasts for y1 in each scenario gets better (which is not common) and the accuracy of forecasts for y2 in each scenario gets worse (which is common). However, the forecasting methods can still be compared.

```
%cfPrint(scEvalTemplate, scEval(where=(col1=1 and col2=1)));

%cfPrint(scEvalTemplate, scEval(where=(col1=1 and col2=2)));
```

Output 42.5.9 The sMAPEs for y1

Conditional Forecasts and Scenario Analysis

Conditional Forecasts, Soft Conditions					
Horizon	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Unconditional
1	0.90733	0.89925	0.90273	0.89949	0.90144
2	0.88731	0.88271	0.88282	0.87862	0.88429
3	0.87891	0.87626	0.87505	0.86820	0.87613
4	0.87062	0.86864	0.86757	0.86206	0.86852

Output 42.5.10 The sMAPEs for y2

Conditional Forecasts and Scenario Analysis

Conditional Forecasts, Soft Conditions					
Horizon	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Unconditional
1	1.06627	1.06999	1.06861	1.06688	1.07902
2	1.14420	1.14813	1.14674	1.14376	1.16096
3	1.20960	1.21150	1.20106	1.19365	1.22223
4	1.26086	1.26050	1.25022	1.24181	1.27045

The following macro calls print the relative sMAPEs for y1 and y2 in Output 42.5.11 and Output 42.5.12, respectively. Most relative sMAPEs for conditional forecasts under soft conditions are less than 1, which means that those conditional forecasts have a better accuracy than the unconditional forecasts have. The forecasts in scenario 4, compared to other forecasts for each horizon, always have the smallest relative sMAPEs, which means that the conditional forecasts under soft conditions successfully take the advantage of the available future information.

```
%cfPrint(scEvalTemplate, scEval(where=(col1=2 and col2=1)));

%cfPrint(scEvalTemplate, scEval(where=(col1=2 and col2=2)));
```

Output 42.5.11 The Relative sMAPEs for y1

Conditional Forecasts and Scenario Analysis

Conditional Forecasts, Soft Conditions					
Horizon	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Unconditional
1	1.00654	0.99757	1.00144	0.99784	1.00000
2	1.00342	0.99821	0.99834	0.99359	1.00000
3	1.00317	1.00014	0.99877	0.99094	1.00000
4	1.00242	1.00014	0.99891	0.99257	1.00000

Output 42.5.12 The Relative sMAPEs for y2

Conditional Forecasts and Scenario Analysis

Conditional Forecasts, Soft Conditions					
Horizon	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Unconditional
1	0.98818	0.99163	0.99035	0.98875	1.00000
2	0.98556	0.98894	0.98775	0.98518	1.00000
3	0.98966	0.99122	0.98268	0.97662	1.00000
4	0.99245	0.99217	0.98408	0.97746	1.00000

The following macro calls print the sizes for y1 and y2 in Output 42.5.13 and Output 42.5.14, respectively. All sizes are around 0.5 (the nominal significance level), which means that all interval forecasts have the correct size.

```
%cfPrint(scEvalTemplate, scEval(where=(col1=3 and col2=1)));

%cfPrint(scEvalTemplate, scEval(where=(col1=3 and col2=2)));
```

Output 42.5.13 The Sizes for y1

Conditional Forecasts and Scenario Analysis

Conditional Forecasts, Soft Conditions					
Horizon	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Unconditional
1	0.50600	0.50900	0.51000	0.50600	0.49700
2	0.50000	0.50600	0.50300	0.51000	0.50500
3	0.50200	0.50300	0.49000	0.50100	0.50000
4	0.50800	0.50700	0.51300	0.51000	0.49900

Output 42.5.14 The Sizes for y2

Conditional Forecasts and Scenario Analysis

Conditional Forecasts, Soft Conditions					
Horizon	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Unconditional
1	0.48900	0.48600	0.48800	0.48800	0.49300
2	0.49500	0.46200	0.46700	0.45900	0.49300
3	0.50800	0.49300	0.49500	0.49100	0.50600
4	0.47800	0.46100	0.46400	0.45900	0.47200

