SAS/ETS® Software: Changes and Enhancements, Release 8.1

The Power to Know™

SAS Publishing

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/ETS® Software: Changes and Enhancements, Release 8.1*, Cary, NC: SAS Institute Inc., 2000

SAS/ETS® Software: Changes and Enhancements, Release 8.1

ISBN 1-58025-657-0

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, May 2000

Table of Contents

Part 1
General Information

Table of Contents

Chapter 1
New Financial Functions

Chapter Table of Contents

Chapter 1

New Financial Functions

TIMEVALUE Function

Overview

The TIMEVALUE function returns the equivalent of a reference amount at a base date using variable interest rates. TIMEVALUE computes the time-value equivalent of a date-amount pair at a specified date.

Syntax

TIMEVALUE(*BaseDate, ReferenceDate, ReferenceAmount, Interval, Date-1, Rate-1* **[** *, Date-2, Rate-2, ...* **]** *)*

Description

BaseDate	is a SAS date. The returned value is the time value of *ReferenceAmount* at *BaseDate*.
ReferenceDate	is a SAS date. This is the date of *ReferenceAmount*.
ReferenceAmount	is a numeric. This is the amount at *ReferenceDate*.
Interval	is a SAS interval. This is the compounding interval.
Date-i	is a SAS date. Each date is paired with a rate. The date *Date-i* is the time that *Rate-i* takes effect.
Rate-i	is a numeric percentage. Each rate is paired with a date. The rate *Rate-i* is the interest rate that starts on *Date-i*.

Details

- The list of date-rate pairs does not need to be sorted by date.
- When more than one date-rate pair is entered, SAS sorts dates chronologically and ignores duplicates (keeping the last occurrence of the date).
- Simple interest is applied for partial periods.
- There must be a valid date-rate pair whose date is at or prior to both the *ReferenceDate* and *BaseDate*.

Examples

You can express the accumulated value of an investment of $1,000 at a nominal interest rate of 10% compounded monthly for one year as

```
amount_base1 = TIMEVALUE ("01jan2000"d, "01jan2001"d, 1000,
                          "MONTH", "01jan2000"d, 10);
```

If the interest rate jumps to 20% halfway through the year, the resulting calculation would be

```
amount_base2 = TIMEVALUE ("01jan2000"d, "01jan2001"d, 1000,
                          "MONTH",
                          "01jan2000"d, 10, "01jul2000"d, 20);
```

Recall that the date-rate pairs do not need to be sorted by date. This flexibility allows amount_base2 and amount_base3 to assume the same value.

```
amount_base3 = TIMEVALUE ("01jan2000"d, "01jan2001"d, 1000,
                          "MONTH",
                          "01jul2000"d, 20, "01jan2000"d, 10);
```

SAVINGS Function

Overview

The SAVINGS function returns the balance of a uniform, periodic savings using variable interest rates. SAVINGS computes the accumulated balance of a savings account. It assumes uniform deposits and uniform compounding of interest.

Syntax

SAVINGS(*BaseDate, DepositDate, DepositAmount, DepositNumber, DepositInterval, Interval, Date-1, Rate-1* **[** *, Date-2, Rate-2, ...* **]** *)*

Description

BaseDate is a SAS date. The returned value is the balance of the savings at *BaseDate*.

DepositDate is a SAS date. This is the date of the first deposit.

DepositAmount is a numeric. All deposits are assumed constant. This is the value of each deposit.

DepositNumber is a positive integer. This is the number of deposits.

DepositInterval is a SAS interval. This is the frequency at which you make deposits.

Interval is a SAS interval. This is the compounding interval.

Date-i is a SAS date. Each date is paired with a rate. The date *Date-i* is the time that *Rate-i* takes effect.

Rate-i is a numeric percentage. Each rate is paired with a date. The rate *Rate-i* is the interest rate that starts on *Date-i*.

Details

- The list of date-rate pairs does not need to be given in chronological order.
- When more than one date-rate pair is entered, SAS sorts dates chronologically and ignores duplicates (keeping the last occurrence of the date).
- Simple interest is applied for partial periods.
- There must be a valid date-rate pair whose date is at or prior to both the *DepositDate* and *BaseDate*.

Examples

Suppose you deposit $300 monthly for two years into an account that compounds quarterly at an annual rate of 4%. The balance of the account after five years can be expressed as

```
amount_base1 = SAVINGS ("01jan2005"d, "01jan2000"d, 300, 24,
                        "MONTH", "QUARTER", "01jan2000"d, 4.00);
```

Suppose the interest rate increases by a quarter-point each year. Then the balance of the account could be expressed as

```
amount_base2 = SAVINGS ("01jan2005"d, "01jan2000"d, 300, 24,
                        "MONTH", "QUARTER", "01jan2000"d, 4.00,
                        "01jan2001"d, 4.25, "01jan2002"d, 4.50,
                        "01jan2003"d, 4.75, "01jan2004"d, 5.00);
```

If you want to know the balance after one year of deposits, the following statement sets amount_base3 to the desired balance.

```
amount_base3 = SAVINGS ("01jan2001"d, "01jan2000"d, 300, 24,
                        "MONTH", "QUARTER", "01jan2000"d, 4);
```

Recall that SAS ignores deposits after the base date, so the deposits after the *ReferenceDate* do not affect the returned value.

EFFRATE Function

Overview

The EFFRATE function returns the effective annual interest rate. EFFRATE computes the effective annual interest rate corresponding to a nominal annual interest rate.

Syntax

EFFRATE(*Interval, Rate* **)**

Description

Interval is a SAS interval. This is how often *Rate* compounds.

Rate is a numeric. This is a nominal annual interest rate (expressed as a percentage) that is compounded each *Interval*.

Details

Consider a nominal interest *Rate* and a compounding *Interval*. Suppose *m Interval*s occur in a year. The value returned by EFFRATE(*Interval, Rate*) equals

$$\left(1 + \frac{Rate}{m}\right)^{m} - 1$$

Examples

Suppose a nominal rate is 10%. The corresponding effective rate when interest is compounded monthly can be expressed as

```
effective_rate1 = EFFRATE ("MONTH", 10);
```

Again, suppose a nominal rate is 10%. The corresponding effective rate when interest is compounded quarterly can be expressed as

```
effective_rate2 = EFFRATE ("QUARTER", 10);
```

NOMRATE Function

Overview

The NOMRATE function returns the nominal annual interest rate. NOMRATE computes the nominal annual interest rate corresponding to an effective annual interest rate.

Syntax

NOMRATE(*Interval, Rate* **)**

Description

Interval is a SAS interval. This is how often the returned value is compounded.

Rate is a numeric. This is an effective annual interest rate (expressed as a percentage) that is compounded each *Interval*.

Details

Consider an effective interest *Rate* and a compounding *Interval*. Suppose *m Interval*s occur in a year. The value returned by NOMRATE(*Interval*, *Rate*) equals

$$m\left((1+Rate)^{\frac{1}{m}}-1\right)$$

Examples

Suppose an effective rate is 10% when compounding monthly. The corresponding nominal rate can be expressed as

```
effective_rate1 = NOMRATE ("MONTH", 10);
```

Suppose an effective rate is 10% when compounding quarterly. The corresponding nominal rate can be expressed as

```
effective_rate2 = NOMRATE ("QUARTER", 10);
```

Chapter 2
The SASECRSP Interface Engine

Chapter Table of Contents

Chapter 2
The SASECRSP Interface Engine

Overview

The SASECRSP interface engine enables SAS users to access and process time series data residing in CRSPAccess data files and provides a seamless interface between CRSP and SAS data processing.

The SASECRSP engine uses the LIBNAME statement to enable you to specify which time series you would like to read from the CRSPAccess data files, and how you would like to perform selection on the CRSP set you choose to access. You choose the daily CRSP data by specifying SETID=10, or the monthly CRSP data by specifying SETID=20. You can select which securities you wish to access by specifying PERMNO=*number*, and specify your range of dates for the selected time series by using RANGE=‘*begdt-enddt*’, or specify an input SAS data set named *setname* as input for issues with the INSET=‘*setname*’ option. The SAS Data step can then be used to perform further subsetting and to store the resulting time series into a SAS data set. You can perform more analysis if desired either in the same SAS session or in another session at a later time.

Getting Started

Structure of a SAS Data Set Containing Time Series Data

SAS requires time series data to be in a specific form that is recognizable by the SAS System. This form is a two-dimensional array, called a SAS data set, whose columns correspond to series variables and whose rows correspond to measurements of these variables at certain time periods. The time periods at which observations are recorded can be included in the data set as a time ID variable. The SASECRSP engine provides a time ID variable called DATE.

Reading CRSP Data Files

The SASECRSP engine supports reading time series, events, and portfolios from CRSPAccess data files. Only the series specified by the CRSP *setid* is written to the SAS data set.

Using the SAS DATA Step

If desired, you can store the selected series in a SAS data set by using the SAS DATA step. You can also perform other operations on your data inside the DATA step. Once the data is stored in a SAS data set you can use it as you would any other SAS data set.

Using SAS Procedures

You can print the output SAS data set by using the PRINT procedure and report information concerning the contents of your data set by using the CONTENTS procedure. You can create a view of the CRSPAccess data base by using the SQL procedure to create your view using the SASECRSP engine in your *libref*, along with the USING clause.

Syntax

The SASECRSP engine uses standard engine syntax. Options used by SASECRSP are summarized in the table below.

Description	Statement	Option
specify a CRSPAccess set id# where 10 designates daily and 20 monthly, which limits the frequency selection of time series that are included in the SAS data set. This is a required option.	LIBNAME *libref* SASECRSP	SETID=
specify the PERMNO of a security to be kept in the SAS data set.	LIBNAME *libref* SASECRSP	PERMNO=
specify the range of CRSP data to access. *begdt* and *enddt* specify the range of CRSPAccess data in YYYYMMDD format.	LIBNAME *libref* SASECRSP	RANGE=
uses a SAS data set named *setname* as input for issues.	LIBNAME *libref* SASECRSP	INSET=

The LIBNAME *libref* SASECRSP Statement

LIBNAME *libref* **SASECRSP** **'***physical name***'** *options;*

The following options can be used in the LIBNAME *libref* SASECRSP statement:

SETID=*crsp_setidnumber*

specifies the CRSP set ID number where 10 is the daily data set ID and 20 is the monthly data set ID number. Two possible values for *crsp_setidnumber* are 10 or 20. SETID is a required option for the SASECRSP engine. For more details about the CRSPAccess *crsp_setidnumber*, refer to "Using Multiple Products" in the *CRSPAccess Database Format Installation Guide*. For a more complete discussion of SASECRSP set names, key fields, and date fields, see Table 2.1 on page 16. As an example, to access monthly data, you would use the following statements:

```
LIBNAME myLib sasecrsp 'physical-name'
   SETID=20;
```

PERMNO=*crsp_permnumber*

By default, the SASECRSP engine reads all PERMNOs in the CRSPAccess database that you name in your SASECRSP *libref*. You can limit the time series read from the CRSP database by specifying the PERMNO= option on your LIBNAME statement. From a performance standpoint, the PERMNO= option does random access and reads only the data for the PERMNOs listed. There is no limit to the number of *crsp_permnumber* options that you can use. As an example, to access monthly data for Microsoft Corporation and for International Business Machine Corporation,

```
LIBNAME myLib sasecrsp 'physical-name'
   SETID=20,
   PERMNO=10107,
   PERMNO=12490;
```

RANGE=*'crsp_begdt-crsp_enddt'*

To limit the time range of data read from the CRSPAccess database, specify the RANGE= option in your SASECRSP *libref*, where *crsp_begdt* is the beginning date in YYYYMMDD format and *crsp_enddt* is the endding date of the range in YYYYMMDD format. From a performance standpoint, the engine reads all the data for a company and then restricts the data range before passing it back.

As an example, to access daily data for Microsoft Corporation and for International Business Machine Corporation for the first quarter of 2000, use the following statements:

```
LIBNAME myLib sasecrsp 'physical-name'
   SETID=20,
   PERMNO=10107,
   PERMNO=12490,
   RANGE='19990101-19990331';
```

INSET=*'setname[,keyfieldname][,keyfieldtype][,date1field][,date2field]'*

When you specify a SAS data set named *setname* as input for issues, the SASECRSP engine assumes a default PERMNO field containing selected CRSP PERMNOs is present in the data set. If optional parameters are included, the first one is the key name, and the second is the key type. The third optional parameter is the beginning date or event date, and the fourth is the ending date in YYYYMMDD format.

Details

The SAS Output Data Set

You can use the SAS DATA step to write the selected CRSP data to a SAS data set. This enables the user to easily analyze the data using SAS. The name of the output data set is specified by you on the DATA statement. This causes the engine supervisor to create a SAS data set using the specified name in either the SAS WORK library, or if specified, the USER library. For more about naming your SAS data set, see the section “Characteristics of SAS Data Libraries” in *SAS Language Reference: Dictionary*.

The contents of the SAS data set include the DATE of each observation, the series names of each series read from the CRSPAccess database, event variables, and the label or description of each series/event.

Available Data Sets

Table 2.1 shows the available data sets provided by the SASECRSP interface. Missing values are represented as ‘.’ in the SAS data set. You can use PROC PRINT and PROC CONTENTS to print your output data set and its contents. You can use PROC SQL along with the SASECRSP engine to create a view of your SAS data set.

Table 2.1. Data Sets Available

Dataset	Fields	Label	Type
STKHEAD	PERMNO	PERMNO	Numeric
Header Identification	PERMCO	PERMCO	Numeric
and Summary Data	COMPNO	Nasdaq Company Number	Numeric
	ISSUNO	Nasdaq Issue Number	Numeric
	HEXCD	Exchange Code Header	Numeric
	HSHRCD	Share Code Header	Numeric
	HSICCD	Standard Industrial Classification Code	Numeric
	BEGDT	Begin of Stock Data	Numeric
	ENDDT	End of Stock Data	Numeric
	DLSTCD	Delisting Code Header	Numeric
	HCUSIP	CUSIP Header	Character
	HTICK	Ticker Symbol Header	Character
	HCOMNAM	Company Name Header	Character
NAMES	PERMNO	PERMNO	Numeric
Name History Array	NAMEDT	Names Date	Numeric
	NAMEENDDT	Names Ending Date	Numeric
	SHRCD	Share Code	Numeric
	EXCHCD	Exchange Code	Numeric
	SICCD	Standard Industrial Classification Code	Numeric
	NCUSIP	CUSIP	Numeric
	TICKER	Ticker Symbol	Character
	COMNAM	Company Name	Character
	SHRCLS	Share Class	Numeric
DISTS	PERMNO	PERMNO	Numeric
Distribution Event	DISTCD	Distribution Code	Numeric
Array	DIVAMT	Dividend Cash Amount	Numeric
	FACPR	Factor to Adjust Price	Numeric
	FACSHR	Factor to Adjust Share	Numeric

Table 2.1. (continued)

Dataset	Fields	Label	Type
	DCLRDT	Distribution Declaration Date	Numeric
	EXDT	Ex-Distribution Date	Numeric
	RCRDDT	Record Date	Numeric
	PAYDT	Payment Date	Numeric
	ACPERM	Acquiring PERMNO	Numeric
	ACCOMP	Acquiring PERMCO	Numeric
SHARES Shares Outstanding Observation Array	PERMNO	PERMNO	Numeric
	SHROUT	Shares Outstanding	Numeric
	SHRSDT	Shares Outstanding	Numeric
		Observation Date	Numeric
	SHRENDDT	Shares Outstanding Observation End Date	Numeric
	SHRFLG	Shares Outstanding Observation Flag	Numeric
DELIST Delisting History Array	PERMNO	PERMNO	Numeric
	DLSTDT	Delisting Date	Numeric
	DLSTCD	Delisting Code	Numeric
	NWPERM	New PERMNO	Numeric
	NWCOMP	New PERMCO	Numeric
	NEXTDT	Delisting Next Price Date	Numeric
	DLAMT	Delisting Amount	Numeric
	DLRETX	Delisting Return Without Dividends	Numeric
	DLPRC	Delisting Price	Numeric
	DLPDT	Delisting Amount Date	Numeric
	DLRET	Delisting Return	Numeric
NASDIN Nasdaq Information Array	PERMNO	PERMNO	Numeric
	TRTSCD	Nasdaq Traits Code	Numeric
	TRTSDT	Nasdaq Traits Date	Numeric
	TRTSENDDT	Nasdaq Traits End Date	Numeric
	NMSIND	Nasdaq National Market Indicator	Numeric
	MMCNT	Market Maker Count	Numeric
	NSDINX	Nasd Index Code	Numeric
PRC Price or Bid/Ask Average Time Series	PERMNO	PERMNO	Numeric
	CALDT	Calendar Trading Date	Numeric
	PRC	Price or Bid/Ask Aver	Numeric
RET Returns Time Series	PERMNO	PERMNO	Numeric
	CALDT	Calendar Trading Date	Numeric
	RET	Returns	Numeric
ASKHI Ask or High Time Series	PERMNO	PERMNO	Numeric
	CALDT	Calendar Trading Date	Numeric
	ASKHI	Ask or High	Numeric

Table 2.1. (continued)

Dataset	Fields	Label	Type
BIDLO Bid or Low Time Series	PERMNO CALDT BIDLO	PERMNO Calendar Trading Date Bid or Low	Numeric Numeric Numeric
BID Bid Time Series	PERMNO CALDT BID	PERMNO Calendar Trading Date Bid	Numeric Numeric Numeric
ASK Ask Time Series	PERMNO CALDT ASK	PERMNO Calendar Trading Date Ask	Numeric Numeric Numeric
RETX Returns without Dividends Time Series	PERMNO CALDT RETX	PERMNO Calendar Trading Date Returns w/o Dividends	Numeric Numeric Numeric
SPREAD Spread Between Bid and Ask Time Series	PERMNO CALDT SPREAD	PERMNO Calendar Trading Date Spread Between Bid Ask	Numeric Numeric Numeric
ALTPRC Price Alternate Time Series	PERMNO CALDT ALTPRC	PERMNO Calendar Trading Date Price Alternate	Numeric Numeric Numeric
VOL Volume Time Series	PERMNO CALDT VOL	PERMNO Calendar Trading Date Volume	Numeric Numeric Numeric
NUMTRD Number of Trades Time Series	PERMNO CALDT NUMTRD	PERMNO Calendar Trading Date Number of Trades	Numeric Numeric Numeric
ALTPRCDT Alternate Price Date Time Series	PERMNO CALDT ALTPRCDT	PERMNO Calendar Trading Date Alternate Price Date	Numeric Numeric Numeric
PORT1 Portfolio Data for Portfolio Type 1	PERMNO CALDT PORT1 STAT1	PERMNO Calendar Trading Date Portfolio Assignment for Portfolio Type 1 Portfolio Statistic for Portfolio Type 1	Numeric Numeric Numeric Numeric Numeric Numeric
PORT2 Portfolio Data for Portfolio Type 2	PERMNO CALDT PORT2 STAT2	PERMNO Calendar Trading Date Portfolio Assignment for Portfolio Type 2 Portfolio Statistic for Portfolio Type 2	Numeric Numeric Numeric Numeric Numeric Numeric
PORT3 Portfolio Data for Portfolio Type 3	PERMNO CALDT PORT3 STAT3	PERMNO Calendar Trading Date Portfolio Assignment for Portfolio Type 3 Portfolio Statistic for Portfolio Type 3	Numeric Numeric Numeric Numeric Numeric Numeric

Table 2.1. (continued)

Dataset	Fields	Label	Type
PORT4	PERMNO	PERMNO	Numeric
Portfolio Data for	CALDT	Calendar Trading Date	Numeric
Portfolio Type 4	PORT4	Portfolio Assignment	Numeric
		for Portfolio Type 4	Numeric
	STAT4	Portfolio Statistic	Numeric
		for Portfolio Type 4	Numeric
PORT5	PERMNO	PERMNO	Numeric
Portfolio Data for	CALDT	Calendar Trading Date	Numeric
Portfolio Type 5	PORT5	Portfolio Assignment	Numeric
		for Portfolio Type 5	Numeric
	STAT5	Portfolio Statistic	Numeric
		for Portfolio Type 5	Numeric
PORT6	PERMNO	PERMNO	Numeric
Portfolio Data for	CALDT	Calendar Trading Date	Numeric
Portfolio Type 6	PORT6	Portfolio Assignment	Numeric
		for Portfolio Type 6	Numeric
	STAT6	Portfolio Statistic	Numeric
		for Portfolio Type 6	Numeric
PORT7	PERMNO	PERMNO	Numeric
Portfolio Data for	CALDT	Calendar Trading Date	Numeric
Portfolio Type 7	PORT7	Portfolio Assignment	Numeric
		for Portfolio Type 7	Numeric
	STAT7	Portfolio Statistic	Numeric
		for Portfolio Type 7	Numeric
PORT8	PERMNO	PERMNO	Numeric
Portfolio Data for	CALDT	Calendar Trading Date	Numeric
Portfolio Type 8	PORT8	Portfolio Assignment	Numeric
		for Portfolio Type 8	Numeric
	STAT8	Portfolio Statistic	Numeric
		for Portfolio Type 8	Numeric
PORT9	PERMNO	PERMNO	Numeric
Portfolio Data for	CALDT	Calendar Trading Date	Numeric
Portfolio Type 9	PORT9	Portfolio Assignment	Numeric
		for Portfolio Type 9	Numeric
	STAT9	Portfolio Statistic	Numeric
		for Portfolio Type 9	Numeric

References

Center for Research in Security Prices (2000), *CRSPAccess Database Format Data Description Guide*, Chicago: The University of Chicago Graduate School of Business, [http://www.crsp.uchicago.edu/file_guides/stock_ind_data_descriptions.pdf].

Center for Research in Security Prices (2000), *CRSPAccess Database Format Programmer's Guide*, Chicago: The University of Chicago Graduate School of Business, [http://www.crsp.uchicago.edu/file_guides/stock_ind_programming.pdf].

Center for Research in Security Prices (2000), *CRSPAccess Database Format Release Notes*, Chicago: The University of Chicago Graduate School of Business, [www.crsp.uchicago.edu/file_guides/ca_release_notes.pdf].

Center for Research in Security Prices (2000), *CRSPAccess Database Format Utilities Guide*, Chicago: The University of Chicago Graduate School of Business, [http://www.crsp.uchicago.edu/file_guides/stock_ind_utilities.pdf].

Center for Research in Security Prices (2000), *CRSP SFA Database Format Guide*, Chicago: The University of Chicago Graduate School of Business, [http://www.crsp.uchicago.edu/file_guides/stock_ind_sfa.pdf].

Chapter 3
The SASEFAME Interface Engine

Chapter Table of Contents

Chapter 3
The SASEFAME Interface Engine

Overview

The SASEFAME interface engine enables SAS users to access and process time series data residing in a FAME database, and provides a seamless interface between FAME and SAS data processing.

The SASEFAME engine uses the LIBNAME statement to enable you to specify which time series you would like to read from the FAME database, and how you would like to convert the selected time series to the same time scale. The SAS DATA step can then be used to perform further subsetting and to store the resulting time series into a SAS data set. You can perform more analysis if desired either in the same SAS session or in another session at a later time.

Getting Started

Structure of a SAS Data Set Containing Time Series Data

SAS requires time series data to be in a specific form recognizable by the SAS System. This form is a two-dimensional array, called a SAS data set, whose columns correspond to series variables and whose rows correspond to measurements of these variables at certain time periods. The time periods at which observations are recorded can be included in the data set as a time ID variable. The SASEFAME engine does provide a time ID variable by the name of DATE.

Reading and Converting FAME Database Time Series

The SASEFAME engine supports reading and converting time series stored in FAME databases. The SASEFAME engine uses the FAME WORK database to temporarily store the converted time series. Only the series specified by the FAME wildcard are written to the FAME WORK database.

Using the SAS DATA Step

If desired, you can store the converted series in a SAS data set by using the SAS DATA step. You can also perform other operations on your data inside the DATA step. Once your data is stored in a SAS data set you can use it as you would any other SAS data set.

Using SAS Procedures

You can print the output SAS data set by using the PRINT procedure and report information concerning the contents of your data set by using the CONTENTS procedure, as in Example 3.1. You can create a view of the FAME data base by using the SQL procedure to create your view using the SASEFAME engine in your *libref*, along with the using clause. See Example 3.5.

Syntax

The SASEFAME engine uses standard engine syntax. Options used by SASEFAME are summarized in the table below.

Description	Statement	Option
specifies the FAME frequency and the FAME technique.	LIBNAME *libref* SASEFAME	CONVERT=
specifies a FAME wildcard to match data object series names within the FAME database, which limits the selection of time series that are included in the SAS data set.	LIBNAME *libref* SASEFAME	WILDCARD=

The LIBNAME *libref* SASEFAME Statement

LIBNAME *libref* **SASEFAME ’physical name’** *options;*

The following options can be used in the LIBNAME libref SASEFAME statement:

CONVERT=*(FREQ=fame_frequency TECH= fame_technique)*

specifies the FAME frequency and the FAME technique just as you would in the FAME CONVERT function. There are four possible values for *fame_technique*: CONSTANT (default), CUBIC, DISCRETE, or LINEAR. All FAME frequencies except PPY and YPP are supported by the SASEFAME engine. For a more complete discussion of FAME frequencies and SAS time intervals see the section “Mapping FAME Frequencies to SAS Time Intervals” on page 26. For all possible *fame_frequency* values, refer to “Understanding Frequencies” in the *User’s Guide to FAME*. As an example,

```
LIBNAME libref sasefame ’physical-name’
   CONVERT=(TECH=CONSTANT FREQ=TWICEMONTHLY);
```

WILDCARD=*"fame_wildcard"*

By default, the SASEFAME engine reads all time series in the FAME database that you name in your SASEFAME libref. You can limit the time series read from the FAME database by specifying the WILDCARD= option on your LIBNAME statement. The *fame_wildcard* is a quoted string containing the FAME wildcard you wish to use. The wildcard is used against the data object names (time series only) in the FAME database that resides in the library you are in the process of assigning. For more information about wildcarding, see “Specifying Wildcards” in the *User’s Guide to FAME*.

For example, to read all time series in the TEST library being accessed by the SASEFAME engine, you would specify

```
LIBNAME test sasefame 'physical name of test data base'
   WILDCARD="?";
```

To read series with names such as A_DATA, B_DATA, C_DATA, you could specify

```
LIBNAME test sasefame 'physical name of test data base'
   WILDCARD="^_DATA";
```

When you use the WILD= option, you are limiting the number of time series that are read and converted to the desired frequency. This option can help you save resources when processing large databases or when processing a large number of observations, such as daily, hourly, or minutely frequencies. Since the SASEFAME engine uses the FAME WORK database to store the converted time series, using wildcards is recommended to prevent your WORK space from getting too large.

Details

The SAS Output Data Set

You can use the SAS DATA step to write the FAME converted series to a SAS data set. This allows the user the ability to easily analyze the data using SAS. You can specify the name of the output data set on the DATA statement. This causes the engine supervisor to create a SAS data set using the specified name in either the SAS WORK library, or if specified, the USER library. For more about naming your SAS data set see the section “Characteristics of SAS Data Libraries” in *SAS Language Reference: Dictionary*.

The contents of the SAS data set include the DATE of each observation, the name of each series read from the FAME database as specified by the WILDCARD option, and the label or FAME description of each series. Missing values are represented as '.' in the SAS data set. You can use PROC PRINT and PROC CONTENTS to print your output data set and its contents. You can use PROC SQL along with the SASEFAME engine to create a view of your SAS data set.

The DATE variable in the SAS data set contains the date of the observation. For FAME weekly intervals that end on a Friday, FAME reports the date on the Friday that ends the week whereas SAS reports the date on the Saturday that begins the week. A more detailed discussion of how to map FAME frequencies to SAS Time Intervals follows.

Mapping FAME Frequencies to SAS Time Intervals

The following table summarizes the mapping of FAME frequencies to SAS time intervals. It is important to note that FAME frequencies often have a sample unit in parentheses following the keyword frequency. This sample unit is an end-of-interval unit. SAS dates are represented using begin-of-interval notation. For more on SAS time intervals, see “Date Intervals, Formats, and Functions” in *SAS/ETS User’s Guide*. For more on FAME frequencies, see the section “Understanding Frequencies” in the *User’s Guide to FAME*.