The following macro calls print the relative interval lengths for y1 and y2 in Output 42.5.15 and Output 42.5.16, respectively. All conditional forecasts under soft conditions have a relative interval length of less than 1, which means that the conditional forecasts provide better interval forecasts than unconditional forecasts, given that all forecasts have the correct size. The smallest relative interval lengths for each horizon almost always lie in the columns of scenario 4, which indicates that more information results in better interval forecasts.

```
%cfPrint(scEvalTemplate, scEval(where=(col1=4 and col2=1)));

%cfPrint(scEvalTemplate, scEval(where=(col1=4 and col2=2)));
```

Output 42.5.15 The Relative Interval Lengths for y1

Conditional Forecasts and Scenario Analysis

Conditional Forecasts, Soft Conditions					
Horizon	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Unconditional
1	0.97668	0.96309	0.95805	0.95808	1.00000
2	0.99957	0.97788	0.96682	0.95940	1.00000
3	0.99829	0.99797	0.98040	0.96568	1.00000
4	0.99659	0.99479	0.99352	0.97636	1.00000

Output 42.5.16 The Relative Interval Lengths for y2

Conditional Forecasts and Scenario Analysis

Conditional Forecasts, Soft Conditions					
Horizon	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Unconditional
1	0.98772	0.98616	0.98671	0.98540	1.00000
2	0.99356	0.97818	0.97409	0.97303	1.00000
3	0.99495	0.98684	0.97211	0.96640	1.00000
4	0.99775	0.99329	0.98459	0.97166	1.00000

In summary, you can make the following conclusions from the preceding results:

- Compared to simulation-based unconditional forecasts, equation-based unconditional forecasts have similar accuracy for point forecasts but worse accuracy for interval forecasts, perhaps because the equation-based method does not consider the uncertainty of parameters.
- Compared to simulation-based unconditional forecasts, simulation-based conditional forecasts under hard or soft conditions have better accuracy for both point and interval forecasts. As more future information becomes available, the conditional forecasts can become more accurate.