FAME FREQUENCY	SAS TIME INTERVAL
WEEKLY (SUNDAY)	WEEK.2
WEEKLY (MONDAY)	WEEK.3
WEEKLY (TUESDAY)	WEEK.4
WEEKLY (WEDNESDAY)	WEEK.5
WEEKLY (THURSDAY)	WEEK.6
WEEKLY (FRIDAY)	WEEK.7
WEEKLY (SATURDAY)	WEEK.1
BIWEEKLY (ASUNDAY)	WEEK2.2
BIWEEKLY (AMONDAY)	WEEK2.3
BIWEEKLY (ATUESDAY)	WEEK2.4
BIWEEKLY (AWEDNESDAY)	WEEK2.5
BIWEEKLY (ATHURSDAY)	WEEK2.6
BIWEEKLY (AFRIDAY)	WEEK2.7
BIWEEKLY (ASATURDAY)	WEEK2.1
BIWEEKLY (BSUNDAY)	WEEK2.9
BIWEEKLY (BMONDAY)	WEEK2.10
BIWEEKLY (BTUESDAY)	WEEK2.11
BIWEEKLY (BWEDNESDAY)	WEEK2.12
BIWEEKLY (BTHURSDAY)	WEEK2.13
BIWEEKLY (BFRIDAY)	WEEK2.14
BIWEEKLY (BSATURDAY)	WEEK2.8
BIMONTHLY (NOVEMBER)	MONTH2.2
BIMONTHLY	MONTH2.1
QUARTERLY (OCTOBER)	QTR.2
QUARTERLY (NOVEMBER)	QTR.3
QUARTERLY	QTR.1

FAME FREQUENCY	SAS TIME INTERVAL
ANNUAL (JANUARY)	YEAR.2
ANNUAL (FEBRUARY)	YEAR.3
ANNUAL (MARCH)	YEAR.4
ANNUAL (APRIL)	YEAR.5
ANNUAL (MAY)	YEAR.6
ANNUAL (JUNE)	YEAR.7
ANNUAL (JULY)	YEAR.8
ANNUAL (AUGUST)	YEAR.9
ANNUAL (SEPTEMBER)	YEAR.10
ANNUAL (OCTOBER)	YEAR.11
ANNUAL (NOVEMBER)	YEAR.12
ANNUAL	YEAR.1
SEMIANNUAL (JULY)	SEMIYEAR.2
SEMIANNUAL (AUGUST)	SEMIYEAR.3
SEMIANNUAL (SEPTEMBER)	SEMIYEAR.4
SEMIANNUAL (OCTOBER)	SEMIYEAR.5
SEMIANNUAL (NOVEMBER)	SEMIYEAR.6
SEMIANNUAL	SEMIYEAR.1
YPP	not supported
PPY	not supported
SECONDLY	SECOND
MINUTELY	MINUTE
HOURLY	HOUR
DAILY	DAY
BUSINESS	WEEKDAY
TENDAY	TENDAY
TWICEMONTHLY	SEMIMONTH
MONTHLY	MONTH

Examples

Example 3.1. Converting an Entire FAME Database

To enable conversion of all Time Series no wildcard is specified, so the default “?” wildcard is used. Always consider both the number of time series and the number of observations generated by the conversion process. The converted series are stored in the FAME WORK database during the SAS DATA step. You may further limit your resulting SAS data set by using KEEP, DROP, or WHERE statements inside your data step.

The following statements convert a FAME database and print out its contents:

```
libname famedir sasefame '.'
                convert=(freq=annual technique=constant);

libname mydir '/mine/data/europe/sas/oecdir';

data mydir.a;  /* add data set to mydir */
   set famedir.oecd1;
   /* do nothing special */
   run;

proc print data=mydir.a; run;
```

In the above example, the FAME database is called oecd1.db and it resides in the famedir directory. The DATA statement names the SAS output data set ’a’ which will reside in mydir. All time series in the FAME oecd1.db data base will be converted to an annual frequency and stored in the mydir.a SAS data set. The PROC PRINT statement creates a listing of all of the observations in the mydir.a SAS data set.

Example 3.2. Reading Time Series from the FAME Database

Use the FAME wildcard option to limit the number of series converted. For example, suppose you want to read only series starting with “WSPCA”. You could use the following code:

```
libname lib1 sasefame '/mine/data/econ_fame/sampdir'
             wildcard="wspca?"
             convert=(technique=constant freq=twicemonthly );

libname lib2 '/mine/data/econ_sas/sampdir';

data lib2.twild(label='Annual Series from the FAMEECON.db');
   set lib1.subecon;
   /* keep only  */
   keep date wspca;
   run;

proc contents data=lib2.twild; run;

proc print data=lib2.twild; run;
```

The wildcard=“wspca?” option limits reading only those series whose names begin with WSPCA. The SAS KEEP statement further restricts the SAS data set to include only the series named WSPCA and the DATE variable. The time interval used for the conversion is TWICEMONTHLY.

Example 3.3. Writing Time Series to the SAS Data Set

You can use the KEEP or DROP statement to include or exclude certain series names from the SAS data set.

```
libname famedir sasefame '.'
        convert=(freq=annual technique=constant);

libname mydir '/mine/data/europe/sas/oecdir';

data mydir.a;  /* add data set to mydir */
   set famedir.oecd1;
   drop ita_dird--jpn_herd tur_dird--usa_herd;
   run;

proc print data=mydir.a; run;
```

You can rename your SAS variables by using the RENAME statement.

```
option validvarname=any;

libname famedir sasefame '.'
        convert=(freq=annual technique=constant);

libname mydir '/mine/data/europe/sas/oecdir';

data mydir.a;  /* add data set to mydir */
   set famedir.oecd1;
   /* keep and rename  */
   keep date ita_dird--jpn_herd tur_dird--usa_herd;
   rename ita_dird='Italy.dirdes'n
          jpn_dird='Japan.dirdes'n
          tur_dird='Turkey.dirdes'n
          usa_dird='UnitedStates.dirdes'n ;
   run;

proc print data=mydir.a; run;
```

Example 3.4. Limiting the Time Range of Data

You may also limit the time range of the data in the SAS data set by using the WHERE statement in the data step to process the time ID variable DATE only when it falls in the range you are interested in.

```
libname famedir SASEFAME '.'
        convert=(freq=annual technique=constant);

libname mydir '/mine/data/europe/sas/oecdir';

data mydir.a;  /* add data set to mydir */
   set famedir.oecd1;
   /* where only  */
   where date between '01jan88'd and '31dec90'd;
   run;

proc print data=mydir.a; run;
```

All data for 1988, 1989, and 1990 are included in the SAS data set. See the *SAS Language: Reference*, *Version 7* for more information on KEEP, DROP, RENAME and WHERE statements.

Example 3.5. Creating a View Using the SQL Procedure and SASEFAME

This example creates a view using the SQL procedure's from and using clauses. See *SQL Procedure Guide, Version 7* for details on SQL views.

```
title1 'famesql5: PROC SQL Dual Embedded Libraries w/ FAME option';
options validvarname=any;

/* Dual Embedded Library Allocations (With FAME Option) */
/********************************************************/

/* OECD1 Database */
/******************/

title2 'OECD1: Dual Embedded Library Allocations
                  with FAME Option';
proc sql;
   create view fameview as
   select date, 'fin.herd'n
   from lib1.oecd1
   using libname lib1 sasefame '/economic/databases/testdat'
                        convert=(tech=constant freq=annual),
         libname temp '/usr/local/scratch/mine'
   quit;

title2 'OECD1: Print of View from Embedded Library
                  with FAME Option';
proc print data=fameview;
run;
```

Output 3.5.1. Printout of the FAME View of OECD Data

```
              PROC SQL Dual Embedded Libraries w/ FAME option
          OECD1: Print of View from Embedded Library with Option

                         Obs    DATE    FIN.HERD

                          1     1985     1097.0
                          2     1986     1234.0
                          3     1987     1401.3
                          4     1988     1602.0
                          5     1989     1725.5
                          6     1990     1839.0
                          7     1991          .
```

```
/* SUBECON Database */
/********************/

title2 'SUBECON: Dual Embedded Library Allocations
                 with FAME Option';
proc sql;
   create view fameview as
   select date, gaa
```

```
   from lib1.subecon
   using libname lib1 sasefame '/economic/databases/testdat'
                      convert=(tech=constant freq=annual),
         libname temp '/usr/local/scratch/mine'
   quit;

title2 'SUBECON: Print of View from Embedded Library
                      with FAME Option';
proc print data=fameview;
run;
```

Output 3.5.2. Printout of the FAME View of DRI Basic Economic Data

```
             PROC SQL Dual Embedded Libraries w/ FAME option
        SUBECON: Print of View from Embedded Library with Option

                        Obs    DATE       GAA

                          1    1946          .
                          2    1947          .
                          3    1948     23174.00
                          4    1949     19003.00
                          5    1950     24960.00
                          6    1951     21906.00
                          7    1952     20246.00
                          8    1953     20912.00
                          9    1954     21056.00
                         10    1955     27168.00
                         11    1956     27638.00
                         12    1957     26723.00
                         13    1958     22929.00
                         14    1959     29729.00
                         15    1960     28444.00
                         16    1961     28226.00
                         17    1962     32396.00
                         18    1963     34932.00
                         19    1964     40024.00
                         20    1965     47941.00
                         21    1966     51429.00
                         22    1967     49164.00
                         23    1968     51208.00
                         24    1969     49371.00
                         25    1970     44034.00
                         26    1971     52352.00
                         27    1972     62644.00
                         28    1973     81645.00
                         29    1974     91028.00
                         30    1975     89494.00
                         31    1976    109492.00
                         32    1977    130260.00
                         33    1978    154357.00
                         34    1979    173428.00
                         35    1980    156096.00
                         36    1981    147765.00
                         37    1982    113216.00
                         38    1983    133495.00
                         39    1984    146448.00
                         40    1985    128521.99
                         41    1986    111337.99
                         42    1987    160785.00
                         43    1988    210532.00
                         44    1989    201637.00
                         45    1990    218702.00
                         46    1991    210666.00
                         47    1992          .
                         48    1993          .
```

```
title2 'DB77: Dual Embedded Library Allocations
                with FAME Option';
proc sql;
   create view fameview as
   select date, ann, 'qandom.x'n
   from lib1.db77
   using libname lib1 sasefame '/economic/databases/testdat'
                      convert=(tech=constant freq=annual),
```

```
        libname temp '/usr/local/scratch/mine'
  quit;

 title2 'DB77: Print of View from Embedded Library
                  with FAME Option';
 proc print data=fameview;
 run;
```

Output 3.5.3. Printout of the FAME View of DB77 Data

```
              PROC SQL Dual Embedded Libraries w/ FAME option
           DB77: Print of View from Embedded Library with Option

                     Obs    DATE    ANN    QANDOM.X

                       1    1959      .     0.56147
                       2    1960      .     0.51031
                       3    1961      .      .
                       4    1962      .      .
                       5    1963      .      .
                       6    1964      .      .
                       7    1965      .      .
                       8    1966      .      .
                       9    1967      .      .
                      10    1968      .      .
                      11    1969      .      .
                      12    1970      .      .
                      13    1971      .      .
                      14    1972      .      .
                      15    1973      .      .
                      16    1974      .      .
                      17    1975      .      .
                      18    1976      .      .
                      19    1977      .      .
                      20    1978      .      .
                      21    1979      .      .
                      22    1980    100      .
                      23    1981    101      .
                      24    1982    102      .
                      25    1983    103      .
                      26    1984    104      .
                      27    1985    105      .
                      28    1986    106      .
                      29    1987    107      .
                      30    1988    109      .
                      31    1989    111      .
```

```
/* DRIECON Database */
/*******************/

title2 'DRIECON: Dual Embedded Library Allocations
                 with FAME Option';
proc sql;
   create view fameview as
   select date, husts
   from lib1.driecon
   using libname lib1 sasefame '/economic/databases/testdat'
                      convert=(tech=constant freq=annual),
         libname temp '/usr/local/scratch/mine'
   quit;
```

```
title2 'DRIECON: Print of View from Embedded Library
                 with FAME Option';
proc print data=fameview;
run;
```

Note that the SAS option VALIDVARNAME=ANY was used at the top of this example due to special characters being present in the time series names. The output from this example shows how each FAME view is the output of the SASEFAME engine's processing. Note that different engine options could have been used in the USING LIBNAME clause if desired.

Output 3.5.4. Printout of the FAME View of DRI Basic Economic Data

```
            PROC SQL Dual Embedded Libraries w/ FAME option
       DRIECON: Print of View from Embedded Library with Option

                      Obs    DATE     HUSTS

                        1    1947    1.26548
                        2    1948    1.33470
                        3    1949    1.43617
                        4    1950    1.90041
                        5    1951    1.43759
                        6    1952    1.44883
                        7    1953    1.40279
                        8    1954    1.53525
                        9    1955    1.61970
                       10    1956    1.32400
                       11    1957    1.17300
                       12    1958    1.31717
                       13    1959    1.53450
                       14    1960    1.25505
                       15    1961    1.31188
                       16    1962    1.45996
                       17    1963    1.58858
                       18    1964    1.53950
                       19    1965    1.46966
                       20    1966    1.16507
                       21    1967    1.28573
                       22    1968    1.50314
                       23    1969    1.48531
                       24    1970    1.43565
                       25    1971    2.03775
                       26    1972    2.36069
                       27    1973    2.04307
                       28    1974    1.32855
                       29    1975    1.16164
                       30    1976    1.53468
                       31    1977    1.96218
                       32    1978    2.00184
                       33    1979    1.71847
                       34    1980    1.29990
                       35    1981    1.09574
                       36    1982    1.05862
                       37    1983    1.70580
                       38    1984    1.76351
                       39    1985    1.74258
                       40    1986    1.81205
                       41    1987    1.62914
                       42    1988    1.48748
                       43    1989    1.38218
                       44    1990    1.20161
                       45    1991    1.00878
                       46    1992    1.20159
                       47    1993    1.29201
                       48    1994    1.44684
                       49    1995    1.35845
                       50    1996    1.48336
```

References

DRI/McGraw-Hill (1997), *DataLink*, Lexington, MA.

DRI/McGraw-Hill Data Search and Retrieval for Windows (1996), *DRIPRO User's Guide*, Lexington, MA.

FAME Information Services (1995), *User's Guide to FAME*, Ann Arbor, Michigan.

FAME Information Services (1995), *Reference Guide to Seamless C HLI*, Ann Arbor, Michigan.

FAME Information Services (1995), *Command Reference for Release 7.6, Vols. 1 and 2*, Ann Arbor, Michigan.

Organization For Economic Cooperation and Development (1992), *Annual National Accounts: Volume I. Main Aggregates Content Documentation for Magnetic Tape Subscription*, Paris, France.

Organization For Economic Cooperation and Development (1992), *Annual National Accounts: Volume II. Detailed Tables Technical Documentation for Magnetic Tape Subscription*, Paris, France.

Organization For Economic Cooperation and Development (1992), *Main Economic Indicators Database Note*, Paris, France.

Organization For Economic Cooperation and Development (1992), *Main Economic Indicators Inventory*, Paris, France.

Organization For Economic Cooperation and Development (1992), *Main Economic Indicators OECD Statistics on Magnetic Tape Document*, Paris, France.

Organization For Economic Cooperation and Development (1992), *OECD Statistical Information Research and Inquiry System Magnetic Tape Format Documentation*, Paris, France.

Organization For Economic Cooperation and Development (1992), *Quarterly National Accounts Inventory of Series Codes*, Paris, France.

Organization For Economic Cooperation and Development (1992), *Quarterly National Accounts Technical Documentation*, Paris, France.

Part 2
Procedure Reference

Table of Contents

Chapter 4
The VARMAX Procedure

Chapter Table of Contents

Chapter 4
The VARMAX Procedure

Overview

Given a multivariate time series, the VARMAX procedure estimates the model parameters and generates forecasts associated with Vector AutoRegressive and Moving-Average processes with eXogenous regressors (VARMAX) models. Often, economic or financial variables are not only contemporaneously correlated to each other, they are also correlated to each other's past values. The VARMAX procedure can be used 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 both the dynamic relationship between the dependent variables and between the dependent and independent variables.

VARMAX models are defined in terms of the orders of the autoregressive or moving-average process (or both). When you use the VARMAX procedure, these orders can be specified by options or they can be automatically determined. Criteria for automatically determining these orders include

- Akaike Information Criterion (AIC)
- Corrected AIC (AICC)
- Hannan-Quinn (HQ) Criterion
- Final Prediction Error (FPE)
- Schwarz Bayesian Criterion (SBC), also known as Bayesian Information Criterion (BIC)

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

For situations where the stationarity of the time series is in question, the VARMAX procedure provides tests to aid in determining the presence of unit roots and cointegration. These tests include

- Dickey-Fuller test
- Johansen cointegration test for nonstationary vector processes of integrated order one

- Stock-Watson common trends test for the possibility of cointegration among nonstationary vector processes of integrated order one
- Johansen cointegration test for nonstationary vector processes of integrated order two

For stationary vector times series (or nonstationary series made stationary by appropriate differencing), the VARMAX procedure provides for both Vector AutoRegressive (VAR) and Bayesian Vector AutoRegressive (BVAR) models. To cope with the problem of high dimensionality in the parameters of the VAR model, the VARMAX procedure provides both the Vector Error Correction Model (VECM) and Bayesian Vector Error Correction Model (BVECM). Bayesian models are used when prior information about the model parameters is available. The VARMAX procedure can allow exogenous (independent) variables with their distributed lags to influence endogenous (dependent) variables. The model parameter estimation methods are

- Least Squares (LS)
- Maximum Likelihood (ML)

The VARMAX procedure provides various hypothesis tests of long-run effects and adjustment coefficients using the likelihood ratio test based on Johansen cointegration analysis. The VARMAX procedure offers the likelihood ratio test of the weak exogeneity for each variable.

After fitting the model parameters, the VARMAX procedure provides for model checks and residual analysis using the following tests:

- Durbin-Watson (DW) statistics
- F test for autoregressive conditional heteroscedastic (ARCH) disturbance
- F test for AR disturbance
- Jarque-Bera normality test
- Portmanteau test

Forecasting is one of the main objectives of multivariate time series analysis. After successfully fitting the VAR, VARX, BVAR, VECM, and BVECM model parameters, the VARMAX procedure computes predicted values based on the parameter estimates and the past values of the vector time series.

The VARMAX procedure supports several modeling features, including

- seasonal deterministic terms
- subset models
- multiple regression with distributed lags

- dead-start model that does not have present values of the exogenous variables

The VARMAX procedure provides a Granger-Causality test to determine the Granger-causal relationships between two distinct groups of variables. It also provides

- infinite order AR representation
- impulse response function (or infinite order MA representation)
- 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

Getting Started

This section outlines the use of the VARMAX procedure and gives five different examples of the kind of models supported.

Vector Autoregressive Process

Let $\mathbf{y}_t = (y_{1t}, \ldots, y_{kt})'$, $t = 0, 1, \ldots$, 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 the $\boldsymbol{\epsilon}_t$ is a vector white noise process with $\boldsymbol{\epsilon}_t = (\epsilon_{1t}, \ldots, \epsilon_{kt})'$ 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 $t \neq s$.

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 utilizing the additional information available from the related series and their forecasts.

Example of Vector Autoregressive Model

Consider the first-order stationary vector autoregressive model

$$\mathbf{y}_t = \left(\begin{array}{rr} 1.2 & -0.5 \\ 0.6 & 0.3 \end{array} \right) \mathbf{y}_{t-1} + \boldsymbol{\epsilon}_t, \quad \text{with } \Sigma = \left(\begin{array}{rr} 1.0 & 0.5 \\ 0.5 & 1.25 \end{array} \right).$$

The following IML procedure statements check whether the specified vector time series model is stationary and 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};
   /* to 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 $\mathbf{y}_t$ shown in Figure 4.1:

```
   data simul1;
      set simul1;
      t+1;

   proc gplot data=simul1;
      symbol1 v = none i = join l = 1;
      symbol2 v = none i = join l = 2;
      plot y1 * t = 1
           y2 * t = 2 / overlay;
   run;
```

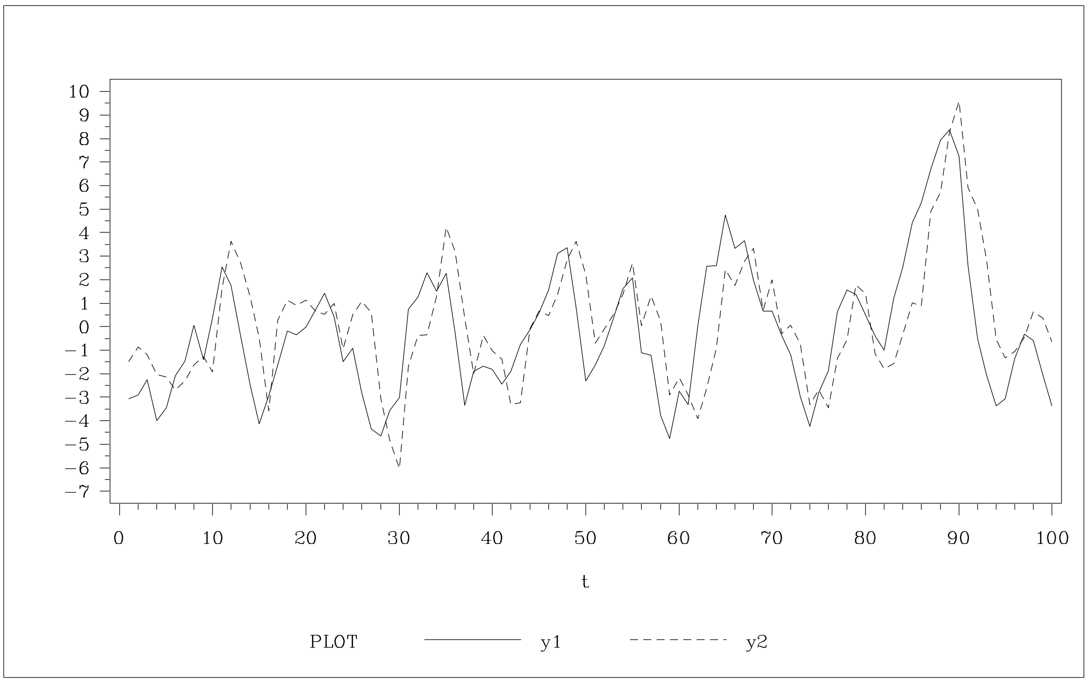

Figure 4.1. Plot of Generated Data Process

The following statements fit a VAR(1) model to the simulated data. You first specify the input data set in the PROC VARMAX statement. Then, you use the MODEL statement to read the time series y_1 and y_2. To estimate a VAR model with mean zero, you specify the order of the autoregressive model with the P= option and the NOINT option. The MODEL statement fits the model to the data and prints parameter estimates and various diagnostic tests. The LAGMAX=3 option is used to print the output for the residual diagnostic checks. For the forecasts, you specify the OUTPUT statement. If you wish to forecast five steps ahead, you use the option LEAD=5.

The VARMAX procedure output is shown in Figure 4.2 through Figure 4.9.

```
proc varmax data=simul1;
   id date interval=year;
   model y1 y2 / p=1 noint lagmax=3;
   output lead=5;
run;
```

```
                              The VARMAX Procedure

                       Number of Observations        100
                       Number of Pairwise Missing      0


Variable   Type    NoMissN         Mean       StdDev          Min          Max

y1         DEP         100     -0.21653      2.78210     -4.75826      8.37032
y2         DEP         100      0.16905      2.58184     -6.04718      9.58487



                              The VARMAX Procedure

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

Figure 4.2. Descriptive Statistics and Model Type

The VARMAX procedure first displays descriptive statistics and the type of the fitted model for the simulated data, as shown in Figure 4.2.

```
                              The VARMAX Procedure

                            AR Coefficient Estimates

                 Lag    Variable            y1             y2

                   1    y1             1.15977       -0.51058
                        y2             0.54634        0.38499


                           Model Parameter Estimates

 Equation  Parameter      Estimate    Std Error   T Ratio  Prob>|T|  Variable

 y1(t)     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(t)     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)
```

Figure 4.3. Parameter Estimates

Figure 4.3 shows the AR coefficient matrix in terms of lag 1, the parameter estimates, and their significances that indicate how well the model fits the data. The first column gives the left-hand-side variable of the equation; the second column, the parameter name ARl_i_j, which indicates the (i, j)th element of the lag l autoregressive coefficient; the last column, the regressor corresponding to its parameter.

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 VARMAX Procedure

                Covariance Matrix for the Innovation

               Variable             y1             y2

               y1              1.28875        0.39751
               y2              0.39751        1.41839


                        Information Criteria

         AICC(Corrected AIC)                         0.554443
         HQC(Hannan-Quinn Criterion)                 0.595201
         AIC(Akaike Information Criterion)           0.552777
         SBC(Schwarz Bayesian Criterion)              0.65763
         FPEC(Final Prediction Error Criterion)      1.738092
```

Figure 4.4. Innovation Covariance Estimates and Information Criteria

The table in Figure 4.4 shows the innovation covariance matrix estimates and the various information criteria results. The smaller value of information criteria fits the data better when it is compared to other models.

The VARMAX Procedure

Residual Cross-Correlation Matrices

Lag	Variable	y1	y2
0	y1	1.00000	0.28113
	y2	0.28113	1.00000
1	y1	0.01309	0.02385
	y2	-0.05569	-0.07328
2	y1	0.05277	0.06052
	y2	0.00847	-0.04307
3	y1	-0.00163	0.06800
	y2	-0.01644	0.05422

Schematic Representation of Residual Cross Correlations

Variable/ Lag	0	1	2	3
y1	++	..	..	..
y2	++	..	..	..

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

Figure 4.5. Multivariate Diagnostic Checks

The VARMAX Procedure

Portmanteau Test for Residual Cross Correlations

To Lag	Chi-Square	DF	Prob>ChiSq
2	1.84	4	0.7659
3	2.57	8	0.9582

Figure 4.6. Multivariate Diagnostic Checks Continued

Figure 4.5 and Figure 4.6 show tests for white noise residuals. The output shows that you cannot reject the hypothesis that the residuals are uncorrelated.

The VARMAX Procedure

Univariate Model Diagnostic Checks

Variable	R-square	StdDev	F Value	Prob>F
y1	0.8369	1.1352	497.67	<.0001
y2	0.7978	1.1910	382.76	<.0001

Figure 4.7. Univariate Diagnostic Checks

The VARMAX procedure provides diagnostic checks for the univariate form of the equations. The table in Figure 4.7 describes how well each univariate equation fits the data. From two univariate regression equations in Figure 4.3, the values of R^2 in the second column are 0.84 and 0.80 for each equation. The standard deviations in the third column are the square root of the diagonal elements of the covariance matrix from Figure 4.4. The F statistics are in the fourth column for hypotheses to test $\phi_{11} = \phi_{12} = 0$ and $\phi_{21} = \phi_{22} = 0$, respectively, where ϕ_{ij} is the (i,j)th 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.

The VARMAX Procedure

Univariate Model Diagnostic Checks

Variable	DW(1)	Normality ChiSq	Prob> ChiSq	ARCH1 F Value	Prob>F
y1	1.97	3.32	0.1900	0.13	0.7199
y2	2.14	5.46	0.0653	2.10	0.1503

Univariate Model Diagnostic Checks

Variable	AR1 F Value	AR1 Prob>F	AR1-2 F Value	AR1-2 Prob>F	AR1-3 F Value	AR1-3 Prob>F	AR1-4 F Value	AR1-4 Prob>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

Figure 4.8. Univariate Diagnostic Checks Continued

The check for white noise residuals in terms of the univariate equation is shown in Figure 4.8. 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; the third and fourth columns show the Jarque-Bera normality test statistics and their p-values; the last two columns show F statistics and their p-values for ARCH(1) disturbances. The second table includes F statistics and their p-values for AR(1) to AR(4) disturbances. The results indicate that the residuals are uncorrelated and homoscedastic.