Example 42.6: Numerous Examples

The following are examples of syntax for model fitting:

```
/* Data 'a' Generated Process */
proc iml;
   sig = {1.0  0.5, 0.5  1.25};
   phi = {1.2 -0.5, 0.6  0.3};
   call varmasim(y,phi) sigma = sig n = 100 seed = 46859;
   cn = {'y1' 'y2'};
   create a from y[colname=cn];
   append from y;
```

```
run;;

/* when the series has a linear trend */
proc varmax data=a;
   model y1 y2 / p=1 trend=linear;
run;

/* Fit subset of AR order 1 and 3 */
proc varmax data=a;
   model y1 y2 / p=(1,3);
run;

/* Check if the series is nonstationary */
proc varmax data=a;
   model y1 y2 / p=1 dftest print=(roots);
run;

/* Fit VAR(1) in differencing */
proc varmax data=a;
   model y1 y2 / p=1 print=(roots) dify=(1);
run;

/* Fit VAR(1) in seasonal differencing */
proc varmax data=a;
   model y1 y2 / p=1 dify=(4) lagmax=5;
run;

/* Fit VAR(1) in both regular and seasonal differencing */
proc varmax data=a;
   model y1 y2 / p=1 dify=(1,4) lagmax=5;
run;

/* Fit VAR(1) in different differencing */
proc varmax data=a;
   model y1 y2 / p=1 dif=(y1(1,4) y2(1)) lagmax=5;
run;

/* Options related to prediction */
proc varmax data=a;
   model y1 y2 / p=1 lagmax=3
                 print=(impulse covpe(5) decompose(5));
run;

/* Options related to tentative order selection */
proc varmax data=a;
   model y1 y2 / p=1 lagmax=5 minic
                 print=(parcoef pcancorr pcorr);
run;

/* Automatic selection of the AR order */
proc varmax data=a;
   model y1 y2 / minic=(type=aic p=5);
run;
```

```
/* Compare results of LS and Yule-Walker Estimators */
proc varmax data=a;
   model y1 y2 / p=1 print=(yw);
run;

/* BVAR(1) of the nonstationary series y1 and y2 */
proc varmax data=a;
   model y1 y2 / p=1
      prior=(lambda=1 theta=0.2 ivar);
run;

/* BVAR(1) of the nonstationary series y1 */
proc varmax data=a;
   model y1 y2 / p=1
      prior=(lambda=0.1 theta=0.15 ivar=(y1));
run;

/* Data 'b' Generated Process */
proc iml;
   sig = { 0.5  0.14 -0.08 -0.03,  0.14 0.71 0.16 0.1,
          -0.08 0.16  0.65  0.23, -0.03 0.1  0.23 0.16};
   sig = sig * 0.0001;
   phi = {1.2 -0.5 0.  0.1,  0.6 0.3 -0.2  0.5,
          0.4  0. -0.2 0.1, -1.0 0.2  0.7 -0.2};
   call varmasim(y,phi) sigma = sig n = 100 seed = 32567;
   cn = {'y1' 'y2' 'y3' 'y4'};
   create b from y[colname=cn];
   append from y;
quit;

/* Cointegration Rank Test using Trace statistics */
proc varmax data=b;
   model y1-y4 / p=2 lagmax=4 cointtest;
run;

/* Cointegration Rank Test using Max statistics */
proc varmax data=b;
   model y1-y4 / p=2 lagmax=4 cointtest=(johansen=(type=max));
run;

/* Common Trends Test using Filter(Differencing) statistics */
proc varmax data=b;
   model y1-y4 / p=2 lagmax=4 cointtest=(sw);
run;

/* Common Trends Test using Filter(Residual) statistics */
proc varmax data=b;
   model y1-y4 / p=2 lagmax=4 cointtest=(sw=(type=filtres lag=1));
run;

/* Common Trends Test using Kernel statistics */
proc varmax data=b;
   model y1-y4 / p=2 lagmax=4 cointtest=(sw=(type=kernel lag=1));
run;
```

```
/* Cointegration Rank Test for I(2) */
proc varmax data=b;
   model y1-y4 / p=2 lagmax=4 cointtest=(johansen=(iorder=2));
run;

/* Fit VECM(2) with rank=3 */
proc varmax data=b;
   model y1-y4 / p=2 lagmax=4 print=(roots iarr);
   cointeg rank=3 normalize=y1;
run;

/* Weak Exogenous Testing for each variable */
proc varmax data=b outstat=bbb;
   model y1-y4 / p=2 lagmax=4;
   cointeg rank=3 exogeneity normalize=y1;
run;

/* Hypotheses Testing for long-run and adjustment parameter */
proc varmax data=b outstat=bbb;
   model y1-y4 / p=2 lagmax=4;
   cointeg rank=3 normalize=y1
      h=(1 0 0, 0 1 0, -1 0 0, 0 0 1)
      j=(1 0 0, 0 1 0, 0 0 1, 0 0 0);
run;

/* ordinary regression model */
proc varmax data=grunfeld;
   model y1 y2 = x1-x3;
run;

/* Ordinary regression model with subset lagged terms */
proc varmax data=grunfeld;
   model y1 y2 = x1 / xlag=(1,3);
run;

/* VARX(1,1) with no current time Exogenous Variables */
proc varmax data=grunfeld;
   model y1 y2 = x1 / p=1 xlag=1 nocurrentx;
run;

/* VARX(1,1) with different Exogenous Variables */
proc varmax data=grunfeld;
   model y1 = x3, y2 = x1 x2 / p=1 xlag=1;
run;

/* VARX(1,2) in difference with current Exogenous Variables */
proc varmax data=grunfeld;
   model y1 y2 = x1 / p=1 xlag=2 difx=(1) dify=(1);
run;
```

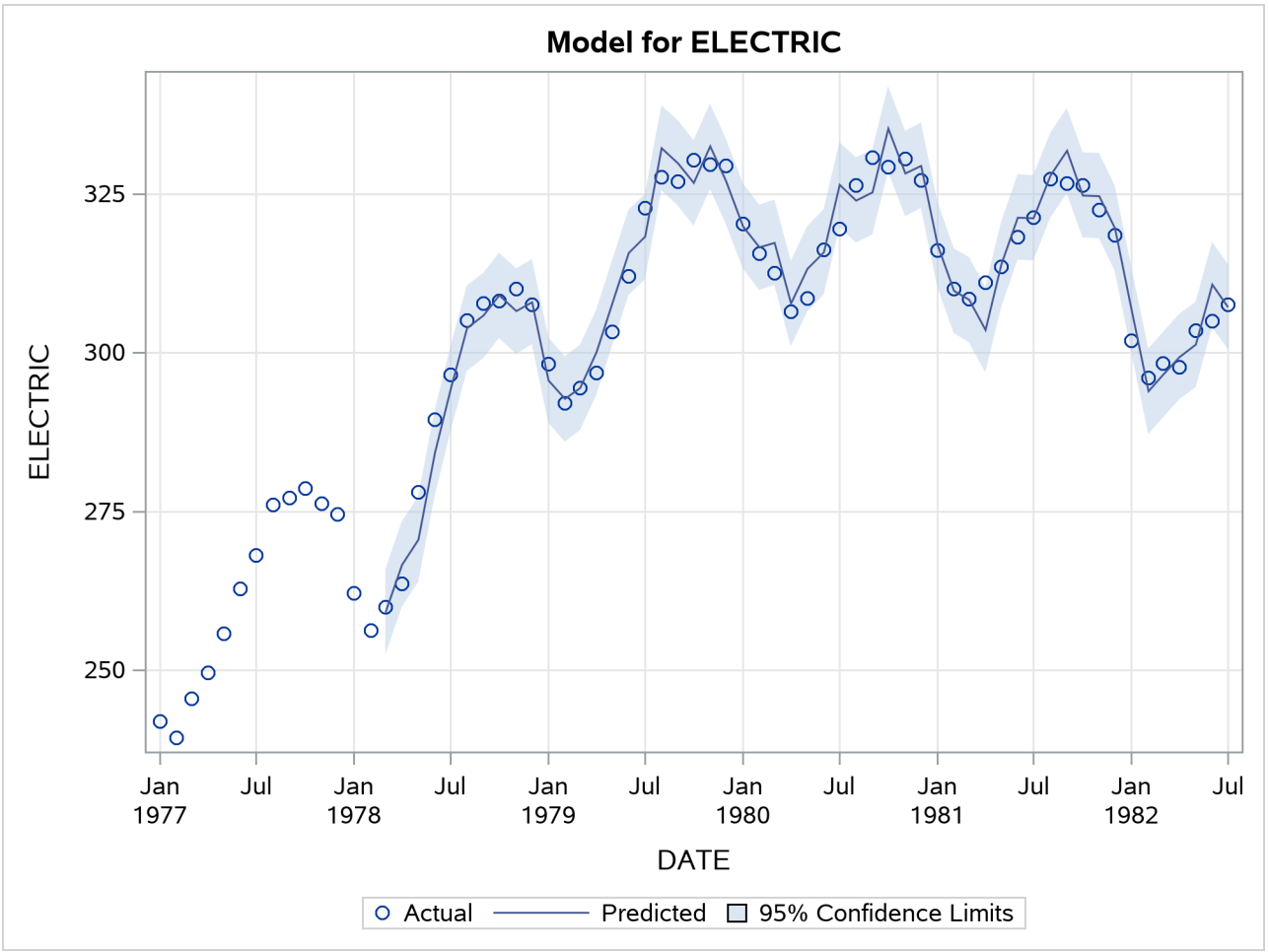
Example 42.7: Illustration of ODS Graphics

This example illustrates the use of ODS Graphics. For information about the graphics available in the VARMAX procedure, see the section "ODS Graphics" on page 3143.