The VARMAX Procedure

Forecasts

Variable	Obs	Time	Forecast	Standard Error	95% Confidence Limits	
y1	101	2000	-3.5921	1.1352	-5.8171	-1.3671
	102	2001	-3.0945	1.7091	-6.4443	0.2554
	103	2002	-2.1743	2.1447	-6.3779	2.0292
	104	2003	-1.1139	2.4317	-5.8799	3.6520
	105	2004	-0.1434	2.5874	-5.2146	4.9278
y2	101	2000	-2.0987	1.1910	-4.4330	0.2355
	102	2001	-2.7705	1.4767	-5.6647	0.1237
	103	2002	-2.7572	1.7421	-6.1717	0.6572
	104	2003	-2.2494	2.0193	-6.2071	1.7082
	105	2004	-1.4746	2.2517	-5.8878	2.9386

Figure 4.9. Forecasts

The table in Figure 4.9 gives forecasts and their prediction error covariances.

Bayesian Vector Autoregressive Process

The Bayesian Vector AutoRegressive (BVAR) model is used to avoid problems of collinearity and over-parameterization that often occur with the use of VAR models. BVAR models do this by imposing priors on the AR parameters.

The following statements fit a BVAR(1) model to the simulated data. You specify the PRIOR= option with the hyper-parameters. The LAMBDA=0.9 and THETA=0.1 are hyper-parameters controlling the prior covariance. Part of the VARMAX procedure output is shown in Figure 4.10.

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

```
                              The VARMAX Procedure

               Type of Model                              BVAR(1)
               Estimation Method    Method of Moments Estimation
               Prior LAMBDA                                   0.9
               Prior THETA                                    0.1


                           Model Parameter Estimates

Equation  Parameter      Estimate    Std Error   T Ratio  Prob>|T|  Variable

y1(t)     AR1_1_1         1.05623      0.05050     20.92    0.0001  y1(t-1)
          AR1_1_2        -0.34707      0.04824     -7.19    0.0001  y2(t-1)
y2(t)     AR1_2_1         0.40068      0.04889      8.20    0.0001  y1(t-1)
          AR1_2_2         0.48728      0.05740      8.49    0.0001  y2(t-1)


                      Covariance Matrix for the Innovation

                     Variable             y1             y2

                     y1              1.35807        0.44152
                     y2              0.44152        1.45070
```

Figure 4.10. Parameter Estimates for BVAR(1) Model

The output in Figure 4.10 shows that parameter estimates are slightly different from those in Figure 4.3. By choosing the appropriate priors, you may be able to get more accurate forecasts using a BVAR model rather than using an unconstrained VAR model.

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 over an unconstrained model.

The VECM(p) form 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 $\Delta \mathbf{y}_t = \mathbf{y}_t - \mathbf{y}_{t-1}$.

It has an equivalent VAR(p) representation as described in the preceding section.

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

Example of Vector Error Correction Model

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

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

This process can be given the following VECM(2) representation:

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

The following PROC IML statements generate simulated data for the VECM(2) form specified above and plot the data as shown in Figure 4.11:

```
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;
   t+1;
```

```
proc gplot data=simul2;
   symbol1 v = none i = join l = 1;
   symbol2 v = none i = join l = 2;
   plot y1 * t = 1
        y2 * t = 2 / overlay;
run;
```

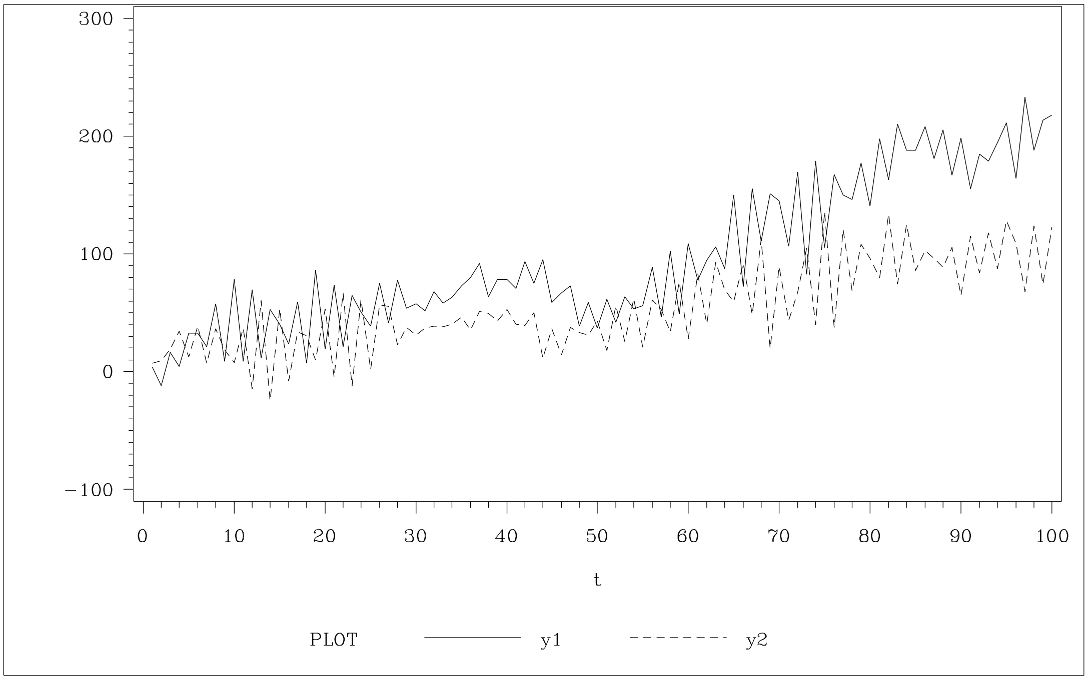

Figure 4.11. Plot of Generated Data Process

Cointegration Testing

The following statements use the Johansen cointegration rank test. The COINTTEST=(JOHANSEN) option does the Johansen trace test and is equivalent to the COINTTEST or COINTTEST=(JOHANSEN=(TYPE=TRACE)) option.

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

The VARMAX Procedure

Cointegration Rank Test

H_0: Rank=r	H_1: Rank>r	Eigenvalue	Trace	Critical Value	Drift InECM	DriftIn Process
0	0	0.5086	70.73	12.21	NOINT	Constant
1	1	0.0111	1.09	4.14		

Figure 4.12. Cointegration Rank Test

Figure 4.12 shows the output for the Johansen cointegration rank test. The last two columns explain the drift in the model or process. Since the NOINT option is specified, the model is

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

but the process has a constant drift before differencing.

H_0 is the null hypothesis and H_1 is the alternative hypothesis. The first row tests $r = 0$ against $r > 0$; 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. By default, the critical values at 5% significance level are used for testing. You can compare the test statistics and critical values in each row; there is one cointegrated process.

The following statements fit a VECM(2) form to the simulated data. From the result in Figure 4.12, the time series are cointegrated with rank=1. You specify the ECM= option with the RANK=1. For normalizing the value of the cointegrated vector, you specify the normalized variable with the NORMALIZE= option. The VARMAX procedure output is shown in Figure 4.13 and Figure 4.14.

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

```
                         The VARMAX Procedure

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


                         Long-Run Parameter
                           BETA Estimates
                           given RANK = 1

                       Variable        Dummy 1

                       y1              1.00000
                       y2             -1.95575


                       Adjustment Coefficient
                           ALPHA Estimates
                           given RANK = 1

                       Variable        Dummy 1

                       y1             -0.46680
                       y2              0.10667
```

Figure 4.13. Parameter Estimates for VECM(2) Form

The ECM= option produces the estimates of the long-run parameter, $\boldsymbol{\beta}$, and the adjustment coefficient, $\boldsymbol{\alpha}$. In Figure 4.13, "Dummy 1" indicates the first column of the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ matrices. Since the cointegration rank is 1 in the bivariate system, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$

are two-dimensional vectors. The estimated cointegrating vector is $\hat{\beta} = (1, -1.96)'$; the long-run relationship between y_{1t} and y_{2t} is $y_{1t} = 1.96 y_{2t}$. The first element of $\hat{\beta}$ is 1 since y_1 is specified as the normalized variable.

The VARMAX Procedure

Parameter ALPHA * BETA′ Estimates

Variable	y1	y2
y1	-0.46680	0.91295
y2	0.10667	-0.20862

AR Coefficient Estimates

DIF_Lag	Variable	y1	y2
1	y1	-0.74332	-0.74621
	y2	0.40493	-0.57157

Model Parameter Estimates

Equation	Parameter	Estimate	Std Error	T Ratio	Prob>|T|	Variable
D_y1(t)	AR1_1_1	-0.46680	0.04786	-9.75	0.0001	y1(t-1)
	AR1_1_2	0.91295	0.09359	9.75	0.0001	y2(t-1)
	AR2_1_1	-0.74332	0.04526	-16.42	0.0001	D_y1(t-1)
	AR2_1_2	-0.74621	0.04769	-15.65	0.0001	D_y2(t-1)
D_y2(t)	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	0.0001	D_y1(t-1)
	AR2_2_2	-0.57157	0.05128	-11.15	0.0001	D_y2(t-1)

Figure 4.14. Parameter Estimates for VECM(2) Form Continued

Figure 4.14 shows parameter estimates in terms of one lagged coefficient, $\mathbf{y}_{t-1}$, and one differenced lagged coefficient, $\Delta\mathbf{y}_{t-1}$, and their significances. "ALPHA * BETA′" indicates the coefficient of $\mathbf{y}_{t-1}$ and is obtained by multiplying the "ALPHA" and "BETA" estimates in Figure 4.13. AR1_i_j corresponds to the elements in the matrix "ALPHA * BETA′"; AR2_i_j corresponds to the elements in the matrix of AR coefficient estimates.

The fitted model is given as

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

A $D\_$ prefixed to a variable name implies differencing.

```
                              The VARMAX Procedure

                        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
```

Figure 4.15. Change VECM(2) Form to VAR(2) Model

The PRINT=(IARR) option in the previous SAS statements prints the reparamerized coefficient estimates. The VECM(2) form in Figure 4.15 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 β obtained from the VECM(p) form. Bayesian vector error correction models can improve forecast accuracy for cointegrated processes. The following statements fit a BVECM(2) form to the simulated data. You specify both the PRIOR= and ECM= options for the Bayesian vector error correction model. The VARMAX procedure output is shown in Figure 4.16.

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

```
                     The VARMAX Procedure

          Type of Model                             BVECM(2)
          Estimation Method    Method of Moments Estimation
          Cointegrated Rank                                1
          Prior LAMBDA                                   0.5
          Prior THETA                                    0.2


                   Adjustment Coefficient
                       ALPHA Estimates
                       given RANK = 1

                   Variable        Dummy 1

                   y1             -0.34392
                   y2              0.16659


              Parameter ALPHA * BETA' Estimates

            Variable             y1             y2

            y1             -0.34392        0.67262
            y2              0.16659       -0.32581


                   AR Coefficient Estimates

          DIF_Lag    Variable             y1             y2

                1    y1             -0.80070       -0.59320
                     y2              0.33417       -0.53480
```

Figure 4.16. Parameter Estimates for BVECM(2) Form

Figure 4.16 shows the model type fitted to the data, the estimates of the adjustment coefficient $\boldsymbol{\alpha}$, the parameter estimates in terms of one lagged coefficient, and one differenced lagged coefficient.

Vector Autoregressive Process 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

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

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;
   ... data lines omitted ...
   ;

proc varmax data=grunfeld;
   model y1-y3 = x1 x2 / p=1;
run;
```

The VARMAX procedure output is shown in Figure 4.17 through Figure 4.19.

```
                              The VARMAX Procedure

                        Number of Observations        20
                        Number of Pairwise Missing     0


Variable   Type    NoMissN         Mean       StdDev          Min          Max

y1         DEP          20    102.29000     48.58450     33.10000    189.60000
y2         DEP          20         1941    413.84329         1171         2803
y3         DEP          20    400.16000    250.61885     97.80000    888.90000
x1         INDEP        20     42.89150     19.11019     12.93000     90.08000
x2         INDEP        20    670.91000    222.39193    191.50000         1194

               Variable   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
```

Figure 4.17. Descriptive Statistics for VARX(1,0) Model

Figure 4.17 shows the descriptive statistics for the endogenous (dependent) and exogenous (independent) variables with labels.

The VARMAX Procedure

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

Constant Estimates

Variable	Constant
y1	-12.01279
y2	702.08673
y3	-22.42110

XLag Coefficient Estimates

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

Figure 4.18. Parameter Estimates for VARX(1,0) Model

Figure 4.18 shows parameter estimates for the constant and the lag zero exogenous variables.

The VARMAX Procedure

Model Parameter Estimates

Equation	Parameter	Estimate	Std Error	T Ratio	Prob>|T|	Variable
y1(t)	CONST1	-12.01279	27.47108	-0.44	0.6691	
	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(t)	CONST2	702.08673	256.48046	2.74	0.0169	
	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(t)	CONST3	-22.42110	10.31166	-2.17	0.0487	
	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)

Figure 4.19. Parameter Estimates for VARX(1,0) Model Continued

Figure 4.19 shows the parameter estimates and their significances.

The fitted model is given as

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

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

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 4.19, you can see the coefficients XL_0_1_2, AR_1_1_2, and AR_1_3_2 are insignificant. The following statements restrict the coefficients of $\theta_{12}^* = \phi_{12} = \phi_{32} = 0$ for the VARX(1,0) model.

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

```
                         The VARMAX Procedure

                      XLag Coefficient Estimates

          Lag    Variable            x1            x2

            0    y1             1.67592             0
                 y2            -6.30880       2.65308
                 y3            -0.03576      -0.00919


                       AR Coefficient Estimates

   Lag    Variable            y1            y2            y3

     1    y1             0.27671             0       0.01747
          y2            -2.16968       0.10945      -0.93053
          y3             0.96398             0       0.93412
```

Figure 4.20. Parameter Estimation on Restrictions

The output in Figure 4.20 shows that three parameters θ^*_{12}, ϕ_{12}, and ϕ_{32} are replaced by the restricted values.

```
                         The VARMAX Procedure

                         Restriction Results

                   Lagrange
  Parameter      Multiplier      Std Error     T Ratio    Prob>|T|

  XL0_1_2           1.74969       21.44026        0.08      0.9389
  AR1_1_2          30.36254       70.74347        0.43      0.6899
  AR1_3_2          55.42191      164.03075        0.34      0.7524
```

Figure 4.21. RESTRICT Statement Results

The output in Figure 4.21 shows the estimates of the Lagrangian parameters and their significances. You cannot reject the null hypotheses $\theta^*_{12} = 0$, $\phi_{12} = 0$, and $\phi_{32} = 0$ with the 0.05 significance level.

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 VARMAX Procedure

                                  Test Results

                                 Chi-              Prob>
                   Test         Square      DF     Chisq

                      1         150.31       1    <.0001
                      2           0.34       3    0.9522
```

Figure 4.22. TEST Statement Results

The output in Figure 4.22 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; you cannot reject the joint hypothesis test $\theta^*_{12} = \phi_{12} = \phi_{32} = 0$ at the 0.05 significance level.

Causality Testing

The following statements use the CAUSAL statement to compute the Granger-Causality test for the VAR(1) model:

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

```
                              The VARMAX Procedure

                          Granger Causality Wald Test

                                 Chi-              Prob>
                   Test         Square      DF     Chisq

                      1           2.95       2    0.2291
                      2           0.89       2    0.6420


                    Test 1:   Group 1 Variables:   y1 y3
                              Group 2 Variables:   x2


                    Test 2:   Group 1 Variables:   y1 y3
                              Group 2 Variables:   y2
```

Figure 4.23. CAUSAL Statement Results

The output in Figure 4.23 is associated with the CAUSAL statement. The first column in the output is the index corresponding to each CAUSAL statement. The output shows that you cannot reject Granger-noncausality from $x2$ to $(y1, y3)$ using a 0.05 significance level; you cannot reject Granger-noncausality from $y2$ to $(y1, y3)$.

Syntax

PROC VARMAX *options* **;**
BY *variables* **;**
CAUSAL group1 *= (variables)* **group2** *= (variables)* **;**
COINTEG rank *= number < options >* **;**
ID *variable* **interval=** *value < option >* **;**
MODEL *dependent variables < = regressors >*
<, dependent variables < = regressors > ... >
</ options > **;**
OUTPUT *< options >* **;**
RESTRICT *restrictions* **;**
TEST *restrictions* **;**

Functional Summary

The statements and options used with the VARMAX procedure are summarized in the following table:

Description	Statement	Option
Data Set Options		
specify the input data set	VARMAX	DATA=
write parameter estimates to an output data set	VARMAX	OUTEST=
include covariances in the OUTEST= data set	VARMAX	OUTCOV
write the diagnostic checking tests for a model and the cointegration test results to an output data set	VARMAX	OUTSTAT=
write actuals, predictions, residuals, and confidence limits to an output data set	OUTPUT	OUT=
BY Groups		
specify BY-group processing	BY	
ID Variable		
specify identifying variable	ID	
specify the time interval between observations	ID	INTERVAL=
control the alignment of SAS Date values	ID	ALIGN=
Printing Control Options		
specify how many lags to print results	MODEL	LAGMAX=
suppress the printed output	MODEL	NOPRINT
request all printing options	MODEL	PRINTALL
request the printing format	MODEL	PRINTFORM=
PRINT= Option		
print the correlation matrix of parameter estimates	MODEL	CORRB

Description	Statement	Option
print the cross-correlation matrices of independent variables	MODEL	CORRX
print the cross-correlation of dependent variables	MODEL	CORRY
print the covariance matrices of prediction errors	MODEL	COVPE
print the cross-covariance matrices of the independent variables	MODEL	COVX
print the cross-covariance matrices of the dependent variables	MODEL	COVY
print the covariance matrix of parameter estimates	MODEL	COVB
print the decomposition of the prediction error covariance matrix	MODEL	DECOMPOSE
print the infinite order AR representation	MODEL	IARR
print the impulse response function	MODEL	IMPULSE=
print the impulse response function in the transfer function	MODEL	IMPULSX=
print the partial autoregressive coefficient matrices	MODEL	PARCOEF
print the partial canonical correlation matrices	MODEL	PCANCORR
print the partial correlation matrices	MODEL	PCORR
print the eigenvalues of the companion matrix	MODEL	ROOTS
print the Yule-Walker estimates	MODEL	YW
Model Estimation and Order Selection Options		
center the dependent variables	MODEL	CENTER
specify the degrees of differencing for the specified model variables	MODEL	DIF=
specify the degrees of differencing for all independent variables	MODEL	DIFX=
specify the degrees of differencing for all dependent variables	MODEL	DIFY=
specify the vector error correction model	MODEL	ECM=
specify the estimation method	MODEL	METHOD=
select the tentative order	MODEL	MINIC=
suppress the current values of independent variables	MODEL	NOCURRENTX
suppress the intercept parameters	MODEL	NOINT
specify the number of seasonal periods	MODEL	NSEASON=
specify the order of autoregressive polynomial	MODEL	P=
specify the Bayesian prior model	MODEL	PRIOR=
center the seasonal dummies	MODEL	SCENTER
specify the degree of time trend polynomial	MODEL	TREND=
specify the denominator for error covariance matrix estimates	MODEL	VARDEF=
specify the lag order of independent variables	MODEL	XLAG=

Description	Statement	Option
Cointegration Related Options		
print the results from the weak exogeneity test of the long-run parameters	COINTEG	EXOGENEITY
specify the restriction on the cointegrated coefficient matrix	COINTEG	H=
specify the restriction on the adjustment coefficient matrix	COINTEG	J=
specify the variable name whose cointegrating vectors are normalized	COINTEG	NORMALIZE=
specify a cointegration rank	COINTEG	RANK=
print the Johansen cointegration rank test	MODEL	COINTTEST= (JOHANSEN=)
print the Stock-Watson common trends test	MODEL	COINTTEST=(SW=)
print the Dickey-Fuller unit root test	MODEL	DFTEST=
Tests and Restrictions on Parameters		
test the Granger Causality	CAUSAL	GROUP1= GROUP2=
place and test restrictions on parameter estimates	RESTRICT	
test hypotheses	TEST	
Output Control Options		
specify the size of confidence limits for forecasting	OUTPUT	ALPHA=
start forecasting before end of the input data	OUTPUT	BACK=
specify how many periods to forecast	OUTPUT	LEAD=
suppress the printed forecasts	OUTPUT	NOPRINT

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 JOHANSEN= option is specified, the results of this option are also written to the output data set.

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=, SCENTER, TREND=, VARDEF=, and XLAG=.

BY Statement

BY *variables***;**

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

CAUSAL Statement

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

A CAUSAL statement prints the Granger-Causality test by fitting the VAR(p) model. 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 $p = 0$ is in the MODEL statement, the CAUSAL statement is not applicable.

See the “VAR Modeling” section on page 101 for details.

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 / p=1;
   causal group1=(y1 y3) group2=(y2);
   causal group1=(y2) group2=(y1 y3);
run;
```

COINTEG Statement

> **COINTEG RANK=** *number* < **H=** *(matrix)* > < **J=** *(matrix)* >
> < **EXOGENEITY** > < **NORMALIZE=** *variable* > **;**

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 the weak exogeneity in the long-run parameters. The cointegrated system uses the maximum likelihood analysis proposed by Johansen and Juselius (1990) and Johansen (1995a, 1995b). Only one COINTEG statement is allowed.

The VECM(p) form is written as

$$\Delta \mathbf{y}_t = \boldsymbol{\alpha}\boldsymbol{\beta}'\mathbf{y}_{t-1} + \sum_{i=1}^{p-1} \Phi_i^* \Delta \mathbf{y}_{t-i} + \boldsymbol{\delta} + \boldsymbol{\epsilon}_t$$

where $\boldsymbol{\alpha}$ is a $k \times r$ matrix called adjustment coefficient, $r \leq k$; $\boldsymbol{\beta}$ is a $k \times r$ matrix called the long-run coefficient; Δ means $\Delta \mathbf{y}_t = \mathbf{y}_t - \mathbf{y}_{t-1}$.

The following statements are examples of the COINTEG statement and the two examples are equivalent. You specify the ECM= option for fitting the VECM(p) with the P= option in the MODEL statement or the COINTEG statement. For testing of the restrictions of either $\boldsymbol{\alpha}$ or $\boldsymbol{\beta}$ or both, you specify either J= or H= or both. You specify the EXOGENEITY option for tests of the weak exogeneity in the long-run parameters.

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

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

EXOGENEITY

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

H= (*matrix***)**

specifies the restrictions H on the $k \times r$ or $(k+1) \times r$ cointegrated coefficient matrix $\boldsymbol{\beta}$ such that $\boldsymbol{\beta} = H\phi$, where H is known and ϕ is unknown. The $k \times m$ or $(k+1) \times m$ matrix H is specified using this option, where k is the number of endogenous variables, and m is $r \leq m < k$ with RANK=r. For example, consider that the system contains four variables and RANK=1 with $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3, \beta_4)'$. The restriction matrix for the test of $\beta_1 + \beta_2 = 0$ can be specified as

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

When the data have no deterministic trend, the constant term should be restricted by $\boldsymbol{\alpha}'_{\perp}\boldsymbol{\delta} = 0$. You can specify the restriction matrix for the previous test as follows:

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

When the cointegrated system contains three endogenous variables and RANK=2, you can specify the restriction matrix for the test of $\beta_{1j} = -\beta_{2j}$ for $j = 1, 2$ as follows:

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

J= (*matrix***)**
specifies the restrictions J on the $k \times r$ adjustment matrix $\boldsymbol{\alpha}$ such that $\boldsymbol{\alpha} = J\psi$, where J is known and ψ is unknown. The $k \times m$ matrix J is specified using this option, where k is the number of endogenous variables, and m is $r \leq m < k$ with RANK=r. For example, when the system contains four variables and RANK=1, you can specify the restriction matrix for the test of $\alpha_j = 0$ for $j = 2, 3, 4$ as follows:

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

When the system contains three variables and RANK=2, you can specify the restriction matrix for the test of $\alpha_{2j} = 0$ for $j = 1, 2$ as follows:

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

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

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

ID Statement

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

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

MODEL Statement

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

The MODEL statement specifies endogenous (dependent) variables and exogenous (independent) 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 with independent variables. The one MODEL statement is required.

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 for the vectors $(y1, y2, y3, y4)$ and $(x1, x2, x3, x4, x5)$:

```
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 corresponding to 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 following options can be used in the MODEL statement after a forward slash (/):

General Options

CENTER

centers endogenous (dependent) 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.

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

specifies the degrees of differencing to be applied to the specified dependent or independent variables. The differencing can be the same for all variables, or it can vary among variables. For example, DIF=(y_1(1,4) y_3(1) x_2(2)) specifies that the y_1 series is differenced at lag 1 and at lag 4, which is $(y_{1t} - y_{1,t-1}) - (y_{1,t-4} - y_{1,t-5})$; y_3 at lag 1, which is $(y_{3t} - y_{3,t-1})$; x_2 at lag 2, which is $(x_{2t} - x_{2,t-2})$.

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

specifies the degrees of differencing to be applied to all exogenous (independent) variables. For example, DIFX=(1) specifies that the series are differenced once at lag 1; DIFX=(1,4) at lag 1 and at lag 4. If exogenous variables are specified in the DIF= option, this option is ignored.

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

specifies the degrees of differencing to be applied to all endogenous (dependent) variables. For details, see the DIFX= option. If endogenous variables are specified in the DIF= option, this option is ignored.

METHOD= *value*

requests the type of estimates to be computed. The possible values of the METHOD= option are

LS	specifies least-squares estimates.
ML	specifies maximum likelihood estimates.

If the PRIOR= or ECM= option or both is specified, the default is METHOD=ML; otherwise, the default is METHOD=ML.

NOCURRENTX

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

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

If this option is specified, it suppresses the current values $\mathbf{x}_t$ and starts with $\mathbf{x}_{t-1}$.

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

NOINT

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

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.

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.

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 when the ECM= option is specified.

VARDEF= *value*

corrects for the degrees of freedom of the denominator. This option is used to calculate an error covariance matrix for the METHOD=LS. If the METHOD=ML is specified, the VARDEF=N is 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.

Printing Control Options

LAGMAX= *number*

specifies the lag to compute and display the results obtained by the PRINT=(CORRX CORRY COVX COVY IARR IMPULSE= IMPULSX= PARCOEF PCANCORR

PCORR) option. This option is also used to print cross-covariances and cross-correlations of residuals. The default is LAGMAX=min(12, T-2), where T is the number of nonmissing observations.

NOPRINT
suppresses all printed output.

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

You can also specify this option as ALL.

PRINTFORM= *value*
requests the printing format of outputs of the PRINT= option and cross-covariances and cross-correlations of residuals. Valid values are as follows:

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

Printing Options

PRINT=(*options***)**
The following options can be used in the PRINT=( ) option. The options are listed within parentheses.

CORRB
prints the estimated correlations of the parameter estimates.

CORRX
prints the cross-correlation matrices of exogenous (independent) variables using the number of lags specified by the LAGMAX=*number*.

CORRY
prints the cross-correlation matrices of endogenous (dependent) variables using the number of lags specified by the LAGMAX=*number*.

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. If the DIF= or DIFY= option is specified, the covariance matrices of multistep-ahead prediction errors are computed based on the differenced data. This option is not applicable when the PRIOR= option is specified. See the “Forecasting” section on page 87 for details.

COVX

prints the cross-covariance matrices of exogenous (independent) variables using the number of lags specified by the LAGMAX=*number*.

COVY

prints the cross-covariance matrices of endogenous (dependent) variables using the number of lags specified by the LAGMAX=*number*.

DECOMPOSE
DECOMPOSE(*number*)

prints the decomposition of the prediction error covariances using the number of lags specified by *number* in parentheses for the VARMAX(p,q,s) model. It can be interpreted as the contribution of innovations in one variable to the mean squared error of the multistep-ahead 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-ahead prediction errors are computed based on the differenced data. This option is not applicable when the PRIOR= option is specified. See the "Forecasting" section on page 87 for details.