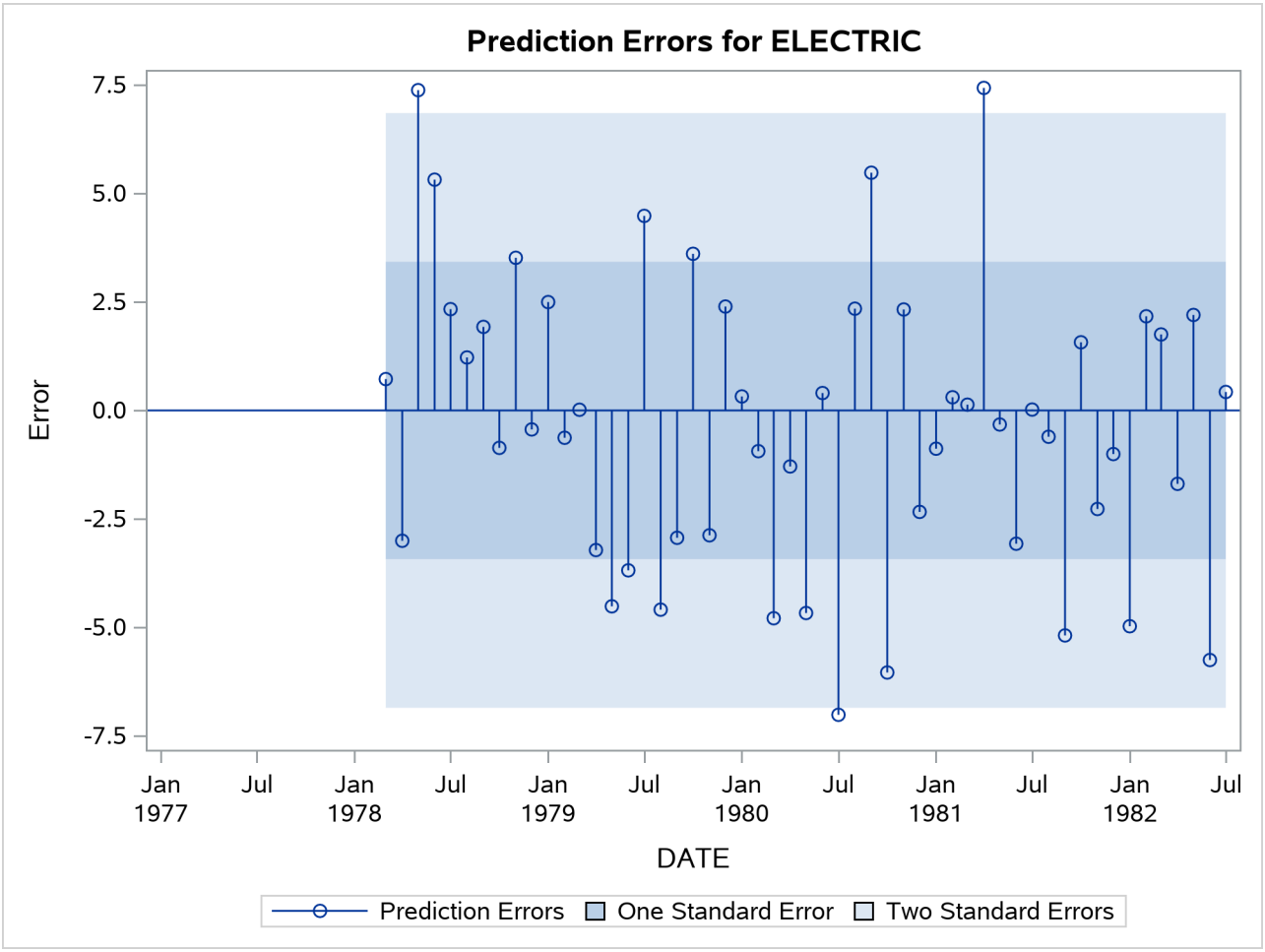
The following statements use the SASHELP.WORKERS data set to study the time series of electrical workers and its interaction with the series of masonry workers. The series and predict plots, the residual plot, and the forecast plot are created in Output 42.7.1 through Output 42.7.3. These are a selection of the plots created by the VARMAX procedure.

```
title "Illustration of ODS Graphics";
proc varmax data=sashelp.workers plot(unpack)=(residual model forecasts);
   id date interval=month;
   model electric masonry / dify=(1,12) noint p=1;
   output lead=12;
run;
```