IARR

prints the infinite order AR representation of a VARMA process. The coefficient matrices print the number of lags specified by the LAGMAX=*number*. If the moving-average order is zero, this option prints the maximum order of the autoregressive characteristic function. If the ECM= option is specified, the reparameterized AR coefficient matrices are printed.

IMPULSE
IMPULSE= (SIMPLE ACCUM ORTH STDERR ALL)

prints the impulse response function using the number of lags specified by the LAGMAX= *number*. 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. See the "Forecasting" section on page 87 for details.

The following options can be used in the IMPULSE=( ) option. The options are listed within parentheses.

ACCUM	prints the accumulated impulse function.
ALL	equivalent to specifying all of SIMPLE, ACCUM, ORTH, and STDERR.
ORTH	prints the orthogonalized impulse 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. If the exogenous variables are used to fit the model, this option is ignored.

IMPULSX
IMPULSX= (SIMPLE ACCUM ALL)

prints the impulse response function related to exogenous (independent) variables using the number of lags specified by the LAGMAX=*number*. See the "Forecasting" section on page 87 for details.

The following options can be used in the IMPULSX=( ) option. The options are listed within parentheses.

ACCUM	prints the accumulated impulse response matrices in the transfer function.
ALL	equivalent to specifying both SIMPLE and ACCUM.
SIMPLE	prints the impulse response matrices in the transfer function. This is the default.

PARCOEF

prints the partial autoregression coefficient matrices, Φ_{mm}. With a VAR process, this option is useful for the identification of the order since the Φ_{mm} have the characteristic property that they equal zero for $m > p$ under the hypothetical assumption of a VAR(p) model. These matrices print the number of lags specified by the LAGMAX=*number*. See the "Tentative Order Selection" section on page 96 for details.

PCANCORR

prints the partial canonical correlations of the process at lag m and the test for testing Φ_m=0 for $m > p$. 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 $\mathbf{y}_{t-1}, \ldots, \mathbf{y}_{t-m+1}$. See the "Tentative Order Selection" section on page 96 for details.

PCORR

prints the partial correlation matrices using the number of lags specified by the LAGMAX=*number*. 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 PARCOEF option. See the "Tentative Order Selection" section on page 96 for details.

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 endogenous (dependent) 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.

YW

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

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 as, for example, P=(1,3,4). P=3 is equivalent to P=(1,2,3). The default is P=0.

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

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

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

specifies the lags of exogenous (independent) variables. Subset models of distributed lags can be specified as, for example, XLAG=(2). The default is XLAG=0. To exclude the present values of exogenous (independent) variables from the model, the NOCURRENTX option must be used.

Tentative Order Selection Options

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

prints the information criterion for the appropriate AR and MA tentative order selection and for the diagnostic checks of the fitted model.

If the MINIC= option is not specified, all types of information criteria are printed for diagnostic checks of the fitted model.

The following options can be used in the MINIC=( ) option. The options are listed within parentheses.

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

specifies the range of AR orders. The default is P=(0:5).

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. The default is Q=(0:5).

TYPE= *value*

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

AIC	specifies the Akaike Information Criterion.
AICC	specifies the Corrected Akaike Information Criterion. This is the default 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.

Cointegration Related Options

COINTTEST

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

The following options can be used in the COINTTEST=( ) option. The options are listed within parentheses.

JOHANSEN

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

prints the cointegration rank test for multivariate time series based on Johansen method. This test is provided when the number of endogenous (dependent) variables is less than or equal to 11. See the "Vector Error Correction Modeling" section on page 113 for details.

The VAR(p) model can be written as the error correction model

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

where Π, Φ_i^*, and A are coefficient parameters; 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

$$\Delta^2 \mathbf{y}_t = \Pi \mathbf{y}_{t-1} - \Psi \Delta \mathbf{y}_{t-1} + \sum_{i=1}^{p-2} \Psi_i \Delta^2 \mathbf{y}_{t-i} + AD_t + \boldsymbol{\epsilon}_t$$

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 $\boldsymbol{\beta}'\boldsymbol{\beta}_{\perp} = 0$.

The following options can be used in the JOHANSEN=( ) option. The options are listed within parentheses.

IORDER= *number* specifies the integrated order.

IORDER=1	prints the cointegration rank test for an integrated order 1 and prints the long-run parameter, β, and the adjustment coefficient, α. This is the default. If IORDER=1 is specified, the AR order should be greater than or equal to 1. When P=0, P is temporarily set to 1.
IORDER=2	prints the cointegration rank test for integrated orders 1 and 2. If IORDER=2 is specified, the AR order should be greater than or equal to 2. If P=1, the IORDER=1 is used; if P=0, P is temporarily set to 2.

NORMALIZE= *variable* specifies the endogenous (dependent) variable name whose cointegration vectors are to be normalized. If the normalized variable is different from that specified in the ECM= option or the COINTEG statement, the latter is used.

TYPE= *value* specifies the type of cointegration rank test to be printed. Valid values are as follows:

MAX	prints the cointegration maximum eigenvalue test.
TRACE	prints the cointegration trace test. This is the default.

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

SIGLEVEL= *value*

sets the size of cointegration rank tests and common trends tests. The SIGLEVEL=*value* option must be one of 0.1, 0.05, or 0.01. The default is SIGLEVEL=0.05.

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 endogenous (dependent) variables is less than or equal to 6. See the “Common Trends” section on page 111 for details.

The following options can be used in the SW=( ) option. The options are listed within parentheses.

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

TYPE= *value* specifies the type of common trends test to be printed. Valid values are as follows:

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

DFTEST
DFTEST= (DLAG=*number*)

prints the Dickey-Fuller unit root test. The DLAG=*number* specifies the regular or seasonal unit root test. If the *number* is greater than one, seasonal Dickey-Fuller tests are performed. The *number* should be one of 1, 2, 4, or 12. The default is DLAG=1.

Bayesian VAR Estimation Options

PRIOR
PRIOR= (MEAN=(*vector*) LAMBDA=*value* THETA=*value* IVAR<=(*variables*)> NREP=*number* SEED=*number*)

specifies the prior value of parameters for the BVAR(p) model. If the ECM= option is specified with the PRIOR option, the BVECM(p) form is fitted. The following options can be used in the PRIOR option. For the standard errors of the predictors, the bootstrap procedure is used. See the "Bayesian VAR Modeling" section on page 108 for details.

The following options can be used in the PRIOR=( ) option. The options are listed within parentheses.

IVAR
IVAR= (*variables*)

specifies an integrated BVAR(p) model. If you use the IVAR option without *variables*, 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 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= is larger, a BVAR(p) model is close 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. See the "Bayesian VAR Modeling" section on page 108 for details.

You can specify the mean vector by order of the equation. Let $B = (\delta, \Phi_1, ..., \Phi_p)'$ be the parameter sets to be estimated. If $\Phi = (\Phi_1, ..., \Phi_p)'$, then MEAN= ($\mathrm{vec}(\Phi)$).

For example, in case of a bivariate ($k = 2$) BVAR(2) model,

$$\mathrm{MEAN} = (\phi_{1,11}\ \phi_{1,12}\ \phi_{2,11}\ \phi_{2,12}\ \phi_{1,21}\ \phi_{1,22}\ \phi_{2,21}\ \phi_{2,22})$$

where $\phi_{l,ij}$ is the (i,j)th element of the matrix Φ_l.

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

For a Bayesian error correction model, you specify a mean vector for only lagged AR coefficients, Φ_i^*, in terms of regressors $\Delta \mathbf{y}_{t-i}$, for $i = 1, ..., (p-1)$ 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.

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

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

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

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

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

NREP= *number*

specifies the number of bootstrap replications. The default is NREP=100.

SEED= *number*

specifies seeds to generate uniform random numbers for resampling. By default, the system clock is used to generate the random seed.

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= is close to 1, a BVAR(p) model is close to a VAR(p) model.

Vector Error Correction Model Options

ECM=(RANK=*number* **NORMALIZE=***variable***)**
specifies a vector error correction model.

The following options can be used in the ECM=( ) option. The options are listed within parentheses.

NORMALIZE= *variable*
specifies a single endogenous variable name whose cointegrating vectors are normalized. If the variable name is different from that specified in the COINTEG statement, the latter is used.

RANK= *number*
specifies the cointegration rank. This option is required in the ECM= option. The value of the RANK= option should be greater than zero and less than or equal to the number of endogenous (dependent) variables, k. If the rank is different from that specified in the COINTEG statement, the latter is used.

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.

ALPHA= *value*
sets the forecast confidence limits. The ALPHA=*value* must be between 0 and 1. When you specify ALPHA=α, the upper and lower confidence limits define the $1-\alpha$ 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-ahead forecasts are to 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 forecast starts at the end of the available data.

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

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

OUT= *SAS-data-set*
writes the forecast values to an output data set.

RESTRICT Statement

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

The RESTRICT statement restricts the specified parameters to the specified values. Only one RESTRICT statement is allowed.

The *restriction*'s form is *parameter = value* and each restriction is separated by commas. Parameters are referred by the following keywords:

- CONST(i), the intercept parameter of the current value ith time series y_{it}
- AR(l,i,j), the autoregressive parameter of the previous lag l value of the jth endogenous (dependent) variable, $y_{j,t-l}$, to the ith endogenous variable at time t, y_{it}
- XL(l,i,j), the exogenous parameter of the previous lag l value of the jth exogenous (independent) variable, $x_{j,t-l}$, to the ith endogenous variable at time t, y_{it}
- SDUMMY(i,j), the jth seasonal dummy of the ith time series at time t, y_{it}, where $j = 1, \ldots, (nseason - 1)$

To use the RESTRICT statement, you need to know the form of the model. If you do not specify any order of the model, the RESTRICT statement is not applicable.

Restricted parameter estimates are computed by introducing a Lagrangian parameter for each restriction (Pringle and Raynor 1971). The Lagrangian parameter measures the sensitivity of the sum of square errors to the restriction. The estimates of these Lagrangian parameters and their significances are printed in the restriction results table.

The following are examples of the RESTRICT statement. The first 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 following shows a bivariate (k=2) VARX(1,1) model with three exogenous variables,

```
proc varmax data=two;
   model y1 = x1 x2, y2 = x2 x3 / p=1 xlag=1;
   restrict XL(0,1,1)=-1.2, XL(1,2,3)=0;
run;
```

TEST Statement

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

The TEST statement performs the Wald test for the joint hypothesis specified in the statement. The *restriction*'s form is *parameter = value* and each restriction is separated by commas. The *restriction*'s form is referred to by the same rule in the RESTRICT statement. Any number of TEST statements can be specified.

To use the TEST statement, you need to know the form of the model. If you do not specify any order of the model, the TEST statement is not applicable.

See the "Granger-Causality Test" section on page 104 for the Wald test.

The following is an example of the TEST statement. In case of 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;
```

Details

Missing Values

The VARMAX procedure currently does not support missing values. The procedure uses the first contiguous group of observations with no missing values for any of the MODEL statement variables. Observations at the beginning of the data set with 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 endogenous (dependent) variables with missing values and exogenous (independent) variables with nonmissing values.

VARMAX Modeling

The vector autoregressive and 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 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

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

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 B, 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, $\mathbf{x}_t$, can be stochastic or nonstochastic. When the exogenous variables, $\mathbf{x}_t$, are stochastic and their future values are unknown, then forecasts of these future values are needed in the forecasting of the future values of the $\mathbf{y}_t$. On occasion, future values of the $\mathbf{x}_t$ can be assumed to be known because they are deterministic variables. Note that the VARMAX procedure assumes that the exogenous variables, $\mathbf{x}_t$, 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.

Dynamic Simultaneous Equations Modeling

In the econometrics literature, the VARMAX(p,q,s) model could be written in the following slightly different form, which is referred to as a *dynamic simultaneous equations* model or a *dynamic structural equations* model.

Since $\mathrm{E}(\boldsymbol{\epsilon}_t\boldsymbol{\epsilon}_t') = \Sigma$ is assumed to be positive-definite, there exists a lower triangular matrix A_0 with ones on the diagonals such that $A_0\Sigma A_0' = \Sigma^d$, where Σ^d is a diagonal matrix with positive diagonal elements.

$$A_0\mathbf{y}_t - \sum_{i=1}^{p} A_i\mathbf{y}_{t-i} = \sum_{i=0}^{s} C_i^*\mathbf{x}_{t-i} + C_0\boldsymbol{\epsilon}_t - \sum_{i=1}^{q} C_i\boldsymbol{\epsilon}_{t-i}$$

where $A_i = A_0\Phi_i$, $C_i^* = A_0\Theta_i^*$, $C_0 = A_0$, $C_i = A_0\Theta_i$. and $C_0\boldsymbol{\epsilon}_t$ has a diagonal covariance matrix Σ^d.

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.

State-Space Modeling

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

$$\mathbf{z}_t = \begin{bmatrix} \mathbf{y}_t \\ \vdots \\ \mathbf{y}_{t-p+1} \\ \mathbf{x}_t \\ \vdots \\ \mathbf{x}_{t-s+1} \\ \boldsymbol{\epsilon}_t \\ \vdots \\ \boldsymbol{\epsilon}_{t-q+1} \end{bmatrix}, \ K = \begin{bmatrix} \Theta_0^* \\ 0_{k\times r} \\ \vdots \\ 0_{k\times r} \\ I_r \\ 0_{r\times r} \\ \vdots \\ 0_{r\times r} \\ 0_{k\times r} \\ \vdots \\ 0_{k\times r} \end{bmatrix}, \ G = \begin{bmatrix} I_k \\ 0_{k\times k} \\ \vdots \\ 0_{k\times k} \\ 0_{r\times k} \\ \vdots \\ 0_{r\times k} \\ I_{k\times k} \\ 0_{k\times k} \\ \vdots \\ 0_{k\times k} \end{bmatrix}$$

$$F = \begin{bmatrix}
\Phi_1 & \cdots & \Phi_{p-1} & \Phi_p & \Theta_1^* & \cdots & \Theta_{s-1}^* & \Theta_s^* & -\Theta_1 & \cdots & -\Theta_{q-1} & -\Theta_q \\
I_k & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \cdots & I_k & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & \cdots & 0 & 0 & I_r & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \cdots & 0 & 0 & 0 & \cdots & I_r & 0 & 0 & \cdots & 0 & 0 \\
0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & I_k & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & I_k & 0
\end{bmatrix}$$

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

$$\mathbf{x}_t = \sum_{i=1}^{p} A_i\mathbf{x}_{t-i} + \mathbf{a}_t - \sum_{i=1}^{q} C_i\mathbf{a}_{t-i}$$

or $A(B)\mathbf{x}_t = C(B)\mathbf{a}_t$, where $A(B) = I_r - A_1B - \cdots - A_pB^p$ and $C(B) = I_r - C_1B - \cdots - C_qB^q$ 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 (such as stationarity) conditions, $\mathbf{x}_t$ 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, MMSE) 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 (MMSE) 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

$$\begin{aligned}
\mathbf{y}_{t+v|t} &= \sum_{j=1}^{p} \Phi_j \mathbf{y}_{t+v-j|t} + \sum_{j=1}^{u} \Pi_j \mathbf{x}_{t+v-j|t} \ \text{ for } \ u \le v \\
\mathbf{y}_{t+v|t} &= \sum_{j=1}^{p} \Phi_j \mathbf{y}_{t+v-j|t} + \sum_{j=1}^{r} \Pi_j \mathbf{x}_{t+v-j|t} \ \text{ for } \ u > v
\end{aligned}$$

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

$$\mathbf{z}_t = \begin{bmatrix} \mathbf{y}_t \\ \mathbf{y}_{t+1|t} \\ \vdots \\ \mathbf{y}_{t+v-1|t} \\ \mathbf{x}_t \\ \mathbf{x}_{t+1|t} \\ \vdots \\ \mathbf{x}_{t+v-1|t} \end{bmatrix}, \ \mathbf{x}_t^* = \begin{bmatrix} \mathbf{x}_{t+v-u} \\ \mathbf{x}_{t+v-u+1} \\ \vdots \\ \mathbf{x}_{t-1} \end{bmatrix}, \ \mathbf{e}_{t+1} = \begin{bmatrix} \mathbf{a}_{t+1} \\ \boldsymbol{\epsilon}_{t+1} \end{bmatrix}$$

$$F = \begin{bmatrix} 0 & I_k & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & I_k & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \Phi_v & \Phi_{v-1} & \Phi_{v-2} & \cdots & \Phi_1 & \Pi_v & \Pi_{v-1} & \Pi_{v-2} & \cdots & \Pi_1 \\ 0 & 0 & 0 & \cdots & 0 & 0 & I_r & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & I_r & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & A_v & A_{v-1} & A_{v-2} & \cdots & A_1 \end{bmatrix}$$

$$K = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \Pi_u & \Pi_{u-1} & \cdots & \Pi_{v+1} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \quad G = \begin{bmatrix} V_0 & I_k \\ V_1 & \Psi_1 \\ \vdots & \vdots \\ V_{v-1} & \Psi_{v-1} \\ I_r & 0_{r\times k} \\ \Psi_1^x & 0_{r\times k} \\ \vdots & \vdots \\ \Psi_{v-1}^x & 0_{r\times k} \end{bmatrix}$$

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 $u > v$.

Forecasting

The optimal (MMSE) l-step-ahead forecast of $\mathbf{y}_{t+l}$ is

$$\begin{aligned} \mathbf{y}_{t+l|t} &= \sum_{j=1}^{p} \Phi_j \mathbf{y}_{t+l-j|t} + \sum_{j=0}^{s} \Theta_j^* \mathbf{x}_{t+l-j|t} - \sum_{j=1}^{q} \Theta_j \boldsymbol{\epsilon}_{t+l-j}, \;\; l \le q \\ \mathbf{y}_{t+l|t} &= \sum_{j=1}^{p} \Phi_j \mathbf{y}_{t+l-j|t} + \sum_{j=0}^{s} \Theta_j^* \mathbf{x}_{t+l-j|t}, \;\; l > q \end{aligned}$$

with $\mathbf{y}_{t+l-j|t} = \mathbf{y}_{t+l-j}$ and $\mathbf{x}_{t+l-j|t} = \mathbf{x}_{t+l-j}$ for $l \le j$. For the forecasts $\mathbf{x}_{t+l-j|t}$, see the previous section.

Impulse Response Function

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 impact that a shock in one variable has on another variable. Let $\psi_{j,in}$ be the *element* of the Ψ_j. The notation i is the index for the impulse variable, and n is the index for the response variable (impulse $\rightarrow$ response). For instance, $\psi_{j,11}$ is an impulse response to $y_{1t} \rightarrow y_{1t}$, and $\psi_{j,12}$ is an impulse response to $y_{1t} \rightarrow y_{2t}$.

The accumulated impulse response function is the cumulative sum of the impulse response function, $\Psi^a_l = \sum_{j=0}^{l} \Psi_j$.

The MA representation with a standardized white noise innovation process offers a further possibility 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 is written as

$$\mathbf{y}_t = \Psi^o(B)\mathbf{u}_t$$

where $\Psi^o(B) = \sum_{j=0}^{\infty} \Psi^o_j B^j$, $\Psi^o_j = \Psi_j P$, and $\mathbf{u}_t = P^{-1}\boldsymbol{\epsilon}_t$.

The elements of the matrices Ψ^o_j, 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 the lag j.

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 the lag j, and is called an impulse response matrix in the transfer function.

The accumulated impulse response in the transfer function is the cumulative sum of the impulse response in the transfer function, $\Psi^{*a}_l = \sum_{j=0}^{l} \Psi^*_j$.

The asymptotic distributions of the impulse functions can be seen in the "VAR Modeling" section on page 101.

The following statements provide the impulse response and the accumulated impulse response in the transfer function for a VARX(1,0) model. Parts of the VARMAX procedure output are shown in Figure 4.24 and Figure 4.25.

```
proc varmax data=grunfeld;
   model y1-y3 = x1 x2 / p=1 print=(impulsx=(all)) lagmax=15
                          printform=univariate;
run;
```

The VARMAX Procedure

Impulse Response Matrices in
Transfer Function by Variable

Variable	Lead	x1	x2
y1	0	1.69281	-0.00859
	1	0.35399	0.01727
	2	0.09090	0.00714
	:	:	:
	14	0.03319	0.00024164
	15	0.03195	0.00023260
y2	0	-6.09850	2.57980
	1	-5.15484	0.45445
	2	-3.04168	0.04391
	:	:	:
	14	-1.32641	-0.00966
	15	-1.27682	-0.00930
y3	0	-0.02317	-0.01274
	1	1.57476	-0.01435
	2	1.80231	0.00398
	:	:	:
	14	1.16268	0.00846
	15	1.11921	0.00815

Figure 4.24. Impulse Response in Transfer Function (IMPULSX= option)

The VARMAX Procedure

Accumulated Impulse Response Matrices
in Transfer Function by Variable

Variable	Lead	x1	x2
y1	0	1.69281	-0.00859
	1	2.04680	0.00868
	2	2.13770	0.01582
	:	:	:
	14	2.63183	0.02162
	15	2.66378	0.02185
y2	0	-6.09850	2.57980
	1	-11.25334	3.03425
	2	-14.29502	3.07816
	:	:	:
	14	-34.35946	2.93139
	15	-35.63628	2.92210
y3	0	-0.02317	-0.01274
	1	1.55159	-0.02709
	2	3.35390	-0.02311
	:	:	:
	14	20.71402	0.10051
	15	21.83323	0.10866

Figure 4.25. Accumulated Impulse Response in Transfer Function (IMPULSX= option)

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 4.26 through Figure 4.28.

```
proc varmax data=simul1;
   model y1 y2 / p=1 noint lagmax=15 print=(impulse=(all))
                 printform=univariate;
run;
```

The VARMAX Procedure

Impulse Response by Variable

Variable	Lead	y1	y2
y1	1	1.15977	-0.51058
	STD	0.05508	0.05898
	2	1.06612	-0.78872
	STD	0.10450	0.10702
	:	:	:
	14	0.08202	0.02870
	STD	0.10579	0.09483
	15	0.11080	-0.03083
	STD	0.09277	0.08778
y2	1	0.54634	0.38499
	STD	0.05779	0.06188
	2	0.84396	-0.13073
	STD	0.08481	0.08556
	:	:	:
	14	-0.03071	0.12557
	STD	0.10081	0.09391
	15	0.03299	0.06403
	STD	0.09375	0.08487

Figure 4.26. Impulse Response Function (IMPULSE= option)

Figure 4.26 is the part of output in a univariate format associated with the IMPULSE= option for the impulse response function. The keyword STD stands for the standard errors of the elements.

The VARMAX Procedure

Accumulated Impulse Response by Variable

Variable	Lead	y1	y2
y1	1	2.15977	-0.51058
	STD	0.05508	0.05898
	2	3.22589	-1.29929
	STD	0.21684	0.22776
	:	:	:
	14	3.11982	-2.53992
	STD	1.90364	1.84193
	15	3.23062	-2.57074
	STD	1.57743	1.52719
y2	1	0.54634	1.38499
	STD	0.05779	0.06188
	2	1.39030	1.25426
	STD	0.17614	0.18392
	:	:	:
	14	2.71782	-0.73442
	STD	2.57030	2.32369
	15	2.75080	-0.67040
	STD	2.08022	1.96462

Figure 4.27. Accumulated Impulse Response Function (IMPULSE= option)

Figure 4.27 is the part of output in a univariate format associated with the IMPULSE= option for the accumulated impulse response function.

The VARMAX Procedure

Orthogonalized Impulse Response by Variable

Variable	Lead	y1	y2
y1	0	1.13523	0
	STD	0.08068	0
	1	1.13783	-0.58120
	STD	0.10666	0.14110
	2	0.93412	-0.89782
	STD	0.13113	0.16776
	:	:	:
	14	0.10316	0.03267
	STD	0.09791	0.10849
	15	0.11499	-0.03509
	STD	0.08426	0.10065
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
	:	:	:
	14	0.00910	0.14294
	STD	0.09504	0.10739
	15	0.05987	0.07288
	STD	0.08779	0.09695

Figure 4.28. Orthogonalized Impulse Response Function (IMPULSE= option)

Figure 4.28 is the part of output in a univariate format associated with the IMPULSE= option for the orthogonalized impulse response function.

Covariance Matrices of Prediction Errors without Exogenous (Independent) Variables

Under the stationarity assumption, the optimal (MMSE) 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 interval forecasts 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 4.29 and Figure 4.30.

```
proc varmax data=simul1;
   model y1 y2 / p=1 noint print=(covpe(15))
                 printform=both;
run;
```

```
                    The VARMAX Procedure

            Prediction Error Covarinace Matrices

    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
       :                      :             :
      14    y1             7.93640       4.89643
            y2             4.89643       6.84041
      15    y1             7.94811       4.90204
            y2             4.90204       6.86092
```

Figure 4.29. Covariances of Prediction Errors (COVPE option)

Figure 4.29 is the output in a matrix format associated with the COVPE option for the prediction error covariance matrices.

```
                         The VARMAX Procedure

              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
                          :         :              :
                         14       7.93640        4.89643
                         15       7.94811        4.90204
           y2             1       0.39751        1.41839
                          2       1.00189        2.18051
                          3       1.98771        3.03498
                          :         :              :
                         14       4.89643        6.84041
                         15       4.90204        6.86092
```

Figure 4.30. Covariances of Prediction Errors Continued

Figure 4.30 is the output in a univariate format associated with the COVPE option for the prediction error covariances. This printing format more easily explains the forecast limit of each variable.

Covariance Matrices of Prediction Errors in 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 "State-space Modeling" section, $\mathbf{y}_{t+l|t}$ 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 $\mathbf{a}_t$, where $\mathbf{a}_t$ 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. See the "Forecasting" section on page 87 for details.

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 $\Sigma(l) = \sum_{j=0}^{l-1} \Psi_j^o \Psi_j^{o'}$, 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)th 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 VARMAX Procedure

Proportions of Prediction Error Covariances by Variable

Variable	Lead	y1	y2
y1	1	1.00000	0
	2	0.88436	0.11564
	3	0.75132	0.24868
	:	:	:
	14	0.55184	0.44816
	15	0.55237	0.44763
y2	1	0.08644	0.91356
	2	0.31767	0.68233
	3	0.50247	0.49753
	:	:	:
	14	0.46611	0.53389
	15	0.46473	0.53527

Figure 4.31. Decomposition of Prediction Error Covariances (DECOMPOSE option)

The proportions of decomposition of prediction error covariances of two variables are given in Figure 4.31. The output explains that about 91% of the one-step-ahead prediction error covariances of the variable y_{2t} is accounted for by its own innovations and about 9% is accounted for by y_{1t} innovations. For the long-term forecasts, 53.5% and 46.5% of the error variance is accounted for by y_{2t} and y_{1t} innovations.

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 endogenous (dependent) 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 zero values appended corresponding to the unobserved past observations. Let $\Delta(B)$ be the $k \times k$ matrix polynomial in the backshift operator corresponding 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

$$\Gamma(l) = \mathrm{E}[(\mathbf{y}_t - \boldsymbol{\mu}_t)(\mathbf{y}_t - \boldsymbol{\mu}_t)']$$

where $\boldsymbol{\mu}_t = \mathrm{E}(\mathbf{y}_t)$, 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, $\hat{\Gamma}(l)$ has (i,j)th element $\hat{\gamma}_{ij}(l) = c_{ij}(l)$. The sample cross-correlation matrix at lag l is computed as

$$\hat{\rho}_{ij}(l) = c_{ij}(l)/[c_{ii}(0)c_{jj}(0)]^{1/2}, \;\; i,j = 1,\ldots,k.$$

The following statements use the CORRY option to compute the sample cross-correlation matrices and their summary indicator plots in terms of $+$, $-$, and $\cdot$:

```
proc varmax data=simul1;
   model y1 y2 / p=1 noint lagmax=3 print=(corry)
                 printform=univariate;
run;
```

```
                          The VARMAX Procedure

                  Cross Correlations of Endogenous
                   (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
```

Figure 4.32. Cross-Correlations (CORRY option)

Figure 4.32 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.

Partial Autoregressive Matrices

For each $m = 1, 2, \ldots$, you can define a sequence of matrices Φ_{mm}, which is called the partial autoregression matrices of lag m:

$$\Gamma(l) = \sum_{i=1}^{m} \Gamma(l-i)\Phi'_{im}, \;\; l = 1, 2, \ldots, m.$$

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 print=(parcoef);
run;
```

```
                      The VARMAX Procedure

                  Partial Autoregression Matrices

      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
```

Figure 4.33. Partial Autoregression Matrices (PARCOEF option)

Figure 4.33 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.

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

$$P(m) = \Sigma_{m-1}^{*1/2} \Phi'_{mm} \Sigma_{m-1}^{-1/2}$$

where

$$\Sigma_{m-1} = \mathrm{Cov}(\mathbf{u}_{m,t}) = \Gamma(0) - \sum_{i=1}^{m-1} \Gamma(-i)\Phi'_{i,m-1}$$

and

$$\Sigma^{*}_{m-1} = \mathrm{Cov}(\mathbf{u}^{*}_{m,t-m}) = \Gamma(0) - \sum_{i=1}^{m-1} \Gamma(m-i)\Phi^{*'}_{m-i,m-1}.$$

$P(m)$ is called the partial cross-correlations 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-correlations matrices:

```
proc varmax data=simul1;
   model y1 y2 / p=1 noint lagmax=3 print=(pcorr)
                 printform=univariate;
run;
```

```
                    The VARMAX Procedure

             Partial Correlations by Variable

   Variable    Lag            y1             y2

   y1            1       0.80348        0.42672
                 2       0.00276        0.03978
                 3      -0.01091     0.00031802
   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
```

Figure 4.34. Partial Correlations (PCORR option)

The partial cross-correlation matrices in Figure 4.34 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.

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 $\rho_i^2(m)$ 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;
```

The VARMAX Procedure

Partial Canonical Correlations

Lag	PCanCorr1	PCanCorr2	Chi-Square	DF	Prob>ChiSq
1	0.91783	0.77335	142.61	4	<.0001
2	0.09171	0.01816	0.86	4	0.9307
3	0.10861	0.01078	1.16	4	0.8854

Figure 4.35. Partial Canonical Correlations (PCANCORR option)

Figure 4.35 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 a 0.05 significance level, indicating that an AR order of $m = 1$ can be an appropriate choice.

The Minimum Information Criterion (MINIC) method

The **MIN**imum **I**nformation **C**riterion (MINIC) method can tentatively identify the orders of a VARMA(p,q) process. Note that Spliid (1983), Koreisha and Pukkila (1989), and Quinn (1980) proposed this method. 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

$$\tilde{\boldsymbol{\epsilon}}_t = \mathbf{y}_t - \sum_{i=1}^{p_\epsilon} \hat{\Phi}_i^{p_\epsilon} \mathbf{y}_{t-i} - \hat{\delta}^{p_\epsilon}, \;\; t = p_\epsilon + 1, \ldots, T.$$

In the second step, you select the order (p, q) of the VARMA model for p in $(p_{min} : p_{max})$ and q in $(q_{min} : q_{max})$

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

which minimizes a selection criterion like SBC or HQ.

The following statements use the MINIC= option to compute a table containing 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;
```

The VARMAX Procedure

Minimum Information Criterion

Lag	MA 0	MA 1	MA 2	MA 3
AR 0	3.2859653	3.0570091	2.7272377	2.3526368
AR 1	0.4862246	0.6507991	0.7120257	0.7859524
AR 2	0.5800236	0.7407785	0.802996	0.850487
AR 3	0.6696452	0.8131261	0.8395558	0.9094856

Figure 4.36. MINIC= option

Figure 4.36 shows the output associated with the MINIC= option. The criterion takes the smallest value at AR order 1.

VAR Modeling

The pth-order VAR process is written as

$$\mathbf{y}_t - \boldsymbol{\mu} = \sum_{i=1}^{p} \Phi_i (\mathbf{y}_{t-i} - \boldsymbol{\mu}) + \boldsymbol{\epsilon}_t \;\text{ or }\; \Phi(B)(\mathbf{y}_t - \boldsymbol{\mu}) = \boldsymbol{\epsilon}_t$$

Equivalently, it can be written as

$$\mathbf{y}_t = \boldsymbol{\delta} + \sum_{i=1}^{p} \Phi_i \mathbf{y}_{t-i} + \boldsymbol{\epsilon}_t \;\text{ or }\; \Phi(B)\mathbf{y}_t = \boldsymbol{\delta} + \boldsymbol{\epsilon}_t$$

with $\boldsymbol{\delta} = (I_k - \sum_{i=1}^{p} \Phi_i)\boldsymbol{\mu}$.

Stationarity

For stationarity of the VAR process, it 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 $\Phi(B)\Psi(B) = I$, and 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) may 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;
```

```
                         The VARMAX Procedure

                  Roots of AR Characteristic Polynomial

Index         Real      Imaginary      Modulus    ATAN(I/R)       Degree

    1      0.77238        0.35899       0.8517       0.4351      24.9284
    2      0.77238       -0.35899       0.8517      -0.4351     -24.9284
```

Figure 4.37. Stationarity (ROOTS option)

Figure 4.37 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.

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,

$$Y = XB + E \;\; \text{or} \;\; \mathbf{y} = (X \otimes I_k)\boldsymbol{\beta} + \mathbf{e}$$

where

$$\begin{aligned}
Y &= (\mathbf{y}_1, \ldots, \mathbf{y}_T)' \\
B &= (\delta, \Phi_1, \ldots, \Phi_p)' \\
X &= (X_0, \ldots, X_{T-1})' \\
X_t &= (1, \mathbf{y}'_t, \ldots, \mathbf{y}'_{t-p+1})' \\
E &= (\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_T)' \\
\mathbf{y} &= \text{vec}(Y') \\
\boldsymbol{\beta} &= \text{vec}(B') \\
\mathbf{e} &= \text{vec}(E')
\end{aligned}$$

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. The commutation matrix K_k defines as $K_k\mathrm{vec}(A) = \mathrm{vec}(A')$; the duplication matrix D_k, $D_k\mathrm{vech}(A) = \mathrm{vec}(A)$; the elimination matrix L_k, $L_k\mathrm{vec}(A) = \mathrm{vech}(A)$, for any $k \times k$ matrix A.

The asymptotic distributions of the impulse response function is

$$\sqrt{T}\mathrm{vec}(\hat{\Psi}_j - \Psi_j) \xrightarrow{d} N(0, G_j \Sigma_{\boldsymbol{\beta}} G_j') \ \ j = 1, 2, \ldots$$

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 distributions of the accumulated impulse response function is

$$\sqrt{T}\mathrm{vec}(\hat{\Psi}_l^a - \Psi_l^a) \xrightarrow{d} N(0, F_l \Sigma_{\boldsymbol{\beta}} F_l') \ \ l = 1, 2, \ldots$$

where $F_l = \sum_{j=1}^{l} G_j$.

The asymptotic distributions of the orthogonalized impulse response function is

$$\sqrt{T}\mathrm{vec}(\hat{\Psi}_j^o - \Psi_j^o) \xrightarrow{d} N(0, C_j \Sigma_{\boldsymbol{\beta}} C_j' + \bar{C}_j \Sigma_{\boldsymbol{\sigma}} \bar{C}_j') \ \ j = 0, 1, 2, \ldots$$

where $C_0 = 0$, $C_j = (\Psi_0^{o'} \otimes I_k) G_j$, $\bar{C}_j = (I_k \otimes \Psi_j) H$ and

$$H = \frac{\partial \mathrm{vec}(\Psi_0^o)}{\partial \boldsymbol{\sigma}'} = L_k' \{ L_k (I_{k^2} + K_k)(\Psi_0^o \otimes I_k) L_k' \}^{-1}$$

and $\Sigma_{\boldsymbol{\sigma}} = 2D_k^{+}(\Sigma \otimes \Sigma) D_k^{+'}$ 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:

$$\Phi(B)\mathbf{y}_t = \begin{bmatrix} \Phi_{11}(B) & \Phi_{12}(B) \\ \Phi_{21}(B) & \Phi_{22}(B) \end{bmatrix} \begin{bmatrix} \mathbf{y}_{1t} \\ \mathbf{y}_{2t} \end{bmatrix} = \boldsymbol{\delta} + \begin{bmatrix} \boldsymbol{\epsilon}_{1t} \\ \boldsymbol{\epsilon}_{2t} \end{bmatrix}.$$

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 $s \times (k^2p + k)$ matrix of rank s and c is a s-dimensional vector where $s = k_1k_2p$. Assuming that

$$\sqrt{T}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \stackrel{d}{\to} N(0, \Gamma_p^{-1} \otimes \Sigma)$$

you get the Wald statistic

$$T(C\hat{\boldsymbol{\beta}} - c)'[C(\Gamma_p^{-1} \otimes \Sigma)C']^{-1}(C\hat{\boldsymbol{\beta}} - c) \stackrel{d}{\to} \chi^2(s).$$

For the Granger-Causality Test, the matrix C consists of zeros or ones and c is the zero vector.

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

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

The parameter estimates can be obtained by representing the general form of the multivariate linear model,

$$Y = XB + E \;\text{ or }\; \mathbf{y} = (X \otimes I_k)\boldsymbol{\beta} + \mathbf{e}$$

where

$$\begin{aligned}
Y &= (\mathbf{y}_1, \ldots, \mathbf{y}_T)' \\
B &= (\delta, \Phi_1, \ldots, \Phi_p, \Theta_0^*, \ldots, \Theta_s^*)' \\
X &= (X_0, \ldots, X_{T-1})' \\
X_t &= (1, \mathbf{y}_t', \ldots, \mathbf{y}_{t-p+1}', \mathbf{x}_{t+1}', \ldots, \mathbf{x}_{t-s+1}')' \\
E &= (\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_T)' \\
\mathbf{y} &= \text{vec}(Y') \\
\boldsymbol{\beta} &= \text{vec}(B') \\
\mathbf{e} &= \text{vec}(E').
\end{aligned}$$

The conditional least-squares estimator of β can be obtained using the same method in a VAR(p) modeling. If the multivariate linear model has different independent variables corresponding 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=nf sur;
   endogenous y1 y2;
   model y1 = x1-x3;
   model y2 = x2 x3;
run;
```

From the output in Figure 4.19, you can see that the parameters, XL0_1_2, XL0_2_1, XL0_3_1, and XL0_3_2, 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;
run;
```

```
                         The VARMAX Procedure

                      XLag Coefficient Estimates

        Lag    Variable              x1              x2

          0    y1               1.83231               _
               y2                     _         2.42110
               y3                     _               _
```

Figure 4.38. Parameter Estimates for VARX(1,0) Model

As you can see in Figure 4.38, the symbol '_' in the elements of matrix correspond to endogenous variables that do not take exogenous variables.

Model Diagnostic Checks

Multivariate Model Diagnostic Checks

- Information Criterion

  Various model selection criteria (normalized by T) can be used to choose the appropriate model. The following list includes the Akaike Information Criterion (AIC), the corrected Akaike 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.

$$\begin{aligned} \mathrm{AIC} &= \log(|\tilde{\Sigma}|) + 2r/T \\ \mathrm{AICC} &= \log(|\tilde{\Sigma}|) + 2r/(T - r/k) \\ \mathrm{FPE} &= (\frac{T + r/k}{T - r/k})^k |\tilde{\Sigma}| \\ \mathrm{HQC} &= \log(|\tilde{\Sigma}|) + 2r\log(\log(T))/T \\ \mathrm{SBC} &= \log(|\tilde{\Sigma}|) + r\log(T)/T \end{aligned}$$

  where r denotes the number of parameters estimated and $\tilde{\Sigma}$ is the maximum likelihood estimate of Σ.

  An example of the output was displayed in Figure 4.4.

- Portmanteau Statistic Q_s

  Let $C_\epsilon(l)$ be the residual cross-covariance matrices and $\hat{\rho}_\epsilon(l)$ 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

$$Q_s = T^2 \sum_{l=1}^{s} (T-l)^{-1} \text{tr}\{\hat{\rho}_\epsilon(l)\Sigma^{-1}\hat{\rho}_\epsilon(-l)\Sigma^{-1}\}$$

The statistic Q_s has approximately the chi-square distribution with $k^2(s-p)$ degrees of freedom. An example of the output was displayed in Figure 4.6.

Univariate Model Diagnostic Checks

There are various ways to perform diagnostic checks for a univariate model. For details, see the chapter on the ARIMA or AUTOREG procedure. An example of the output was displayed in Figure 4.7 and Figure 4.8.

- Durbin-Watson (DW) statistics: The test statistics are computed from the residuals of the autoregressive model with order 1.
- F tests for autoregressive conditional heteroscedastic (ARCH) disturbances: These test statistics are computed from the residuals of the ARCH(1) model.
- F tests for AR disturbance: These test statistics are computed from the residuals of the univariate AR(1), AR(2), AR(3), and AR(4) models.
- Jarque-Bera normality test: This test is helpful in determining whether the model residuals represent a white noise process.

Bayesian VAR 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

$$f(\boldsymbol{\beta}) = (\frac{1}{2\pi})^{k^2p/2} |V_\beta|^{-1/2} \exp[-\frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\beta}^*) V_\beta^{-1} (\boldsymbol{\beta} - \boldsymbol{\beta}^*)].$$

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)th 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,

$$\begin{aligned}y_{1t} &= 0 + \phi_{1,11}y_{1,t-1} + \phi_{1,12}y_{2,t-1} + \phi_{2,11}y_{1,t-2} + \phi_{2,12}y_{2,t-2} + \epsilon_{1t} \\ y_{2t} &= 0 + \phi_{1,21}y_{1,t-1} + \phi_{1,22}y_{2,t-1} + \phi_{2,21}y_{1,t-2} + \phi_{2,22}y_{2,t-2} + \epsilon_{1t}\end{aligned}$$

with the prior covariance matrix

$$\begin{aligned}V_{\beta} = \mathrm{Diag} \quad (\quad & \infty, \lambda^2, (\lambda\theta\sigma_1/\sigma_2)^2, (\lambda/2)^2, (\lambda\theta\sigma_1/2\sigma_2)^2, \\ & \infty, (\lambda\theta\sigma_2/\sigma_1)^2, \lambda^2, (\lambda\theta\sigma_2/2\sigma_1)^2, (\lambda/2)^2 \quad).\end{aligned}$$

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 bootstrap procedure is used to estimate standard errors of the forecast (Litterman 1986). NREP=B simulations are performed. In each simulation the following steps are taken:

1. The procedure generates the available number of observations, T, and uniform random integers I_t, where $t = 1, \ldots T$.

2. A new observation, $\tilde{\mathbf{y}}_t$, is obtained as a sum of the forecast based on the estimates coefficients plus the vector of residuals from the I_t; that is,

$$\tilde{\mathbf{y}}_t = \sum_{j=1}^{p} \hat{\Phi}_j \mathbf{y}_{t-j} + \hat{\boldsymbol{\epsilon}}_{I_t}.$$

3. A new BVAR model is estimated by using the most recent observations, and a prediction value is made of the most recent observations.

The MSE measure of the l-step-ahead forecast is

$$MSE(l) = \frac{1}{B} \sum_{i=1}^{B} (\tilde{\mathbf{y}}^i_{t+l|t} - \bar{\tilde{\mathbf{y}}}_t)^2$$

where $\bar{\tilde{\mathbf{y}}}_t = (1/B) \sum_{i=1}^{B} \bar{\tilde{\mathbf{y}}}^i_t$.

Cointegration

This section briefly introduces the concepts of cointegration.

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.

$$\Delta\mathbf{y}_t = \boldsymbol{\delta} + \Psi(B)\boldsymbol{\epsilon}_t, \quad \sum_{j=0}^{\infty} j|\Psi_j| < \infty$$

where $\boldsymbol{\epsilon}_t$ is $iid(0, \Omega)$ and $\Psi(B) = \sum_{j=0}^{\infty} \Psi_j B^j$ with $\Psi_0 = I_k$.

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 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

$$\boldsymbol{\delta} = C(1)\tilde{\boldsymbol{\delta}}$$

where $\tilde{\boldsymbol{\delta}}$ is a k-dimensional vector. The common trends representation is written as

$$\begin{aligned}\mathbf{y}_t &= \mathbf{y}_0 + \Psi(1)[\tilde{\boldsymbol{\delta}} t + \sum_{i=0}^{t} \boldsymbol{\epsilon}_i] + \Psi^*(B)\boldsymbol{\epsilon}_t \\ &= \mathbf{y}_0 + \Psi(1)H[H^{-1}\tilde{\boldsymbol{\delta}} t + H^{-1}\sum_{i=0}^{t} \boldsymbol{\epsilon}_i] + \mathbf{a}_t \\ &= \mathbf{y}_0 + A\boldsymbol{\tau}_t + \mathbf{a}_t\end{aligned}$$

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 k-dimensional time series $\mathbf{y}_t$ has $m \leq k$ common stochastic trends, and the alternative is that it has $s < m$ common trends. The test procedure of m vs 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 test Stock-Watson common trends:

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

```
                         The VARMAX Procedure

                   Testing for Stock-Watson's Common
                   Trends using Differencing Filter

     H_0:      H_1:                                Critical
   Rank=m    Rank=s    Eigenvalue         Filt        Value     Lag

        1         0        1.0009       0.0906       -14.10       2
        2         0        0.9968      -0.3237        -8.80
                  1        0.6489     -35.1092       -23.00
```

Figure 4.39. Common Trends Test (COINTTEST=(SW) option)

In Figure 4.39, the first column is the null hypothesis that $\mathbf{y}_t$ has $m \leq k$ common trends; the second column, the alternative hypothesis that $\mathbf{y}_t$ has $s < m$ common trends; the fourth column, the test statistics using AR(2) filtering the data. The test statistics for testing for 2 versus 1 common trends are more negative (-35.1) than the critical value (-23.0). 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. Assume that the cointegrated series can be represented by a vector error correction model according to the Granger representation theorem (Engle and Granger 1987). Consider the vector autoregressive process with 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 $\boldsymbol{\epsilon}_t \sim N(0, \Sigma)$. Since the AR operator $\Phi(B)$ can be re-expressed as $\Phi(B) = \Phi^*(B)(1-B) + \Phi(1)B$ where $\Phi^*(B) = I_k - \sum_{i=1}^{p-1} \Phi_i^* B^i$ with $\Phi_i^* = -\sum_{j=i+1}^{p} \Phi_j$, the vector error correction model is

$$\Phi^*(B)(1-B)\mathbf{y}_t = \boldsymbol{\alpha}\boldsymbol{\beta}'\mathbf{y}_{t-1} + \boldsymbol{\epsilon}_t$$

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 $\boldsymbol{\alpha}\boldsymbol{\beta}' = -\Phi(1) = -I_k + \Phi_1 + \Phi_2 + \cdots + \Phi_p$.

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 and assume that the agents react to the disequilibrium error $\boldsymbol{\beta}'\mathbf{y}_t - \mathbf{c}$ through the adjustment coefficient $\boldsymbol{\alpha}$ to restore equilibrium; that is, they satisfy the economic relations. The cointegrating vector, $\boldsymbol{\beta}$ is sometimes called the *long-run parameters*.

You can consider a vector error correction model with a deterministic term. The deterministic term D_t can contain a constant, a linear trend, seasonal dummy variables, or nonstochastic regressors.

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

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

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

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 $rank(\Pi) = 0$. 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 $\boldsymbol{\beta}$ since the coefficient matrix Π can also be decomposed as

$$\Pi = \boldsymbol{\alpha}MM^{-1}\boldsymbol{\beta}' = \boldsymbol{\alpha}^*\boldsymbol{\beta}^{*'}$$

where M is an $r \times r$ nonsingular matrix.

Test for the Cointegration

The cointegration rank test determines the linearly independent columns of Π. Johansen (1988, 1995a) and Johansen and Juselius (1990) proposed the cointegration rank test using the reduced rank regression.

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 will not be present in the VECM(p) form. On the other hand, if there are stochastic cointegrated relationships, deterministic terms appear in the VECM(p) form via the error correction term or as an independent term 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.

$$\Delta \mathbf{y}_t = \boldsymbol{\alpha}(\boldsymbol{\beta}', \beta_0)(\mathbf{y}_{t-1}', 1)' + \sum_{i=1}^{p-1} \Phi_i^* \Delta \mathbf{y}_{t-i} + \boldsymbol{\epsilon}_t$$

- **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.

$$\Delta \mathbf{y}_t = \boldsymbol{\alpha}(\boldsymbol{\beta}', \beta_1)(\mathbf{y}_{t-1}', t)' + \sum_{i=1}^{p-1} \Phi_i^* \Delta \mathbf{y}_{t-i} + \boldsymbol{\delta}_0 + \boldsymbol{\epsilon}_t$$

- **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; $(1, t)$ for Case 5. Let Ψ be the matrix of parameters consisting of $\Phi_1^*, \ldots, \Phi_{p-1}^*$ and A, where parameters A corresponds to 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

$$\ell = -\frac{kT}{2}\log 2\pi - \frac{T}{2}\log|\Sigma| - \frac{1}{2}\sum_{t=1}^{T}(Z_{0t} - \boldsymbol{\alpha}\boldsymbol{\beta}' Z_{1t} - \Psi Z_{2t})'\Sigma^{-1}(Z_{0t} - \boldsymbol{\alpha}\boldsymbol{\beta}' Z_{1t} - \Psi Z_{2t})$$

The residuals, R_{0t} and R_{1t}, are obtained by regressing Z_{0t} and Z_{1t} on Z_{2t}, respectively. The regression equation in residuals is

$$R_{0t} = \boldsymbol{\alpha}\boldsymbol{\beta}' R_{1t} + \hat{\boldsymbol{\epsilon}}_t$$

The crossproducts matrices are computed

$$S_{ij} = \frac{1}{T}\sum_{t=1}^{T} R_{it}R'_{jt}, \;\; i,j = 0,1$$

Then the maximum likelihood estimator for $\boldsymbol{\beta}$ is obtained from the eigenvectors corresponding 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 corresponding 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) suggested 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

$$\lambda_{trace} = -T\sum_{i=r+1}^{k}\log(1-\lambda_i)$$

The asymptotic distribution of this statistic is given by

$$tr\left\{\int_0^1 (dW)\tilde{W}'\left(\int_0^1 \tilde{W}\tilde{W}'dr\right)^{-1}\int_0^1 \tilde{W}(dW)'\right\}$$

where $tr(A)$ 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 demeaned or detrended Brownian motion according to the different specifications of deterministic trends in the vector error correction model.

Maximum Eigenvalue Test

$$\lambda_{max} = -T \log(1 - \lambda_{r+1})$$

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 the detailed tables of critical values of these statistics.

In case 2, consider that $Z_{1t} = (\mathbf{y}_t', 1)'$ and $Z_{2t} = (\Delta\mathbf{y}_{t-1}', \ldots, \Delta\mathbf{y}_{t-p+1}')'$. In case 4, $Z_{1t} = (\mathbf{y}_t', t)'$ and $Z_{2t} = (\Delta\mathbf{y}_{t-1}', \ldots, \Delta\mathbf{y}_{t-p+1}', 1)'$.

The following statements use the JOHANSEN option to compute the Johansen cointegration rank test of integrated order 1:

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

The VARMAX Procedure

Cointegration Rank Test

H_0: Rank=r	H_1: Rank>r	Eigenvalue	Trace	Critical Value	Drift InECM	DriftIn Process
0	0	0.4644	61.75	15.34	Constant	Linear
1	1	0.0056	0.56	3.84		

Cointegration Rank Test under the Restriction

H_0: Rank=r	H_1: Rank>r	Eigenvalue	Trace	Critical Value	Drift InECM	DriftIn Process
0	0	0.5209	76.38	19.99	Constant	Constant
1	1	0.0426	4.27	9.13		

Figure 4.40. Cointegration Rank Test (JOHANSEN option)

Suppose that the model has an intercept term. The first table in Figure 4.40 shows the trace statistics based on case 3; the second table, case 2. The output indicates that the series are cointegrated with rank 1.

```
                              The VARMAX Procedure

                            Test of the Restriction
                                  when Rank=r

                                 Drift        DriftIn
                         Hypo    InECM        Process

                         H_0     Constant     Linear
                         H_1     Constant     Constant


                      Test of the Restriction when Rank=r

              Eigenvalue
                      On                          Chi-                Prob>
         Rank    Restrict    Eigenvalue         Square      DF        ChiSq

            0      0.5209        0.4644          14.63       2       0.0007
            1      0.0426        0.0056           3.71       1       0.0540
```

Figure 4.41. Cointegration Rank Test Continued

Figure 4.41 shows which result, case 3 (the hypothesis H_0) and case 2 (the hypothesis H_1), is appropriate. Since the cointegration rank is chosen to be 1 by the result in Figure 4.40, look at the last row corresponding to rank=1.