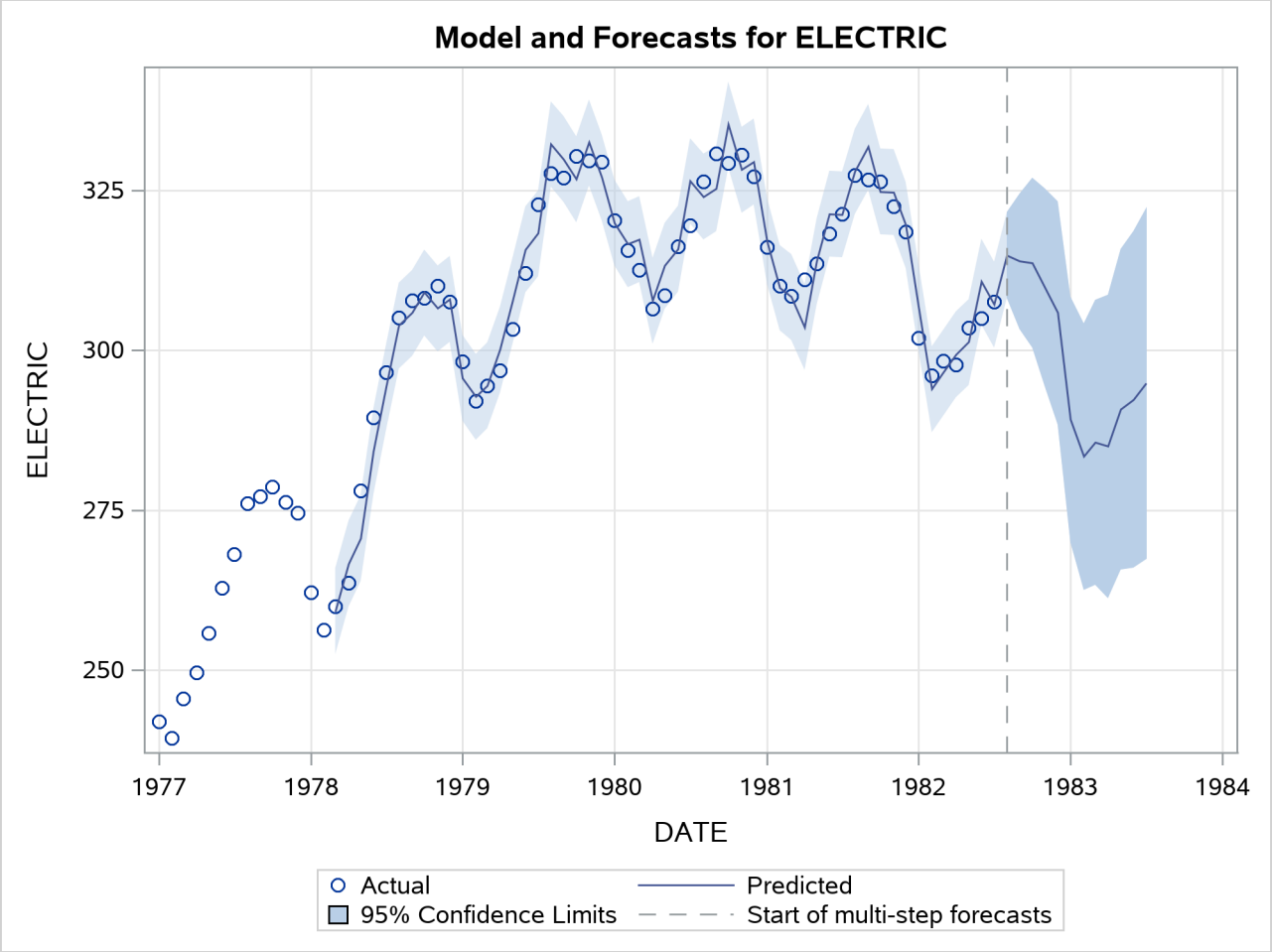
Output 42.7.1 Series and Predicted Series Plots