```
                              The VARMAX Procedure

                       Long-Run Parameter BETA Estimates

                     Variable        Dummy 1        Dummy 2

                     y1              1.00000        1.00000
                     y2             -2.04869       -0.02854


                     Adjustment Coefficient ALPHA Estimates

                     Variable        Dummy 1        Dummy 2

                     y1             -0.46421       -0.00502
                     y2              0.17535       -0.01275
```

Figure 4.42. Cointegration Rank Test Continued

Figure 4.42 shows the estimates of long-run parameter and adjustment coefficients based on case 3. Considering that the cointegration rank is 1, the long-run relationship of the series is

$$y_{1t} = 2.049\, y_{2t}$$

The VARMAX Procedure

Long-Run Coefficient BETA based on the Restricted Trend

Variable	Dummy 1	Dummy 2	Dummy 3
y1	1.00000	1.00000	1.00000
y2	-2.04366	-2.75773	-0.73198
1	6.75919	101.37051	-42.50051

Adjustment Coefficient ALPHA based on the Restricted Trend

Variable	Dummy 1	Dummy 2	Dummy 3
y1	-0.48015	0.01091	5.96583E-14
y2	0.12538	0.03722	-2.834E-13

Figure 4.43. Cointegration Rank Test Continued

Figure 4.43 shows the estimates of long-run parameter and adjustment coefficients based on case 2. Considering that the cointegration rank is 1, the long-run relationship of the series is

$$y_{1t} = 2.044\, y_{2t} - 6.760$$

Estimation of Vector Error Correction Model

Now consider cases 1, 3, and 5 for the 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

$$\hat{\boldsymbol{\alpha}}_{\perp} = S_{00}^{-1} S_{01}[v_{r+1}, \ldots, v_k]$$

The ML estimators have the following asymptotic properties:

$$\sqrt{T}\mathrm{vec}([\hat{\Pi}, \hat{\Psi}] - [\Pi, \Psi]) \stackrel{d}{\to} N(0, \Sigma_{co})$$

where

$$\Sigma_{co} = \Sigma \otimes \left(\begin{bmatrix} \boldsymbol{\beta} & 0 \\ 0 & I_k \end{bmatrix} \Omega^{-1} \begin{bmatrix} \boldsymbol{\beta}' & 0 \\ 0 & I_k \end{bmatrix} \right)$$

and

$$\Omega = \text{plim}\frac{1}{T}\begin{bmatrix} \boldsymbol{\beta}' Z_1' Z_1 \boldsymbol{\beta} & \boldsymbol{\beta}' Z_1' Z_2 \\ Z_2' Z_1 \boldsymbol{\beta} & Z_2' Z_2 \end{bmatrix}$$

Test for the Linear Restriction of β

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 $\boldsymbol{\beta} = H\psi$, 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} are given in the preceding section. Then choose $\hat{\phi} = (v_1, \ldots, v_r)$ corresponding 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 data have no deterministic trend, the constant term should be restricted by $\boldsymbol{\alpha}'_{\perp} \boldsymbol{\delta}_0 = 0$ like 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 $\beta_1 + 2\beta 2 = 0$:

```
proc varmax data=simul2;
   model y1 y2 / p=2 ecm=(rank=1 normalize=y1);
   cointeg rank=1 h=(1,-2);
run;
```

```
                          The VARMAX Procedure

                         Long-Run Coefficient
                         BETA with respect to
                          Hypothesis on BETA

                        Variable        Dummy 1

                        y1              1.00000
                        y2             -2.00000


                        Adjustment Coefficient
                         ALPHA with respect to
                          Hypothesis on BETA

                        Variable        Dummy 1

                        y1             -0.47404
                        y2              0.17534


              Test for Restricted Long-Run Coefficient BETA

              Eigenvalue                         Chi-           Prob>
     Index    OnRestrict    Eigenvalue        Square     DF     ChiSq

         1        0.4616        0.4644          0.51      1    0.4738
```

Figure 4.44. Testing of Linear Restriction β (H= option)

Figure 4.44 shows the results of testing $H_0 : \beta_1 + 2\beta 2 = 0$. 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.

Test for the Weak Exogeneity and Restrictions of α

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 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 $\mathbf{y}_{2t}$,

$$\Delta \mathbf{y}_{2t} = \alpha_2 \boldsymbol{\beta}'\mathbf{y}_{t-1} + \sum_{i=1}^{p-1} \Phi^*_{2i}\Delta \mathbf{y}_{t-i} + A_2 D_t + \boldsymbol{\epsilon}_{2t}$$

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 $H_0\colon \boldsymbol{\alpha} = J\psi$

Consider the null hypothesis $H_0\colon \boldsymbol{\alpha} = J\psi$, 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

$$\Omega_{JJ_\perp} = \bar{J}'\Omega J_\perp \text{ and } \Omega_{J_\perp J_\perp} = J'_\perp \Omega J_\perp$$

and let $\omega = \Omega_{JJ_\perp}\Omega^{-1}_{J_\perp J_\perp}$. Then $\bar{J}' R_{0t}$ can be written

$$\bar{J}' R_{0t} = \psi\boldsymbol{\beta}' R_{1t} + \omega J'_\perp R_{0t} + \bar{J}'\hat{\boldsymbol{\epsilon}}_t - \omega J'_\perp \hat{\boldsymbol{\epsilon}}_t$$

Using the marginal distribution of $J'_\perp R_{0t}$ and the conditional distribution of $\bar{J}'R_{0t}$, the new residuals are computed as

$$\begin{aligned}\tilde{R}_{Jt} &= \bar{J}'R_{0t} - S_{JJ_\perp}S^{-1}_{J_\perp J_\perp}J'_\perp R_{0t} \\ \tilde{R}_{1t} &= R_{1t} - S_{1J_\perp}S^{-1}_{J_\perp J_\perp}J'_\perp R_{0t}\end{aligned}$$

where

$$S_{JJ_\perp} = \bar{J}'S_{00}J_\perp,\;\; S_{J_\perp J_\perp} = J'_\perp S_{00}J_\perp,\;\; \text{and}\;\; S_{J_\perp 1} = J'_\perp S_{01}$$

In terms of $\tilde{R}_{Jt}$ and $\tilde{R}_{1t}$, the MLE of $\boldsymbol{\beta}$ is computed by using the reduced rank regression. Let

$$S_{ij.J_\perp} = \frac{1}{T}\sum_{t=1}^{T}\tilde{R}_{it}\tilde{R}'_{jt},\;\; \text{for}\;\; i,j = 1, J$$

Under the null hypothesis $H_0 : \boldsymbol{\alpha} = J\psi$, the MLE $\tilde{\boldsymbol{\beta}}$ is computed by solving the equation

$$|\rho S_{11.J_\perp} - S_{1J.J_\perp}S^{-1}_{JJ.J_\perp}S_{J1.J_\perp}| = 0$$

Then $\tilde{\boldsymbol{\beta}} = (v_1, \ldots, v_r)$, where the eigenvectors correspond to the r largest eigenvalues. The likelihood ratio test for $H_0 : \boldsymbol{\alpha} = J\psi$ is

$$T\sum_{i=1}^{r}\log\{(1-\rho_i)/(1-\lambda_i)\} \xrightarrow{d} \chi^2_{r(k-m)}$$

The test of weak exogeneity of $\mathbf{y}_{2t}$ is the 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^b_t, i^d_t). If $r = 1$, you formulate the weak exogeneity of (y_t, i^b_t, i^d_t) for m_t as $J = [1, 0, 0, 0]'$ and the weak exogeneity of i^d_t for (m_t, y_t, i^b_t) as $J = [I_3, 0]'$.

The following statements test the weak exogeneity of other variables:

```
proc varmax data=simul2;
   model y1 y2 / p=2 ecm=(rank=1 normalize=y1);
   cointeg rank=1 exogeneity;
run;
```

The VARMAX Procedure

Tests of Weak Exogeneity
of Each of Variables

Variable	Chi-Square	DF	Prob> ChiSq
y1	53.46	1	<.0001
y2	8.76	1	0.0031

Figure 4.45. Testing of Weak Exogeneity (EXOGENEITY option)

Figure 4.45 shows that each variable is not the weak exogeneity of other variable.

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

$$\mathbf{y}_t = \boldsymbol{\delta} t + \sum_{i=1}^{t} \sum_{j=0}^{t-i} \Psi_j \boldsymbol{\epsilon}_i$$

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:

$$\mathbf{y}_{t+l|t} = \boldsymbol{\delta}(t+l) + \sum_{i=1}^{t} \sum_{j=0}^{t+l-i} \Psi_j \boldsymbol{\epsilon}_i$$

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, though 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 $\mathbf{z}_t = (\mathbf{y}'_{t-1}, \Delta\mathbf{y}'_t, \Delta\mathbf{y}'_{t-1}, \cdots, \Delta\mathbf{y}'_{t-p+2})'$ 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 $\underline{0}$ is a k-dimensional zero vector. The observation equation can be written

$$\mathbf{y}_t = \delta t + H\mathbf{z}_t$$

where $H = [I_k, I_k, 0, \ldots, 0]$.

The l-step-ahead forecast is computed as

$$\mathbf{y}_{t+l|t} = \delta(t+l) + HF^l\mathbf{z}_t$$

$\mathrm{I}(2)$ Model

The VAR(p) model can be written as the error correction form.

$$\Delta\mathbf{y}_t = \boldsymbol{\alpha}\boldsymbol{\beta}'\mathbf{y}_{t-1} + \sum_{i=1}^{p-1} \Phi_i^* \Delta\mathbf{y}_{t-i} + AD_t + \boldsymbol{\epsilon}_t$$

Let $\Phi^* = I_k - \sum_{i=1}^{p-1} \Phi_i^*$. If $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ have full rank r, and if

$$rank(\boldsymbol{\alpha}'_\perp \Phi^* \boldsymbol{\beta}_\perp) = k - r$$

then $\mathbf{y}_t$ is an $I(1)$ process. If the condition $rank(\boldsymbol{\alpha}'_\perp \Phi^* \boldsymbol{\beta}_\perp) = k - r$ 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 $s \leq k - r$, $\boldsymbol{\alpha}_\perp$ 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 details, see Johansen (1995a).

The implication of the $I(2)$ model for the autoregressive representation is given by

$$\Delta^2\mathbf{y}_t = \Pi\mathbf{y}_{t-1} - \Phi^*\Delta\mathbf{y}_{t-1} + \sum_{i=1}^{p-2} \Psi_i \Delta^2\mathbf{y}_{t-i} + AD_t + \boldsymbol{\epsilon}_t$$

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

Test for $\mathrm{I}(2)$

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 $\leq s$. The following table shows the relation between the $I(1)$ models and the $I(2)$ models.

Table 4.1. 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 (1995a) 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 $\Delta^2 \mathbf{y}_{t-1}, \ldots, \Delta^2 \mathbf{y}_{t-p+2}, D_t$. This gives residuals R_{0t}, R_{1t}, and R_{2t} and residual product moment matrices

$$M_{ij} = \frac{1}{T} \sum_{t=1}^{T} R_{it} R_{jt}' \text{ for } i, j = 0, 1, 2$$

Perform the reduced rank regression analysis $\Delta^2 \mathbf{y}_t$ on $\mathbf{y}_{t-1}$ corrected for $\Delta \mathbf{y}_{t-1}$, $\Delta^2 \mathbf{y}_{t-1}, \ldots, \Delta^2 \mathbf{y}_{t-p+2}, D_t$ 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

$$|\rho M_{\boldsymbol{\beta}_\perp \boldsymbol{\beta}_\perp . \boldsymbol{\beta}} - M_{\boldsymbol{\beta}_\perp \boldsymbol{\alpha}_\perp . \boldsymbol{\beta}} M_{\boldsymbol{\alpha}_\perp \boldsymbol{\alpha}_\perp . \boldsymbol{\beta}}^{-1} M_{\boldsymbol{\alpha}_\perp \boldsymbol{\beta}_\perp . \boldsymbol{\beta}}| = 0$$

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 $\leq s$ 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 are to test the rank test for the cointegrated order 2:

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

The VARMAX Procedure

Trace of Cointegration Rank Test for I(2)

r\k-r-s	2	1	Trace_I1	Critical Value_I1
0	720.40735	308.69199	61.75	15.34
1		211.84512	0.56	3.84
Critical Value_I2	15.34000	3.84000		

Figure 4.46. Cointegrated I(2) Test (IORDER= option)

The last two columns in Figure 4.46 explain the cointegration rank test with integrated order 1. The results indicate that there is the cointegrated relationship with the cointegration rank 1 with respect to a 0.05 significance level. Now, look at the row in the case of $r = 1$. Compare the value to the critical value for the cointegrated order 2. There is no evidence that the series are integrated order 2 with a 0.05 significance level.

OUT= Data Set

The OUT= data set contains the forecast values produced by the OUTPUT statement. The following output variables can be created:

- the BY variables
- the ID variable

- the MODEL statement endogenous (dependent) variables. These variables contain the actual values from the input data set.
- FOR*i*, numeric variables containing the forecasts. The FOR*i* variables contain the forecasts for the *i*th endogenous variable in the MODEL statement list. Forecasts are 1-step-ahead predictions until the end of the data or until the observation specified by the BACK= option.
- RES*i*, numeric variables containing the residual for the forecast of the *i*th endogenous variable in the MODEL statement list. For forecast observations, the actual values are missing and the RES*i* variables contain missing values.
- STD*i*, numeric variables containing 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 containing the lower confidence limits for the corresponding forecasts of the *i*th endogenous variable in the MODEL statement list.
- UCI*i*, numeric variables containing the upper confidence limits for the corresponding forecasts of the *i*th endogenous variable in the MODEL statement list.

Table 4.2. 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}
⋮							

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}
⋮						

The OUT= data set contains the values shown in Table 4.2 for a bivariate case.

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;
```

Obs	date	y1	FOR1	RES1	STD1	LCI1	UCI1
98	1997	-0.58433	-0.13500	-0.44934	1.13523	-2.36001	2.09002
99	1998	-2.07170	-1.00649	-1.06522	1.13523	-3.23150	1.21853
100	1999	-3.38342	-2.58612	-0.79730	1.13523	-4.81113	-0.36111
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

Obs	y2	FOR2	RES2	STD2	LCI2	UCI2
98	0.64397	-0.34932	0.99329	1.19096	-2.68357	1.98492
99	0.35925	-0.07132	0.43057	1.19096	-2.40557	2.26292
100	-0.64999	-0.99354	0.34355	1.19096	-3.32779	1.34070
101	.	-2.09873	.	1.19096	-4.43298	0.23551
102	.	-2.77050	.	1.47666	-5.66469	0.12369
103	.	-2.75724	.	1.74212	-6.17173	0.65725
104	.	-2.24943	.	2.01925	-6.20709	1.70823
105	.	-1.47460	.	2.25169	-5.88782	2.93863

Figure 4.47. OUT= Data Set

The output in Figure 4.47 shows part of the results of the OUT= data set.

OUTEST= Data Set

The OUTEST= data set contains estimation results of the fitted model. The following output variables can be created:

- the BY variables
- NAME, a character variable containing the name of endogenous (dependent) variables or the name of the parameters for the covariance of the matrix of the parameter estimates if the OUTCOV option is specified
- TYPE, a character variable containing 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 the OUTCOV option is specified
- CONST, a numeric variable containing the estimates of constant parameters and their standard errors
- SD_i, a numeric variable containing the estimates of seasonal dummy parameters and their standard errors, where $i = 1, \ldots, (nseason - 1)$
- XLl_i, numeric variables containing the estimates of exogenous parameters and their standard errors, where l is the lag lth coefficient matrix and $i = 1, \ldots, r$, r is the number of exogenous variables
- ARl_i, numeric variables containing the estimates of autoregressive parameters and their standard errors, where l is the lag lth coefficient matrix and $i = 1, \ldots, k$, k is the number of endogenous variables
- MAl_i, numeric variables containing the estimates of moving-average parameters and their standard errors, where l is the lag lth coefficient matrix and $i = 1, \ldots, k$, k is the number of endogenous variables

Table 4.3. 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($\phi_{2,22}$)

The OUTEST= data set contains the values shown Table 4.3 for a bivariate case.

Consider the following example:

```
proc varmax data=simul2 outest=est;
   model y1 y2 / p=2 noint noprint
                 ecm=(rank=1 normalize=y1);
run;
proc print data=est;
run;
```

```
Obs    NAME    TYPE        AR1_1        AR1_2        AR2_1        AR2_2

 1      y1     EST      -0.46680      0.91295     -0.74332     -0.74621
 2             STD       0.04786      0.09359      0.04526      0.04769
 3      y2     EST       0.10667     -0.20862      0.40493     -0.57157
 4             STD       0.05146      0.10064      0.04867      0.05128
```

Figure 4.48. OUTEST= Data Set

The output in Figure 4.48 shows the part of results of the OUTEST= data set.

OUTSTAT= Data Set

The OUTSTAT= data set contains estimation results of the fitted model. The following output variables can be created. The sub-index i is $1, \ldots k$, k is the number of endogenous variables.

- the BY variables
- NAME, a character variable containing the name of endogenous (dependent) variables
- SIGMA_i, numeric variables containing the estimate covariance of the innovation covariance matrix
- AICC, a numeric variable containing the corrected Akaike's information criteria value
- RSquare, a numeric variable containing R^2
- FValue, a numeric variable containing the F statistics
- PValue, a numeric variable containing p-value for the F statistics
- EValueI2_i, numeric variables containing eigenvalues for the cointegration rank test of integrated order 2
- EValueI1, a numeric variable containing eigenvalues for the cointegration rank test of integrated order 1
- BETA_i, numeric variables containing $\boldsymbol{\beta}$ long-run effect parameter estimates

- ALPHA_i, numeric variables containing $\boldsymbol{\alpha}$ adjustment parameter estimates
- ETA_i, numeric variables containing $\boldsymbol{\eta}$ parameter estimates in integrated order 2
- Xi_i, numeric variables containing $\boldsymbol{\xi}$ parameter estimates in integrated order 2

Table 4.4. OUTSTAT= Data Set

Obs	NAME	SIGMA_1	SIGMA_2	AICC	RSquare	FValue	PValue
1	y1	σ_{11}	σ_{12}	$aicc$	R_1^2	F_1	$prob_1$
2	y2	σ_{21}	σ_{22}	.	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}

The OUTSTAT= data set contains the values shown Table 4.4 for a bivariate case.

Consider the following example:

```
proc varmax data=simul2 outstat=stat;
   model y1 y2 / p=2 noint
                 cointtest=(johansen=(iorder=2))
                 ecm=(rank=1 normalize=y1) noprint;
run;
proc print data=stat;
run;
```

```
Obs  NAME      SIGMA_1      SIGMA_2         AICC  RSquare    FValue  PValue

 1    y1      94.75575      4.52684      9.37221  0.9391     482.78  0.0000
 2    y2       4.52684    109.57038            .  0.9409     498.42  0.0000

     EValue  EValue  EValue
Obs   I2_1    I2_2     I1         BETA_1       BETA_2      ALPHA_1      ALPHA_2

 1   0.9849  0.9508  0.5086      1.00000      1.00000     -0.46680      0.00794
 2   0.8145   .      0.0111     -1.95575     -1.33622      0.10667      0.03353


Obs        ETA_1          ETA_2           XI_1           XI_2

 1      -0.01231        0.02703       54.16057      -52.31444
 2       0.01555        0.02309      -79.42404      -18.33083
```

Figure 4.49. OUTSTAT= Data Set

The output in Figure 4.49 shows the part of results of the OUTSTAT= data set.

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 estimate method
- the estimates of the constant vector (or seasonal constant matrix), the trend vector, the coefficients matrices of the distributed lags, the AR coefficients matrices, and the MA coefficients matrices
- a table of parameter estimates showing the following for each parameter: the variable name for the left-hand side of equations, 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
- the cross-covariance and cross-correlation matrices of the residuals and 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 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 the following table:

Table 4.5. 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)
AccumImpulsX	Accumulated Transfer Function Matrices	IMPULSX=(ACCUM) IMPULSX=(ALL)
Alpha	α Coefficients	JOHANSEN=
AlphaInECM	α Coefficients	ECM=
AlphaOnDrift	α Coefficients on Restriction of a Deterministic Term	JOHANSEN=
AlphaBetaInECM	$\pi = \alpha\beta'$ Coefficients	ECM=
ARCoef	AR Coefficients	P=
ARRoots	Roots of AR Characteristic Polynomial	ROOTS
Beta	β Coefficients	JOHANSEN=
BetaInECM	β Coefficients	ECM=
BetaOnDrift	β Coefficients on Restriction of a Deterministic Term	JOHANSEN=
Constant	Constant Estimates	default
CorrB	Correlations of Parameter Estimates	CORRB
CorrResiduals	Cross-Correlations of Residuals	default
CorrResidualsGraph	Schematic Representation of Residual Cross-Correlations	default
CorrGraph	Schematic Representation of Sample Cross-Correlations	CORRX CORRY
CorrXLags	Cross-Correlation Matrices of Independent Series	CORRX
CorrYLags	Cross-Correlation Matrices of Dependent Series	CORRY
CovB	Covariance of Parameter Estimates	COVB
CovInnov	Covariance Matrix for the Innovation	default
CovPredError	Covariance Matrices of the Prediction Error	COVPE
CovResiduals	Cross-Covariance Matrices of Residuals	default
CovXLags	Cross-Covariance Matrices of Independent Series	COVX
CovYLags	Cross-Covariance Matrices of Dependent Series	COVY

Table 4.5. (ODS Tables Continued)

ODS Table Name	Description	Option
DecompCovPredError	Decomposition of the Prediction Error Covariance	DECOMPOSE
DFTest	Dickey-Fuller Tests	DFTEST
EigenvalueI2	Eigenvalues in Integrated Order 2	JOHANSEN= (IORDER=2)
Eta	η Coefficients	JOHANSEN= (IORDER=2)
DriftHypo	Hypothesis of Different Deterministic Terms in Cointegration Rank Test	JOHANSEN=
DriftHypoTest	Test Hypothesis of Different Deterministic Terms in Cointegration Rank Test	JOHANSEN=
InfiniteARRepresent	Infinite Order AR Representation	IARR
InfoCriterion	Information criterion	default
LinearTrend	Linear Trend Estimates	TREND=
MaxTest	Cointegration Rank Test Using the Maximum Eigenvalue	JOHANSEN= (TYPE=MAX)
MaxTestOnDrift	Cointegration Rank Test Using the Maximum Eigenvalue on Restriction of a Deterministic Term	JOHANSEN= (TYPE=MAX)
ModelType	Type of Model	default
NObs	Number of Observations	default
OrthoImpulse	Orthogonalized Impulse Response Matrices	IMPULSE=(ORTH) IMPULSE=(ALL)
ParameterEstimates	Parameter Estimates Table	default
PartialAR	Partial Autoregression Matrices	PARCOEF
PartialARGraph	Schematic Representation of Partial Autoregression	PARCOEF
PartialCanCorr	Partial Canonical Correlation Analysis	PCANCORR
PartialCorr	Partial Cross-Correlation Matrices	PCORR
PartialCorrGraph	Schematic Representation of Partial Cross-Correlations	PCORR
PortmanteauTest	Chi-Square Test Table for Residual Cross-Correlations	default
ProportionDecomp	Proportions of Prediction Error Covariance Decomposition	DECOMPOSE
RankTestI2	Cointegration Rank Test in Integrated Order 2	JOHANSEN= (IORDER=2)
QuadTrend	Quadratic Trend Estimates	TREND=QUAD
SConstant	Seasonal Constant Estimates	NSEASON=
SimpleImpulse	Impulse Response Matrices	IMPULSE IMPULSE=(SIMPLE) IMPULSE=(ALL)
SimpleImpulsX	Impulse Response Matrices in Transfer Function	IMPULSX IMPULSX=(SIMPLE) IMPULSX=(ALL)

Table 4.5. (ODS Tables Continued)

ODS Table Name	Description	Option
Summary	Simple Summary Statistics	default
SWTest	Common Trends Test	SW SW=
TentativeOrders	Tentative Order Selection	MINIC MINIC=
TraceTest	Cointegration Rank Test Using the Trace	JOHANSEN= (TYPE=TRACE)
TraceTestOnDrift	Cointegration Rank Test Using the Trace on Restriction of a Deterministic Term	JOHANSEN= (TYPE=TRACE)
UnivarDiagnostAR	Check the AR Disturbance for the residuals	default
UnivarDiagnostCheck	Univariate Model Diagnostic Checks	default
UnivarDiagnostTest	Check the ARCH Disturbance and Normality for the residuals	default
Xi	ξ Coefficient Matrix	JOHANSEN= (IORDER=2)
XLagCoef	Dependent Coefficients	XLAG=
YWEstimates	Yule-Walker Estimates	YW
*ByVariable	Prints by Variable	PRINTFORM=
	ODS Tables Created by the COINTEG Statement	
AlphaInECM	α Coefficients	default
AlphaBetaInECM	$\pi = \alpha\beta'$ Coefficients	default
BetaInECM	β Coefficients	default
AlphaOnTest	α Coefficients under Restriction	H= J=
BetaOnTest	β Coefficients under Restriction	H= J=
RestrictMatrix	Restriction Matrix for α or β	H= J=
RestrictTest	Hypothesis Testing of α or β	H= J=
WeakExogeneity	Testing Weak Exogeneity of each Dependent Variable with respect to BETA	EXOGENEITY
	ODS Tables Created by the CAUSAL Statement	
Causality	Granger-Causality Test	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	PRINT

Note that the symbol * corresponds to AccumImpulse, AccumImpulsX, CorrResiduals, CorrXLags, CorrYLags, CovResiduals, CovXLags, CovYLags, OrthoImpulse, PartialCorr, PredictMSE, DecompCovPredError, ProportionDecomp, SimpleImpulse, and SimpleImpulsX.

Examples

Example 4.1. Analysis of Real Output Series

Consider the following four-dimensional system of U.S. economic variables. Quarterly data for the years 1954 to 1987 are used (Lütkepohl 1993, Table E.3.).

```
symbol1 v=none height=1 c=black;
symbol2 v=none height=1 c=black;

   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;
      ... data lines omitted ...
      ;

   legend1 across=1 frame label=none;

   proc gplot data=us_money;
      symbol1 i = join l = 1;
      symbol2 i = join l = 2;
      axis2 label = (a=-90 r=90 " ");
      plot y1 * date = 1 y2 * date = 2 /
           overlay vaxis=axis2 legend=legend1;
   run;

   proc gplot data=us_money;
      symbol1 i = join l = 1;
      symbol2 i = join l = 2;
      axis2 label = (a=-90 r=90 " ");
      plot y3 * date = 1 y4 * date = 2 /
           overlay vaxis=axis2 legend=legend1;
   run;

   proc varmax data=us_money;
      id date interval=qtr;
      model y1-y4 / p=2 lagmax=6 dftest
                    print=(iarr)
                    cointtest=(johansen=(iorder=2))
                    ecm=(rank=1 normalize=y1);
     cointeg rank=1 normalize=y1 exogeneity;
   run;
   quit;
```

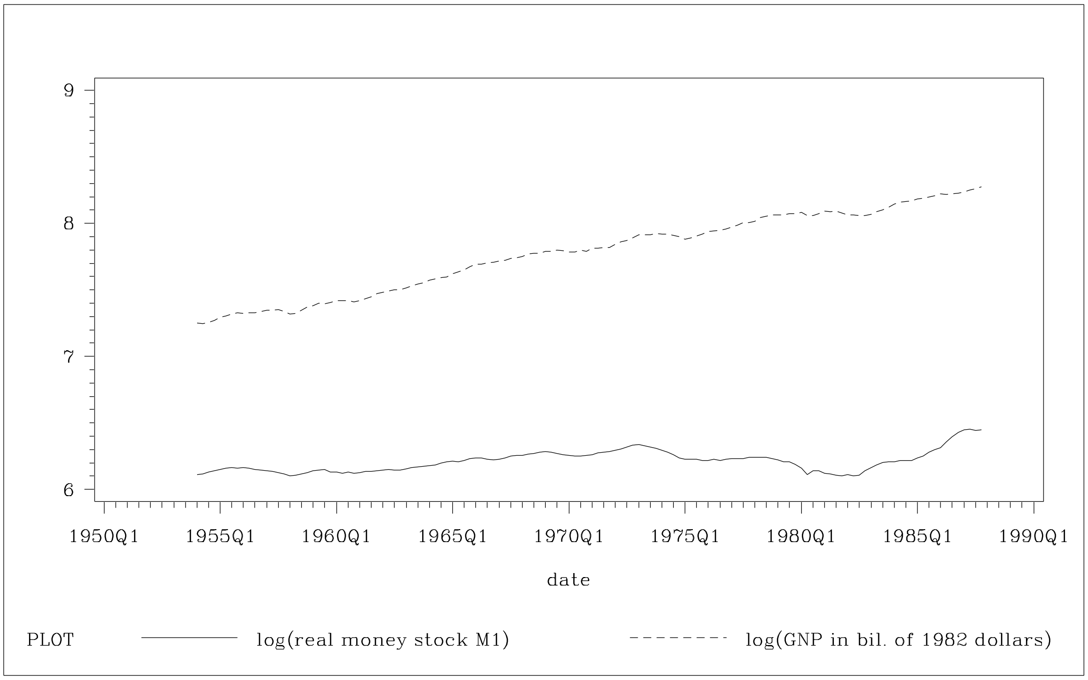

Figure 4.50. Plot of Data

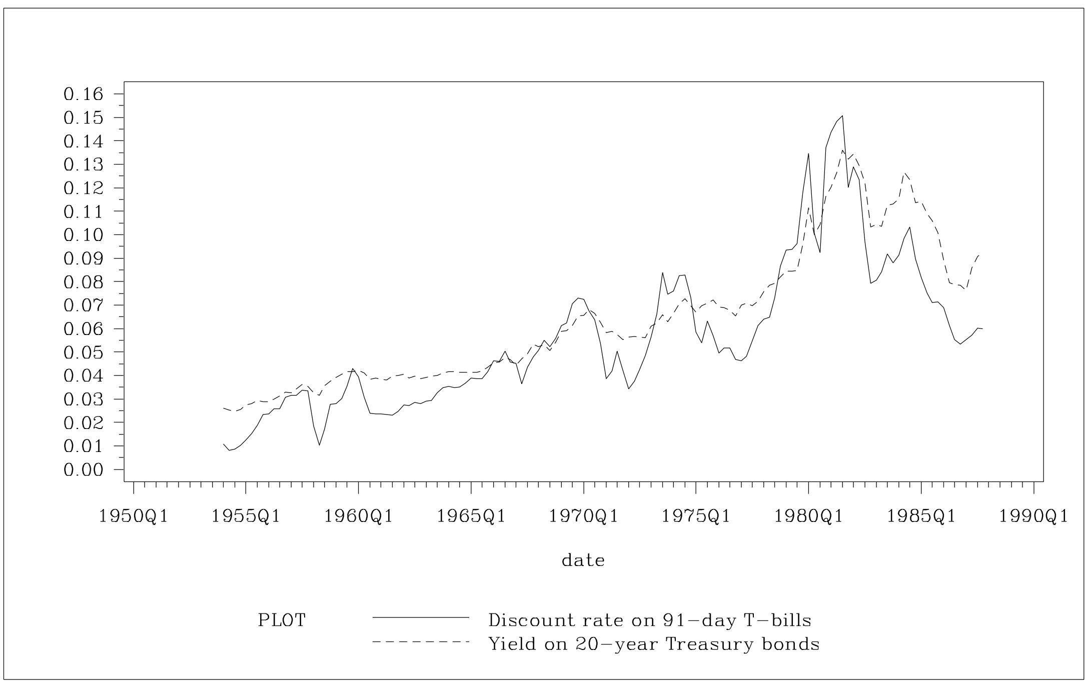

Figure 4.51. Plot of Data Continued

The VARMAX procedure output is shown in Output 4.1.1 through Output 4.1.10.