Output 42.7.2 Residual Plot



Output 42.7.3 Series and Forecast Plots



References

Anderson, T. W. (1951). "Estimating Linear Restrictions on Regression Coefficients for Multivariate Normal Distributions." *Annals of Mathematical Statistics* 22:327–351.

Ansley, C. F., and Newbold, P. (1979). "Multivariate Partial Autocorrelations." In *Proceedings of the Business and Economic Statistics Section*, 349–353. Washington, DC: American Statistical Association.

Bańbura, M., Giannone, D., and Lenza, M. (2015). "Conditional Forecasts and Scenario Analysis with Vector Autoregressions for Large Cross-Sections." *International Journal of Forecasting* 31:739–756.

Beran, J., Feng, Y., Ghosh, S., and Kulik, R. (2013). *Long-Memory Processes: Probabilistic Properties and Statistical Methods*. Heidelberg: Springer.

Bollerslev, T. (1990). "Modeling the Coherence in Short-Run Nominal Exchange Rates: A Multivariate Generalized ARCH Model." *Review of Economics and Statistics* 72:498–505.

Boswijk, H. P., and Doornik, J. A. (2004). "Identifying, Estimating and Testing Restricted Cointegrated Systems: An Overview." *Statistica Neerlandica* 58:440–465.

Chung, C.-F. (2001). "Calculating and Analyzing Impulse Responses for the Vector ARFIMA Model." *Economics Letters* 71:17–25.

Clark, T. E., and McCracken, M. W. (2017). "Tests of Predictive Ability for Vector Autoregressions Used for Conditional Forecasting." *Journal of Applied Econometrics* 32:533–553.

Ding, Z., Granger, C. W. J., and Engle, R. F. (1993). "A Long Memory Property of Stock Market Returns and a New Model." *Journal of Empirical Finance* 1:83–106.

Engle, R. F. (2002). "Dynamic Conditional Correlation: A Simple Class of Multivariate Generalized Autoregressive Conditional Heteroskedasticity Models." *Journal of Business and Economic Statistics* 20:339–350.

Engle, R. F., and Granger, C. W. J. (1987). "Co-integration and Error Correction: Representation, Estimation, and Testing." *Econometrica* 55:251–276.

Engle, R. F., and Kroner, K. F. (1995). "Multivariate Simultaneous Generalized ARCH." *Econometric Theory* 11:122–150.

Engle, R. F., and Ng, V. K. (1993). "Measuring and Testing the Impact of News on Volatility." *Journal of Finance* 48:1749–1778.

Geweke, J., and Porter-Hudak, S. (1983). "The Estimation and Application of Long Memory Time Series Models." *Journal of Time Series Analysis* 4:221–238.

Glosten, L., Jaganathan, R., and Runkle, D. (1993). "Relationship between the Expected Value and Volatility of the Nominal Excess Returns on Stocks." *Journal of Finance* 48:1779–1802.