Output 4.1.1. Descriptive Statistics

The VARMAX Procedure

Number of Observations	136
Number of Pairwise Missing	0

Variable	Type	NoMissN	Mean	StdDev	Min	Max
y1	DEP	136	6.21295	0.07924	6.10278	6.45331
y2	DEP	136	7.77890	0.30110	7.24508	8.27461
y3	DEP	136	0.05608	0.03109	0.00813	0.15087
y4	DEP	136	0.06458	0.02927	0.02490	0.13600

Variable	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

Output 4.1.2. Unit Root Tests

The VARMAX Procedure

Dickey-Fuller Unit Root Tests

Variable	Type	Rho	Prob<Rho	Tau	Prob<Tau
y1	Zero Mean	0.051968	0.6934	1.14	0.9343
	Single Mean	-2.96706	0.6572	-0.76	0.8260
	Trend	-5.90836	0.7454	-1.34	0.8725
y2	Zero Mean	0.132569	0.7124	5.14	0.9999
	Single Mean	-0.43009	0.9309	-0.79	0.8176
	Trend	-9.20862	0.4787	-2.16	0.5063
y3	Zero Mean	-1.28445	0.4255	-0.69	0.4182
	Single Mean	-8.85767	0.1700	-2.27	0.1842
	Trend	-18.9729	0.0742	-2.86	0.1803
y4	Zero Mean	0.403766	0.7803	0.45	0.8100
	Single Mean	-2.79092	0.6790	-1.29	0.6328
	Trend	-12.1185	0.2923	-2.33	0.4170

Output 4.1.3. Cointegration Rank Test

The VARMAX Procedure

Eigenvalues of Cointegration Rank Test for I(2)

r\s	0	1	2	3	Eigenvalue_I1
0	0.71927	0.54812	0.41092	0.24142	0.231643
1	0.62213	0.36989	0.18448		0.125740
2	0.31902	0.15225			0.019455
3	0.24857				0.000111

Long-Run Parameter BETA Estimates

Variable	Dummy 1	Dummy 2	Dummy 3	Dummy 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

Adjustment Coefficient ALPHA Estimates

Variable	Dummy 1	Dummy 2	Dummy 3	Dummy 4
y1	-0.01396	0.01396	-0.01119	0.00007569
y2	-0.02811	-0.02739	-0.00031751	0.00076370
y3	-0.00215	-0.04967	-0.00183	-0.00072255
y4	0.00510	-0.02514	-0.00220	0.00015940

Parameter ETA Estimates

Variable	Dummy 1	Dummy 2	Dummy 3	Dummy 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

Parameter XI Estimates

Variable	Dummy 1	Dummy 2	Dummy 3	Dummy 4
y1	-0.00842	-0.00052033	-0.00208	-0.00250
y2	0.00141	0.00213	-0.00736	-0.00057797
y3	-0.00445	0.00541	-0.00150	0.00310
y4	-0.00211	-0.00063722	-0.00130	0.00197

Output 4.1.4. Parameter Estimates

The VARMAX Procedure

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

Long-Run Parameter
BETA Estimates
given RANK = 1

Variable	Dummy 1
y1	1.00000
y2	-0.46458
y3	14.51619
y4	-9.35520

Adjustment Coefficient
ALPHA Estimates
given RANK = 1

Variable	Dummy 1
y1	-0.01396
y2	-0.02811
y3	-0.00215
y4	0.00510

Output 4.1.5. Parameter Estimates Continued

The VARMAX Procedure

Constant Estimates

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	0.00099887	-0.03121	0.02011
y4	0.00510	-0.00237	0.07407	-0.04774

AR Coefficient Estimates

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 4.1.6. Parameter Estimates Continued

The VARMAX Procedure

Model Parameter Estimates

Equation	Parameter	Estimate	Std Error	T Ratio	Prob>|T|	Variable
D_y1(t)	CONST1	0.04076	0.01418	2.87	0.0048	
	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	0.0001	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	0.0001	D_y4(t-1)
D_y2(t)	CONST2	0.08595	0.01679	5.12	0.0001	
	AR1_2_1	-0.02811	0.00586	-4.79	0.0001	y1(t-1)
	AR1_2_2	0.01306	0.00272	4.79	0.0001	y2(t-1)
	AR1_2_3	-0.40799	0.08514	-4.79	0.0001	y3(t-1)
	AR1_2_4	0.26294	0.05487	4.79	0.0001	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(t)	CONST3	0.00518	0.01608	0.32	0.7476	
	AR1_3_1	-0.00215	0.00562	-0.38	0.7024	y1(t-1)
	AR1_3_2	0.00099887	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(t)	CONST4	-0.01438	0.00803	-1.79	0.0758	
	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 4.1.7. Diagnostic Checks

The VARMAX Procedure

Covariance Matrix for the Innovation

Variable	y1	y2	y3	y4
y1	0.00005072	0.00001486	-0.00000803	-0.00000326
y2	0.00001486	0.00007110	0.00001636	0.00001008
y3	-0.00000803	0.00001636	0.00006517	0.00002296
y4	-0.00000326	0.00001008	0.00002296	0.00001627

Information Criteria

AICC(Corrected AIC)	-40.6284
HQC(Hannan-Quinn Criterion)	-40.4343
AIC(Akaike Information Criterion)	-40.6452
SBC(Schwarz Bayesian Criterion)	-40.1262
FPEC(Final Prediction Error Criterion)	2.23E-18

Schematic Representation of Residual Cross Correlations

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 Residual Cross Correlations

To Lag	Chi-Square	DF	Prob> ChiSq
3	53.90	16	<.0001
4	74.03	32	<.0001
5	103.08	48	<.0001
6	116.94	64	<.0001

Output 4.1.8. Diagnostic Checks Continued

The VARMAX Procedure

Univariate Model Diagnostic Checks

Variable	R-square	StdDev	F Value	Prob>F
y1	0.6754	0.0071	32.51	<.0001
y2	0.3070	0.0084	6.92	<.0001
y3	0.1328	0.0081	2.39	0.0196
y4	0.0831	0.0040	1.42	0.1963

Univariate Model Diagnostic Checks

Variable	DW(1)	Normality ChiSq	Prob> ChiSq	ARCH1 F Value	Prob>F
y1	2.13	7.19	0.0275	1.62	0.2053
y2	2.04	1.20	0.5483	1.23	0.2697
y3	1.87	253.76	<.0001	1.78	0.1847
y4	1.98	105.21	<.0001	21.01	<.0001

Univariate Model Diagnostic Checks

Variable	AR1 F Value	Prob>F	AR1-2 F Value	Prob>F	AR1-3 F Value	Prob>F	AR1-4 F Value	Prob>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

Output 4.1.9. Infinite Order AR Representation

The VARMAX Procedure

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

Output 4.1.10. Weak Exogeneity Test

```
                         The VARMAX Procedure

                      Tests of Weak Exogeneity
                        of Each of Variables

                              Chi-               Prob>
              Variable      Square      DF      ChiSq

              y1              6.55       1     0.0105
              y2             12.54       1     0.0004
              y3              0.09       1     0.7695
              y4              1.81       1     0.1786
```

Example 4.2. Analysis of Real Output Series

This example considers a three-dimensional VAR(2) model. The model contains the logarithms of quarterly seasonally adjusted West German fixed investment, disposable income, and consumption expenditures. The data used are in Lütkepohl (1993, Table E.1).

```
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;
   ... data lines omitted ...
   ;

data use;
   set west;
   where  date < '01jan79'd;
   keep date y1 y2 y3;

proc varmax data=use;
   id date interval=qtr align=E;
   model y1-y3 / p=2 dify=(1)
                 print=(decompose(6) impulse=(stderr))
                 printform=both lagmax=3;
   causal group1=(y1) group2=(y2 y3);
   output lead=5;
run;
quit;
```

The VARMAX procedure output is shown in Output 4.2.1 through Output 4.2.7.

Output 4.2.1. Descriptive Statistics

```
                              The VARMAX Procedure

                Number of Observations                         75
                Number of Pairwise Missing                      0
                Observation(s) eliminated by differencing       1


Variable Type  NoMissN       Mean     StdDev        Min        Max Difference

y1       DEP        75    0.01811    0.04680   -0.14018    0.19358          1
y2       DEP        75    0.02071    0.01208   -0.02888    0.05023          1
y3       DEP        75    0.01987    0.01040   -0.01300    0.04483          1

                       Variable Label

                       y1       logarithm of investment
                       y2       logarithm of income
                       y3       logarithm of consumption
```

Output 4.2.2. Parameter Estimates

The VARMAX Procedure

Type of Model	VAR(2)
Estimation Method	Least Squares Estimation

Constant Estimates

Variable	Constant
y1	-0.01672
y2	0.01577
y3	0.01293

AR Coefficient Estimates

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

Model Parameter Estimates

Equation	Parameter	Estimate	Std Error	T Ratio	Prob>|T|	Variable
y1(t)	CONST1	-0.01672	0.01723	-0.97	0.3352	
	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(t)	CONST2	0.01577	0.00437	3.60	0.0006	
	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(t)	CONST3	0.01293	0.00353	3.67	0.0005	
	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 4.2.3. Diagnostic Checks

```
                         The VARMAX Procedure

                Covariance Matrix for the Innovation

       Variable             y1             y2             y3

       y1              0.00213     0.00007162     0.00012324
       y2           0.00007162     0.00013734     0.00006146
       y3           0.00012324     0.00006146     0.00008920


                         Information Criteria

         AICC(Corrected AIC)                      -24.4884
         HQC(Hannan-Quinn Criterion)              -24.2869
         AIC(Akaike Information Criterion)        -24.5494
         SBC(Schwarz Bayesian Criterion)          -23.8905
         FPEC(Final Prediction Error Criterion)    2.18E-11


                  Residual Cross-Correlation Matrices

    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
                     Residual Cross Correlations

                Variable/
                Lag            0      1      2      3

                y1           +.+    ...    ...    ...
                y2           .++    ...    ...    ..+
                y3           +++    ...    ...    ...

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


                    Portmanteau Test for Residual
                          Cross Correlations
                   To         Chi-            Prob>
                   Lag      Square     DF     ChiSq

                     3        9.69      9    0.3766
```

Output 4.2.4. Diagnostic Checks Continued

The VARMAX Procedure

Univariate Model Diagnostic Checks

Variable	R-square	StdDev	F Value	Prob>F
y1	0.1286	0.0461	1.62	0.1547
y2	0.1142	0.0117	1.42	0.2210
y3	0.2513	0.0094	3.69	0.0032

Univariate Model Diagnostic Checks

Variable	DW(1)	Normality ChiSq	Prob> ChiSq	ARCH1 F Value	Prob>F
y1	1.96	10.22	0.0060	12.39	0.0008
y2	1.98	11.98	0.0025	0.38	0.5386
y3	2.15	34.25	<.0001	0.10	0.7480

Univariate Model Diagnostic Checks

Variable	AR1 F Value	Prob>F	AR1-2 F Value	Prob>F	AR1-3 F Value	Prob>F	AR1-4 F Value	Prob>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 4.2.5. Impulse Response Function

The VARMAX Procedure

Impulse Response by Variable

Variable	Lead	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	-0.00055044	-0.09818	0.09096
	STD	0.01646	0.07823	0.10280

Output 4.2.6. Proportions of Prediction Error Covariance Decomposition

The VARMAX Procedure

Proportions of Prediction Error Covariances by Variable

Variable	Lead	y1	y2	y3
y1	1	1.00000	0	0
	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
	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

Output 4.2.7. Forecasts

The VARMAX Procedure

Forecasts

Variable	Obs	Time	Forecast	Standard Error	95% Confidence Limits	
y1	77	1979:1	6.5403	0.0461	6.4498	6.6307
	78	1979:2	6.5511	0.0583	6.4369	6.6652
	79	1979:3	6.5722	0.0688	6.4373	6.7071
	80	1979:4	6.5845	0.0802	6.4273	6.7417
	81	1980:1	6.6019	0.0912	6.4232	6.7806
y2	77	1979:1	7.6847	0.0117	7.6618	7.7077
	78	1979:2	7.7051	0.0169	7.6719	7.7382
	79	1979:3	7.7221	0.0216	7.6798	7.7643
	80	1979:4	7.7427	0.0262	7.6914	7.7939
	81	1980:1	7.7624	0.0301	7.7035	7.8213
y3	77	1979:1	7.5402	0.00944	7.5217	7.5587
	78	1979:2	7.5549	0.0128	7.5298	7.5800
	79	1979:3	7.5747	0.0181	7.5393	7.6101
	80	1979:4	7.5934	0.0221	7.5502	7.6367
	81	1980:1	7.6123	0.0258	7.5618	7.6629

Output 4.2.8. Granger Causality Tests

```
                          The VARMAX Procedure

                      Granger Causality Wald Test

                              Chi-             Prob>
                   Test      Square     DF     Chisq

                      1        6.37      4    0.1734


                    Test 1:  Group 1 Variables:  y1
                             Group 2 Variables:  y2 y3
```

```
proc varmax data=use;
   id date interval=qtr align=E;
   model y2 y3 = y1 / p=2 dify(1) difx(1) xlag=1
                       lagmax=3;
run;
```

The VARMAX procedure output is shown in Output 4.2.9 through Output 4.2.12.

Output 4.2.9. Parameter Estimates

```
                          The VARMAX Procedure

           Type of Model                       VARX(2,1)
           Estimation Method    Least Squares Estimation


                          Constant Estimates

                         Variable       Constant

                         y2              0.01542
                         y3              0.01319


                        XLag Coefficient Estimates

                       Lag    Variable             y1

                         0    y2              0.02520
                              y3              0.05130
                         1    y2              0.03870
                              y3              0.00363


                         AR Coefficient Estimates

           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 4.2.10. Parameter Estimates Continued

The VARMAX Procedure

Model Parameter Estimates

Equation	Parameter	Estimate	Std Error	T Ratio	Prob>|T|	Variable
y2(t)	CONST1	0.01542	0.00443	3.48	0.0009	
	XL0_1	0.02520	0.03130	0.81	0.4237	y1(t)
	XL1_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(t)	CONST2	0.01319	0.00346	3.81	0.0003	
	XL0_2	0.05130	0.02441	2.10	0.0394	y1(t)
	XL1_2	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 4.2.11. Diagnostic Checks

```
                     The VARMAX Procedure

             Covariance Matrix for the Innovation

            Variable             y2             y3

            y2           0.00014113     0.00006214
            y3           0.00006214     0.00008583


                     Information Criteria

      AICC(Corrected AIC)                      -18.3902
      HQC(Hannan-Quinn Criterion)              -18.2558
      AIC(Akaike Information Criterion)        -18.4309
      SBC(Schwarz Bayesian Criterion)          -17.9916
      FPEC(Final Prediction Error Criterion)    9.91E-9


             Residual Cross-Correlation Matrices

        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
                 Residual Cross Correlations

              Variable/
              Lag           0     1     2     3

              y2           ++    ..    ..    .+
              y3           ++    ..    ..    ..

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


                Portmanteau Test for Residual
                      Cross Correlations
               To         Chi-             Prob>
               Lag      Square     DF     ChiSq

                 3        8.38      4    0.0787
```

Output 4.2.12. Diagnostic Checks Continued

The VARMAX Procedure

Univariate Model Diagnostic Checks

Variable	R-square	StdDev	F Value	Prob>F
y2	0.0897	0.0119	1.08	0.3809
y3	0.2796	0.0093	4.27	0.0011

Univariate Model Diagnostic Checks

Variable	DW(1)	Normality ChiSq	Prob> ChiSq	ARCH1 F Value	Prob>F
y2	2.02	14.54	0.0007	0.49	0.4842
y3	2.13	32.27	<.0001	0.08	0.7782

Univariate Model Diagnostic Checks

Variable	AR1 F Value	Prob>F	AR1-2 F Value	Prob>F	AR1-3 F Value	Prob>F	AR1-4 F Value	Prob>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 4.3. 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;
  quit;

/* 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 prediction */
  proc varmax data=a;
     model y1 y2 / p=1 lagmax=3
                   print=(impulse covpe(5) decompose(5));
  run;

/* Options related 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 Estimator */
  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 nrep=200);
  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) seed=12345);
  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)
                   ecm=(rank=3 normalize=y1);
  run;
```

```
/* Weak Exogenous Testing for each variable */
  proc varmax data=b outstat=bbb;
     model y1-y4 / p=2 lagmax=4
                   ecm=(rank=3 normalize=y1);
     cointeg rank=3 exogeneity;
  run;

/* Hypotheses Testing for long-run and adjustment parameter */
  proc varmax data=b outstat=bbb;
     model y1-y4 / p=2 lagmax=4
                   ecm=(rank=3 normalize=y1);
     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 time Exogenous Variables */
  proc varmax data=grunfeld;
     model y1 y2 = x1 / p=1 xlag=2 difx=(1) dify=(1);
  run;
```

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 Kohn, R. (1986), “A Note on Reparameterizing a Vector Autoregressive Moving Average Model to Enforce Stationarity”, *Journal of Statistical Computation and Simulation*, 24, 99–106.

Ansley, C. F., Kohn, R., and Shively, T. S. (1992), “Computing *p*-values for the Generalized Durbin-Watson and Other Invariant Test Statistics,” *Journal of Econo-*

metrics, 54, 277–300.

Brockwell, P. J. and Davis, R. A. (1991), *Time Series: Theory and Methods*, Second Edition, New York: Springer-Verlag.

Dickey, D. A., Hasza, D. P., and Fuller, W. A. (1984), "Testing for Unit Roots in Seasonal Time Series," *Journal of the American Statistical Association*, 79, 355–367.

Doan, T., Litterman, R., and Sims, C. (1984), "Forecasting and Conditional Projection Using Realistic Prior Distributions," *Econometric Reviews*, 3, 1–144.

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 Yoo, B. S. (1987), "Forecasting and Testing in Co-Integrated Systems," *Journal of Econometrics*, 35, 143–159.

Hamilton, J. D. (1994), *Time Series Analysis*, Princeton: Princeton University Press.

Hannan, E. J. (1970), *Multiple Time Series*, New York: John Wiley & Sons.

Hannan, E. J. and Deistler, M. (1988), *The Statistical Theory of Linear Systems*, New York: John Wiley & Sons.

Hosking, J. R. M. (1980), "The Multivariate Portmanteau Statistic," *Journal of the American Statistical Association*, 75, 602–608.

Hurvichm, C. M. and Tsai, C. (1993), "A Corrected Akaike Information Criterion for Vector Autoregressive Model Selection," *Journal of Time Series Analysis*, 14, 271–279.

Johansen, S. (1988), "Statistical Analysis of Cointegration Vectors," *Journal of Economic Dynamics and Control*, 12, 231–254.

Johansen, S. (1992a), "A Representation of Vector Autoregressive Processes Integrated of Order 2," *Econometric Theory*, 8, 188–202.

Johansen, S. (1992b), "Testing Weak Exogeneity and the Order of Cointegration in UK Money Demand Data," *Journal of Policy Modeling*, 14, 313–334.

Johansen, S. (1995a), "A Statistical Analysis of Cointegration for I(2) Variables," *Econometric Theory*, 11, 25–59.

Johansen, S. (1995b), *Likelihood-Based Inference in Cointegrated Vector Autoregressive Models*, New York: Oxford University Press.

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.

Johansen, S. and Juselius, K. (1992), "Testing Structural Hypotheses in a Multivariate Cointegration Analysis of the PPP and the UIP for UK," *Journal of Econometrics*, 53, 211–244.

Kim, M. (1996), "A Remark on Algorithm AS 279: Computing p-values for the Generalized Durbin-Watson Statistic and Residual Autocorrelation in Regression," *Applied Statistics*, 45, 273–274.

Kohn, R., Shively, T. S., and Ansley, C. F. (1993), "Algorithm AS 279: Computing *p*-values for the Generalized Durbin-Watson Statistic and Residual Autocorrelation in Regression," *Applied Statistics*, 42, 249–269.

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 & Economic Statistics*, 4, 25–38.

Lütkepohl, H. (1993), *Introduction to Multiple Time Series Analysis*, Berlin: Springer-Verlag.

Maddala, G. S. and Kim, I. (1998), *Unit Roots, Cointegration, and Structural Change*, Cambridge: Cambridge University Press.

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 Raynor, D. L. (1971), *Generalized Inverse Matrices with Applications to Statistics*, Second Edition, New York: McGraw-Hill Inc.

Reinsel, G. C. (1997), *Elements of Multivariate Time Series Analysis*, Second Edition, New York: Springer-Verlag.

Quinn, B. G. (1980), "Order Determination for a Multivariate Autoregression," *Journal of the Royal Statistical Society*, B, 42, 182–185.

Spliid, H. (1983), "A Fast Estimation for the Vector Autoregressive Moving Average Models 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.

Todd, R. M. (1984), "Improving Economic Forecasting with Bayesian Vector Autoregression," *Federal Reserve Bank of Minneapolis Quarterly Review*, 4 (Fall), 18–29.

Chapter 5
The X12 Procedure

Chapter Table of Contents

Chapter 5
The X12 Procedure

Overview

The X12 procedure, an adaptation of the U.S. Bureau of the Census X-12-ARIMA Seasonal Adjustment program, seasonally adjusts monthly or quarterly time series. The procedure makes additive or multiplicative adjustments and creates an output data set containing the adjusted time series and intermediate calculations.

The X-12-ARIMA program combines the capabilities of the X-11 program (Shiskin, Young, and Musgrave 1967) and the X-11-ARIMA/88 program (Dagum 1988) and also introduces some new features (Findley et al. 1988). Thus, the X-12-ARIMA program contains methods developed by both the U.S. Census Bureau and Statistics Canada. The four major components of the X-12-ARIMA program are regARIMA modeling, model diagnostics, seasonal adjustment using enhanced X-11 methodology, and post-adjustment diagnostics. Statistics Canada's X-11 method fits an ARIMA model to the original series, then uses the model forecast to extend the original series. This extended series is then seasonally adjusted by the standard X-11 seasonal adjustment method. The extension of the series improves the estimation of the seasonal factors and reduces revisions to the seasonally adjusted series as new data become available.

Seasonal adjustment of a series is based on the assumption that seasonal fluctuations can be measured in the original series, O_t, t = 1, . . . ,n, and separated from trend cycle, trading-day, and irregular fluctuations. The seasonal component of this time series, S_t, is defined as the intrayear variation that is repeated constantly or in an evolving fashion from year to year. The trend cycle component, C_t, includes variation due to the long-term trend, the business cycle, and other long-term cyclical factors. The trading-day component, D_t, is the variation that can be attributed to the composition of the calendar. The irregular component, I_t, is the residual variation. Many economic time series are related in a multiplicative fashion ($O_t = S_t C_t D_t I_t$). Other economic series are related in an additive fashion ($O_t = S_t + C_t + D_t + I_t$). A seasonally adjusted time series, $C_t I_t$ or $C_t + I_t$, consists of only the trend cycle and irregular components.

Getting Started

The most common use of the X12 procedure is to produce a seasonally adjusted series. Eliminating the seasonal component from an economic series facilitates comparison among consecutive months or quarters. A plot of the seasonally adjusted series is often more informative about trends or location in a business cycle than a plot of the unadjusted series.

The following example shows how to use PROC X12 to produce a seasonally adjusted series, $C_t I_t$, from an original series $O_t = S_t C_t D_t I_t$.

In the multiplicative model, the trend cycle component C_t keeps the same scale as the original series O_t, while S_t, D_t, and I_t vary around 1.0. In all displayed tables, these latter components are expressed as percentages and thus vary around 100.0 (in the additive case, they vary around 0.0). However, in the output data set, the data displayed as percentages will be expressed as the decimal equivalent and thus will vary around 1.0 in the multiplicative case.

The naming convention used in PROC X12 for the tables follows the convention used in the Census Bureau's X-12-ARIMA program; refer to *X-12-ARIMA Reference Manual* and *X-12-ARIMA Quick Reference for Unix* (U.S. Bureau of the Census 1999). Also see the section "Displayed Output" later in this chapter. The table names are outlined in Figure 5.5 on page 177.

The tables corresponding to parts A through C are intermediate calculations. The final estimates of the individual components are found in the D tables: table D10 contains the final seasonal factors, table D12 contains the final trend cycle, and table D13 contains the final irregular series. If you are primarily interested in seasonally adjusting a series without consideration of intermediate calculations or diagnostics, you need only to look at table D11, the final seasonally adjusted series. Tables in part E contain information regarding extreme values and changes in the original and seasonally adjusted series. The tables in part F are seasonal adjustment quality measures. Spectral analysis is performed in part G. For further information concerning the tables produced by the X11 statement, refer to Ladiray and Quenneville (1999).

Basic Seasonal Adjustment

Suppose that you have monthly retail sales data starting in September 1978 in a SAS data set named SALES. At this point, you do not suspect that any calendar effects are present, and there are no prior adjustments that need to be made to the data.