Golub, G. H., and Van Loan, C. F. (1983). *Matrix Computations*. Baltimore: Johns Hopkins University Press.

Goodnight, J. H. (1979). "A Tutorial on the Sweep Operator." *American Statistician* 33:149–158.

Hosking, J. R. M. (1980). "The Multivariate Portmanteau Statistic." *Journal of the American Statistical Association* 75:602–608.

Johansen, S. (1988). "Statistical Analysis of Cointegration Vectors." *Journal of Economic Dynamics and Control* 12:231–254.

Johansen, S. (1995a). *Likelihood-Based Inference in Cointegrated Vector Autoregressive Models*. New York: Oxford University Press.

Johansen, S. (1995b). "A Statistical Analysis of Cointegration for I(2) Variables." *Econometric Theory* 11:25–59.

Johansen, S., and Juselius, K. (1990). "Maximum Likelihood Estimation and Inference on Cointegration: With Applications to the Demand for Money." *Oxford Bulletin of Economics and Statistics* 52:169–210.

Karlsson, S. (2013). "Forecasting with Bayesian Vector Autoregression." In *Handbook of Economic Forecasting,* vol. 2B, edited by G. Elliott and T. Timmermann, 791–897. Elsevier.

Kechagias, S., and Pipiras, V. (2015). "Definitions and Representations of Multivariate Long-Range Dependent Time Series." *Journal of Time Series Analysis* 36:1–25.

Koreisha, S., and Pukkila, T. (1989). "Fast Linear Estimation Methods for Vector Autoregressive Moving Average Models." *Journal of Time Series Analysis* 10:325–339.

Litterman, R. B. (1986). "Forecasting with Bayesian Vector Autoregressions: Five Years of Experience." *Journal of Business and Economic Statistics* 4:25–38.

Lütkepohl, H. (1993). *Introduction to Multiple Time Series Analysis*. 2nd ed. Berlin: Springer-Verlag.

Lütkepohl, H. (2007). *New Introduction to Multiple Time Series Analysis*. Berlin: Springer.

Nelson, D. B. (1991). "Conditional Heteroskedasticity in Asset Returns: A New Approach." *Econometrica* 59:347–370.

Nelson, D. B., and Cao, C. Q. (1992). "Inequality Constraints in the Univariate GARCH Model." *Journal of Business and Economic Statistics* 10:229–235.

Osterwald-Lenum, M. (1992). "A Note with Quantiles of the Asymptotic Distribution of the Maximum Likelihood Cointegration Rank Test Statistics." *Oxford Bulletin of Economics and Statistics* 54:461–472.

Pringle, R. M., and Rayner, A. A. (1971). *Generalized Inverse Matrices with Applications to Statistics*. New York: Hafner Publishing.

Quinn, B. G. (1980). "Order Determination for a Multivariate Autoregression." *Journal of the Royal Statistical Society, Series B* 42:182–185.

Reinsel, G. C. (1997). *Elements of Multivariate Time Series Analysis*. 2nd ed. New York: Springer-Verlag.

Spliid, H. (1983). "A Fast Estimation for the Vector Autoregressive Moving Average Model with Exogenous Variables." *Journal of the American Statistical Association* 78:843–849.

Stock, J. H., and Watson, M. W. (1988). "Testing for Common Trends." *Journal of the American Statistical Association* 83:1097–1107.

Tsay, W.-J. (2010). "Maximum Likelihood Estimation of Stationary Multivariate ARFIMA Processes." *Journal of Statistical Computation and Simulation* 80:729–745.

Waggoner, D. F., and Zha, T. (1999). "Conditional Forecasts in Dynamic Multivariate Models." *Review of Economics and Statistics* 81:639–651.

Zakoian, J. M. (1994). "Threshold Heteroscedastic Models." *Journal of Economic Dynamics and Control* 18:931–955.

Subject Index

Syntax Index