In this simplest case, you need only specify the DATE= variable in the PROC X12 statement and request seasonal adjustment in the X11 statement. The results of the seasonal adjustment are in table D11 (the final seasonally adjusted series) in the displayed output.

```
data sales;
   input sales @@;
   date = intnx( 'month', '01sep78'd, _n_-1 );
   format date monyy.;
   datalines;
   ... datalines omitted ...
run;

proc x12 data=sales date=date;
   var sales;
   x11;
run ;
```

```
                           The X12 Procedure

         Table D 11:  Final seasonally adjusted data
                      For variable sales
Year      JAN      FEB      MAR      APR      MAY      JUN
          JUL      AUG      SEP      OCT      NOV      DEC    Total

1978       .        .        .        .        .        .
           .        .     124.560  124.649  124.920  129.002  503.131
1979  125.087  126.759  125.252  126.415  127.012  130.041
      128.056  129.165  127.182  133.847  133.199  135.847  1547.86
1980  128.767  139.839  143.883  144.576  148.048  145.170
      140.021  153.322  159.128  161.614  167.996  165.388  1797.75
1981  175.984  166.805  168.380  167.913  173.429  175.711
      179.012  182.017  186.737  197.367  183.443  184.907  2141.71
1982  186.080  203.099  193.386  201.988  198.322  205.983
      210.898  213.516  213.897  218.902  227.172  240.453  2513.69
1983  231.839  224.165  219.411  225.907  225.015  226.535
      221.680  222.177  222.959  212.531  230.552  232.565  2695.33
1984  237.477  239.870  246.835  242.642  244.982  246.732
      251.023  254.210  264.670  266.120  266.217  276.251  3037.03
1985  275.485  281.826  294.144  286.114  293.192  296.601
      293.861  309.102  311.275  319.239  319.936  323.663  3604.44
1986  326.693  330.341  330.383  330.792  333.037  332.134
      336.444  341.017  346.256  350.609  361.283  362.519  4081.51
1987  364.951  371.274  369.238  377.242  379.413  376.451
      378.930  375.392  374.940  373.612  368.753  364.885  4475.08
1988  371.618  383.842  385.849  404.810  381.270  388.689
      385.661  377.706  397.438  404.247  414.084  416.486  4711.70
1989  426.716  419.491  427.869  446.161  438.317  440.639
      450.193  454.638  460.644  463.209  427.728  485.386  5340.99
1990  477.259  477.753  483.841  483.056  481.902  499.200
      484.893  485.245     .        .        .        .     3873.15
-------------------------------------------------------------------
Avg   277.330  280.422  282.373  286.468  285.328  288.657
      288.389  291.459  265.807  268.829  268.774  276.446

            Total:  40323  Mean:  280.02  S.D.:  111.31
                            Min:   124.56  Max:   499.2
```

Figure 5.1. Basic Seasonal Adjustment

You can compare the original series (table A1) and the final seasonally adjusted series (table D11) by plotting them together. These tables are requested in the OUTPUT statement and are written to the OUT= data set. Note that the default variable name used in the output data set is the input variable name followed by an underscore and the corresponding table name.

```
proc x12 data=sales date=date;
   var sales;
   x11;
   output out=out a1 d11;
run;

symbol1 i=join v='star';
symbol2 i=join v='circle';
legend1 label=none value=('original' 'adjusted');

proc gplot data=out;
   plot sales_A1    * date = 1
```

```
          sales_D11   * date = 2 / overlay legend=legend1;
run;
```

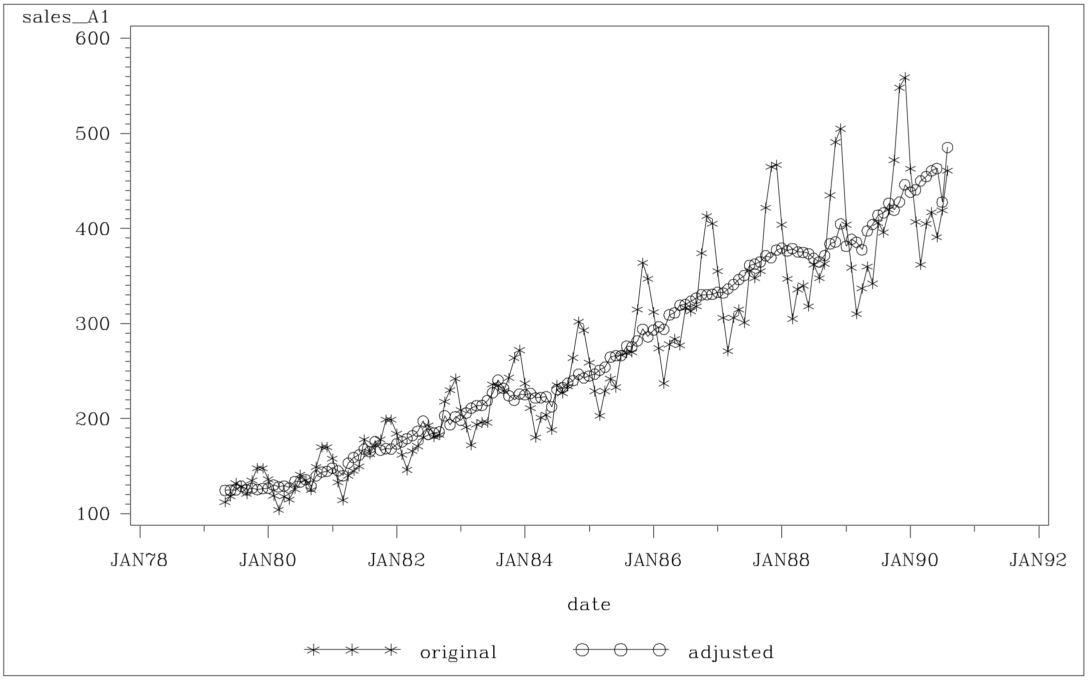

Figure 5.2. Plot of Original and Seasonally Adjusted Data

Syntax

The X12 procedure uses the following statements:

PROC X12 *options*;
  TRANSFORM *options*;
  ESTIMATE;
  IDENTIFY *options*;
  REGRESSION *options*;
  ARIMA *options*;
  X11 *options*;
  VAR *variables*;
  OUTPUT *options*;

The PROC X12 statements perform basically the same function as the Census Bureau's X-12-ARIMA specs. *Specs* or specifications are used in X-12-ARIMA to control the computations and output. The PROC X12 statement performs some of the same functions as the Series spec in the Census Bureau's X-12-ARIMA software. The TRANSFORM, ESTIMATE, IDENTIFY, REGRESSION, ARIMA, and X11 statements are designed to perform the same functions as the corresponding

X-12-ARIMA *specs*, although full compatibility is not yet available. The Census Bureau documentation *X-12-ARIMA Reference Manual* can provide added insight to the functionality of these statements.

Functional Summary

The following table outlines the options available for the X12 procedure classified by function.

Description	Statement	Option
Data Set Options		
specify input data set	PROC X12	DATA=
write table values to an output data set	OUTPUT	OUT=
Printing Control Options		
suppress all printed output	PROC X12	NOPRINT
Date Information Options		
specify the date variable	PROC X12	DATE=
specify the date of the first observation	PROC X12	START=
specify the beginning date of the subset	PROC X12	SPAN=(mmmyy)
	PROC X12	SPAN=(’yyQq’)
specify the ending date of the subset	PROC X12	SPAN=(, mmmyy)
	PROC X12	SPAN=(, ’yyQq’)
specify monthly time series	PROC X12	INTERVAL=MONTH
specify monthly time series	PROC X12	SEASONS=12
specify quarterly time series	PROC X12	INTERVAL=QTR
specify quarterly time series	PROC X12	SEASONS= 4
Declaring the Role of Variables		
specify the variables to be seasonally adjusted	VAR	
Controlling the Table Computations		
transform or prior adjust the series	TRANSFORM	POWER=
estimate the regARIMA model specified by the REGRESSION and ARIMA statements	ESTIMATE	
use differencing to identify the ARIMA part of the model	IDENTIFY	
specify regression information	REGRESSION	PREDEFINED=
specify the ARIMA part of the model	ARIMA	MODEL=
specify seasonal adjustment	X11	

PROC X12 Statement

PROC X12 *options;*

The PROC X12 statement provides information about the time series to be processed by PROC X12. Either the START= or the DATE= option must be specified.

The original series is displayed in table A1. If there are missing values in the original series, and a regARIMA model is specified or automatically selected, then table MV1 is displayed. Table MV1 contains the original series with missing values replaced by the predicted values from the fitted model.

The following options can appear in the PROC X12 statement.

DATA= *SAS-data-set*
specifies the input SAS data set used. If this option is omitted, the most recently created SAS data set is used.

DATE= *variable*
DATEVAR= *variable*
specifies a variable that gives the date for each observation. Unless specified in the SPAN= option, the starting and ending dates are obtained from the first and last values of the DATE= variable, which must contain SAS datetime values. The procedure checks values of the DATE= variable to ensure that the input observations are sequenced correctly in ascending order. If the INTERVAL= or SEASONS= option is specified, its values must agree with the values of the date variable. If neither the INTERVAL= or SEASONS= option is specified, then the procedure tries to determine the type of data from the values of the date variable. This variable is automatically added to the OUTPUT= data set if one is requested and is extrapolated if necessary. If the DATE= option is not specified, the START= option must be specified.

START= *mmmyy*
START= *'yyQq'*
STARTDATE= *mmmyy*
STARTDATE= *'yyQq'*
gives the date of the first observation. Unless the SPAN= option is used, the starting and ending dates are the dates of the first and last observations, respectively. Either this option or the DATE= option is required. When using this option, use either the INTERVAL= or SEASONS= option to specify monthly or quarterly data. If neither the INTERVAL= or SEASONS= is present, monthly data are assumed. Note that for a quarterly date, the specification must be enclosed in quotes. A 4-digit year can be specified, but if a 2-digit year is given, the value specified in the YEARCUTOFF= SAS system option applies.

SPAN= (*mmmyy***,***mmmyy***)**
SPAN= (*'yyQq'***,***'yyQq'***)**
gives the dates of the first and last observations to define a subset for processing. A single date in parentheses is interpreted to be the starting date of the subset. To specify only the ending date, use SPAN=(,*mmmyy*). If the starting or ending date is

omitted, then the first or last date, respectively, of the input data set is assumed. A 4-digit year can be specified, but if a 2-digit year is given, the value specified in the YEARCUTOFF= SAS system option applies.

INTERVAL= *interval*

specifies the frequency of the input time series. If the input data consist of quarterly observations, then INTERVAL=QTR should be used. If the input data consist of monthly observations, then INTERVAL=MONTH should be used. If the INTERVAL= option is not specified and SEASONS=4, then INTERVAL=QTR is assumed; likewise, SEASONS=12 implies INTERVAL=MONTH. If both are used, the values should not be conflicting. If neither the INTERVAL= nor SEASONS= option is specified and the START= option is specified, then the data are assumed to be monthly. If a date variable is specified using the DATE= option, it is not necessary to specify the INTERVAL= or SEASONS= option; however, if specified, the values of the INTERVAL= or SEASONS= option should not be in conflict with the values of the date variable.

SEASONS= *period*

specifies the number of observations in a seasonal cycle. If the SEASONS= option is not specified and INTERVAL=QTR, then SEASONS=4 is assumed. If the SEASONS= option is not specified and INTERVAL=MONTH, then SEASONS=12 is assumed. If the SEASONS= option is specified, its value should not conflict with the values of the INTERVAL= option or the values of the date variable. See the preceding descriptions for the START=, DATE=, and INTERVAL= options for more details.

NOPRINT

suppresses any printed output.

ARIMA Statement

ARIMA *options;*

The ARIMA statement specifies the ARIMA part of the regARIMA model. This statement defines a pure ARIMA model if the regression statement is omitted. The ARIMA part of the model can include multiplicative seasonal factors.

The following option can appear in the ARIMA statement.

MODEL= ((*p d q* **)(** *P D Q* **) s)**

specifies the ARIMA model. The format follows standard Box-Jenkins notation (Box, Jenkins, and Reinsel 1994). The nonseasonal AR and MA orders are given by p and q, respectively, while the seasonal AR and MA orders are given by P and Q. The number of differences and seasonal differences are given by d and D, respectively. The notation (p d q) and (P D Q) can also be specified as (p, d, q) and (P, D, Q). The lag corresponding to seasonality is s. If s is omitted, it is set equal to the value used in the PROC X12 SEASONS= statement.

For example,

```
proc x12 data=ICMETI seasons=12 start=jan1968;
   arima model=((2,1,1)(1,1,0));
```

specifies an ARIMA (2,1,1)(1,1,0)12 model.

ESTIMATE Statement

ESTIMATE *;*

The ESTIMATE statement estimates the regARIMA model specified by the REGRESSION and ARIMA statements. Estimation output includes point estimates and standard errors for all estimated AR, MA, and regression parameters; the maximum likelihood estimate of the variance σ^2; t statistics for individual regression parameters; χ^2 statistics for assessing the joint significance of the parameters associated with certain regression effects (if included in the model); and likelihood-based model selection statistics (if the exact likelihood function is used). The regression effects for which χ^2 statistics are produced are fixed seasonal effects.

Tables displayed in association with estimation are Exact ARMA Likelihood Estimation Iteration Tolerances, Exact ARMA Likelihood Estimation Iteration Summary, Regression Model Parameter Estimates, Exact ARMA Maximum Likelihood Estimation, and Estimation Summary.

No options are currently available for the ESTIMATE statement.

IDENTIFY Statement

IDENTIFY *options;*

The IDENTIFY statement must be used to produce plots of the sample Autocorrelation Functions (ACFs) and Partial Autocorrelation Functions (PACFs) for identifying the ARIMA part of a regARIMA model. Sample ACFs and PACFs are produced for all combinations of the nonseasonal and seasonal differences of the data specified by the DIFF and SDIFF options. If the REGRESSION statement is present, the ACFs and PACFs are calculated for the specified differences of a series of regression residuals. If the REGRESSION statement is not present, the ACFs and PACFs are calculated for the specified differences of the original data.

Tables printed in association with identification are “Autocorrelation of Model Residuals” and “Partial Autocorrelation of Model Residuals”. If the REGRESSION statement is present, then the “Regression Model Parameter Estimates” table will also be printed.

The following options can appear in the IDENTIFY statement.

DIFF= *(order, order, order)*

specifies orders of nonseasonal differencing. The value 0 specifies no differencing, the value 1 specifies one nonseasonal difference $(1 - B)$, the value 2 specifies two nonseasonal differences $(1 - B)^2$, and so forth. The ACFs and PACFs are produced for all orders of nonseasonal differencing specified, in combination with all orders of seasonal differencing specified in the SDIFF= option. The default is DIFF=(0). You can specify up to three values for nonseasonal differences.

SDIFF= *(order, order, order)*

specifies orders of seasonal differencing. The value 0 specifies no seasonal differencing, the value 1 specifies one seasonal difference $(1 - B^s)$, the value 2 specifies two seasonal differences $(1 - B^s)^2$, and so forth. Here the value for *s* will correspond to the value of the SEASONS= option in the PROC X12 statement. The value of SEASONS= is supplied explicitly or is implicitly supplied through the INTERVAL= option or the values of the DATE= variable. The ACFs and PACFs are produced for all orders of seasonal differencing specified, in combination with all orders of non-seasonal differencing specified in the DIFF= option. The default is SDIFF=(0). You can specify up to three values for seasonal differences.

For example,

```
identify diff=(1) sdiff=(0, 1);
```

produces ACFs and PACFs for two models: $(1 - B)$ and $(1 - B)(1 - B^s)$.

OUTPUT Statement

OUTPUT OUT= *SAS-data-set tablename1 tablename2 ... ;*

The OUTPUT statement creates an output data set containing specified tables. The data set is named by the OUT= option.

OUT= *SAS-data-set*

If the OUT= option is omitted, the SAS System names the new data set using the default DATA*n* convention.

For each table to be included in the output data set, you must specify the X12 *tablename* keyword. The keyword corresponds to the title label used by the Census Bureau X12-ARIMA software. Currently available tables are A1, A6, A8, B1, C17, D8, D9, D10, D10D, D11, D12, D13, D16, D16B, D18, E5, E6, E7, and MV1. If no table is specified, table A1 will be displayed by default.

The tablename keywords that can be used in the OUTPUT statement are listed in the “Displayed Output/ODS Table Names” section on page 176. The following is an example of a VAR statement and an OUTPUT statement:

```
var sales costs;
output out=out_x12  b1 d11;
```

Note that the default variable name used in the output data set is the input variable name followed by an underscore and the corresponding table name. The variable sales_B1 contains the table B1 values for the variable sales, the variable costs_B1 contains the table B1 values for costs, while the table D11 values for sales are contained in the variable sales_D11, and the variable costs_D11 contains the table D11 values for costs. If necessary, the variable name is shortened so that the table name can be added. Currently, you cannot specify the output variable names. A variable named _DATE_ is the date identifier.

REGRESSION Statement

REGRESSION *options ;*

The REGRESSION statement includes regression variables in a regARIMA model or specifies regression variables whose effects are to be removed by the IDENTIFY statement to aid in ARIMA model identification. Predefined regression variables are selected with the PREDEFINED option. The only currently available predefined variables are length-of-month, length-of-quarter, and leap year. Table A6 provides information related to trading-day effects. You should note that missing values in the input series automatically create missing value regressors. Combining your model with additional predefined regression variables may result in a singularity problem. If a singularity occurs, then you may need to alter either the model or the choices of the predefined regressors in order to successfully perform the regression.

The following options can appear in the REGRESSION statement.

PREDEFINED= LOM
PREDEFINED= LOQ
PREDEFINED= LPYEAR

lists the predefined regression variables to be included in the model. Data values for these variables are calculated by the program, mostly as functions of the calendar. The values LOM and LOQ are actually equivalent: the actual regression is controlled by the PROC X12 SEASONS= option. Multiple predefined regression variables may be used. The syntax for using both a length-of-month and a leap-year regression could be one of the following forms:

```
regression predefined=lom lpyear;

regression predefined=(lom lpyear);

regression predefined=lom predefined=lpyear;
```

Table 5.1. Predefined Regression Variables in X-12-ARIMA

Regression Effect	Variable Definitions
Length-of-Month (monthly flow) LOM	$m_t - \bar{m}$ where m_t = length of month t (in days) and $\bar{m} = 30.4375$ (average length of month)
Length-of-Quarter (quarterly flow) LOQ	$q_t - \bar{q}$ where q_t = length of quarter t (in days) and $\bar{q} = 91.3125$ (average length of quarter)
Leap Year (monthly and quarterly flow) LPYEAR	$LY_t = \begin{cases} 0.75 & \text{in leap year February (first quarter)} \\ -0.25 & \text{in other Februaries (first quarter)} \\ 0 & \text{otherwise} \end{cases}$

TRANSFORM Statement

TRANSFORM *options;*

The TRANSFORM statement transforms or adjusts the series prior to estimating a regARIMA model. With this statement, the series can be Box-Cox (power) transformed.

The following option can appear in the TRANSFORM statement.

POWER= *value*

Transform the input series Y_t using a Box-Cox power transformation,

$$Y_t \to y_t = \begin{cases} log(Y_t) & \lambda = 0 \\ \lambda^2 + (Y_t^\lambda - 1)/\lambda & \lambda \neq 0 \end{cases}$$

The power λ must be specified (for example, POWER= .33). The default is no transformation ($\lambda = 1$); that is, POWER= 1. The log transformation (POWER= 0), square root transformation (POWER= .5), and the inverse transformation (POWER= -1) are equivalent to the corresponding Census Bureau function argument.

Table 5.2. Power Values Related to the Census Bureau Function Argument

function=	transformation	range for Y_t	equivalent power argument
none	Y_t	all values	$power = 1$
log	$log(Y_t)$	$Y_t > 0$ for all t	$power = 0$
sqrt	$2(\sqrt{Y_t} - 0.875)$	$Y_t \geq 0$ for all t	$power = 0.5$
inverse	$2 - \frac{1}{Y_t}$	$Y_t \neq 0$ for all t	$power = -1$

Note that there are restrictions on the value used in the POWER option when preadjustment factors for seasonal adjustment are generated from a regARIMA model. When seasonal adjustment is requested with the X11 statement, any value of the POWER option can be used for the purpose of forecasting the series with a regARIMA model. However, this is not the case when factors generated from the regression coefficients are used to adjust either the original series or the final seasonally adjusted series. In this case, the only accepted transformations are the log transformation, which can be specified as POWER = 0 (for multiplicative or log-additive seasonal adjustments) and no transformation, which can be specified as POWER = 1 (for additive seasonal adjustments). If no seasonal adjustment is performed, any POWER transformation can be used.

VAR Statement

VAR *variables;*

The VAR statement is used to specify the variables in the input data set that are to be analyzed by the procedure. Only numeric variables can be specified. If the VAR statement is omitted, all numeric variables contained in the input data set are analyzed.

X11 Statement

X11 *options;*

The X11 statement is an optional statement for invoking seasonal adjustment by an enhanced version of the methodology of the Census Bureau X-11 and X-11Q programs. You can control the type of seasonal adjustment decomposition calculated with the MODE= option. The output includes the final tables and diagnostics for the X-11 seasonal adjustment method listed below in Table 5.3.

Table 5.3. Tables Related to X11 Seasonal Adjustment

Table Name	Description
B1	original series, adjusted for prior effects and forecast extended
C17	final weights for the irregular component
D8	final unmodified SI ratios (differences)
D8A	*F* tests for stable and moving seasonality, D8
D9	final replacement values for extreme SI ratios (differences), D iteration
D9A	moving seasonality ratios for each period
D10	final seasonal factors
D10D	final seasonal difference
D11	final seasonally adjusted series
D12	final trend-cycle
D13	final irregular component
D16	combined seasonal and trading day factors
D16B	final adjustment differences
D18	combined calendar adjustment factors
E4	ratio of yearly totals of original and seasonally adjusted series
E5	percent changes (differences) in original series
E6	percent changes (differences) in seasonally adjusted series
E7	percent changes (differences) in final trend component series
F2A - F2I	X11 diagnostic summary
F3	monitoring and quality assessment statistics
G	spectral plots

For more details on the X-11 seasonal adjustment diagnostics, refer to Shiskin, Young, and Musgrave (1967), Lothian and Morry (1978), and Ladiray and Quenneville (1999).

The following options can appear in the X11 statement.

MODE= ADD
MODE= MULT
MODE= LOGADD
MODE= PSEUDOADD

determines the mode of the seasonal adjustment decomposition to be performed. There are four choices: multiplicative (MODE=MULT), additive (MODE=ADD), pseudo-additive (MODE=PSEUDOADD), and log-additive (MODE=LOGADD) decomposition. If this option is omitted, the procedure performs multiplicative adjustments. Table 5.4 shows the values of the MODE= option and the corresponding models for the original (O) and the seasonally adjusted (SA) series.

Table 5.4. Modes of Seasonal Adjustment and Their Models

Value of Mode Option	Name	Model for O	Model for SA
mult	Multiplicative	$O = C \times S \times I$	$SA = C \times I$
add	Additive	$O = C + S + I$	$SA = C + I$
pseudoadd	Pseudo-Additive	$O = C \times [S + I - 1]$	$SA = C \times I$
logadd	Log-Additive	$Log(O) = C + S + I$	$SA = exp(C + I)$

Details

Computations

For more details on the computations used in PROC X12, refer to *X-12-ARIMA Reference Manual* (U.S. Bureau of the Census 1999).

Displayed Output/ODS Table Names

The options specified in PROC X12 control both the tables produced by the procedure and the tables available for output to the OUT= data set specified in the OUTPUT statement.

The displayed output is organized into tables identified by a part letter and a sequence number within the part. The six major parts of the X12 procedure are as follows:

A prior adjustments (optional)

B preliminary estimates of irregular component weights and trading-day regression factors

C final estimates of irregular component weights and trading-day regression factors

D final estimates of seasonal, trend cycle, and irregular components

E analytical tables

F summary measures

G charts

Table 5.5 describes the individual tables and charts. “P” indicates that the table is displayed only and is not available for output to the OUT= data set. Data from displayed tables can be extracted into data sets using the Output Delivery System (ODS). Refer to Chapter 6, “Using the Output Delivery System,” in the *SAS/ETS User’s Guide*.

Table 5.5. Table Names and Descriptions

Table	Description	Notes
A1	original series	
RegParameterEstimates	regression model parameter estimates	P
ACF	autocorrelation factors	P
PACF	partial autocorrelation factors	P
ARMAIterationTolerances	exact ARMA likelihood estimation iteration tolerances	P
ARMAIterationSummary	exact ARMA likelihood estimation iteration summary	P
ARMAParameterEstimates	exact ARMA maximum likelihood estimation	P
MLESummary	estimation summary	P
ForecastCL	forecasts, standard errors, and confidence limits	P
MV1	original series adjusted for missing value regressors	
A6	regARIMA trading day component	
A8	regARIMA combined outlier component	
B1	prior adjusted or original series	
C17	final weight for irregular components	
D8	final unmodified S-I ratios	
D8A	seasonality tests	P
D9	final replacement values for extreme S-I ratios	
D9A	moving seasonality ratio	P
D10	final seasonal factors	
D10D	final seasonal difference	
D11	final seasonally adjusted series	
D12	final trend cycle	
D13	final irregular series	
D16	combined adjustment factors	
D16B	final adjustment differences	
D18	combined calendar adjustment factors	
E4	ratios of annual totals	P
E5	percent changes in original series	
E6	percent changes in final seasonally adjusted series	
E7	differences in final trend cycle	
F2A-I	summary measures	P
F3	quality assessment statistics	P
G	spectral analysis	P

The actual number of tables displayed depends on the options and statements specified. If a table is not computed, it is not displayed.

Examples

Example 5.1. Model Identification

An example of the statements typically invoked when using PROC X12 for model identification might follow the same format as the following example. This example invokes the X12 procedure and uses the TRANSFORM and IDENTIFY statements.

It specifies the time series data, takes the logarithm of the series (TRANSFORM statement), and generates ACFs and PACFs for the specified levels of differencing (IDENTIFY statement). The same data set can be used as in the section “Basic Seasonal Adjustment” on page 164.

```
proc x12 data=sales seasons=12 start=jul1972;
   var sales;
   transform power=0;
   identify diff=(0,1) sdiff=(0,1);
   run ;
```

```
                           The X12 Procedure

                   Autocorrelation of Model Residuals

       Differencing:  Nonseasonal Order=1 Seasonal Order=1
  Lag    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

    1      -0.34112     |             *******|  .                |
    2       0.10505     |                .   |** .               |
    3      -0.20214     |                ****|   .               |
    4       0.02136     |                .   |   .               |
    5       0.05565     |                .   |*  .               |
    6       0.03080     |                .   |*  .               |
    7      -0.05558     |                .  *|   .               |
    8      -0.00076     |                .   |   .               |
    9       0.17637     |                .   |****               |
   10      -0.07636     |                . **|   .               |
   11       0.06438     |                .   |*  .               |
   12      -0.38661     |            ********|   .               |
   13       0.15160     |               .    |*** .              |
   14      -0.05761     |               .   *|    .              |
   15       0.14957     |               .    |*** .              |
   16      -0.13894     |               . ***|    .              |
   17       0.07048     |               .    |*   .              |
   18       0.01563     |               .    |    .              |
   19      -0.01061     |               .    |    .              |
   20      -0.11673     |               .  **|    .              |
   21       0.03855     |               .    |*   .              |
   22      -0.09136     |               .  **|    .              |
   23       0.22327     |               .    |****.              |
   24      -0.01842     |               .    |    .              |
   25      -0.10029     |               .  **|    .              |
   26       0.04857     |               .    |*   .              |
   27      -0.03024     |               .   *|    .              |
   28       0.04713     |               .    |*   .              |
   29      -0.01803     |               .    |    .              |
   30      -0.05107     |               .   *|    .              |
   31      -0.05377     |               .   *|    .              |
   32       0.19573     |               .    |****.              |
   33      -0.12242     |               .  **|    .              |
   34       0.07775     |               .    |**  .              |
   35      -0.15245     |               . ***|    .              |
   36      -0.01000     |               .    |    .              |
```

Figure 5.3. ACFs (Nonseasonal Order=1 Seasonal Order=1)

Autocorrelation of Model Residuals
Differencing: Nonseasonal Order=1 Seasonal Order=1

Lag	Standard Error	Chi-Square	DF	Pr > ChiSq
1	0.08737	15.5957	1	<.0001
2	0.09701	17.0860	2	0.0002
3	0.09787	22.6478	3	<.0001
4	0.10101	22.7104	4	0.0001
5	0.10104	23.1387	5	0.0003
6	0.10128	23.2709	6	0.0007
7	0.10135	23.7050	7	0.0013
8	0.10158	23.7050	8	0.0026
9	0.10158	28.1473	9	0.0009
10	0.10389	28.9869	10	0.0013
11	0.10432	29.5887	11	0.0018
12	0.10462	51.4728	12	<.0001
13	0.11501	54.8664	13	<.0001
14	0.11653	55.3605	14	<.0001
15	0.11674	58.7204	15	<.0001
16	0.11820	61.6452	16	<.0001
17	0.11944	62.4045	17	<.0001
18	0.11975	62.4421	18	<.0001
19	0.11977	62.4596	19	<.0001
20	0.11978	64.5984	20	<.0001
21	0.12064	64.8338	21	<.0001
22	0.12074	66.1681	22	<.0001
23	0.12126	74.2099	23	<.0001
24	0.12436	74.2652	24	<.0001
25	0.12438	75.9183	25	<.0001
26	0.12500	76.3097	26	<.0001
27	0.12514	76.4629	27	<.0001
28	0.12520	76.8387	28	<.0001
29	0.12533	76.8943	29	<.0001
30	0.12535	77.3442	30	<.0001
31	0.12551	77.8478	31	<.0001
32	0.12569	84.5900	32	<.0001
33	0.12799	87.2543	33	<.0001
34	0.12888	88.3401	34	<.0001
35	0.12924	92.5584	35	<.0001
36	0.13061	92.5767	36	<.0001

NOTE: The P-values approximate the probability of observing a Q-value at least this large when the model fitted is correct. When DF is positive, small values of P, customarily those below 0.05 indicate model inadequacy.

Figure 5.4. ACFs (Nonseasonal Order=1 Seasonal Order=1)

```
                          The X12 Procedure

              Partial Autocorrelation of Model Residuals



         Differencing:  Nonseasonal Order=1 Seasonal Order=1
   Lag    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

     1       -0.34112    |             *******|  .                 |
     2       -0.01281    |                 .  |  .                 |
     3       -0.19266    |                ****|  .                 |
     4       -0.12503    |                 ***|  .                 |
     5        0.03309    |                 .  |* .                 |
     6        0.03468    |                 .  |* .                 |
     7       -0.06019    |                 . *|  .                 |
     8       -0.02022    |                 .  |  .                 |
     9        0.22558    |                 .  |*****               |
    10        0.04307    |                 .  |* .                 |
    11        0.04659    |                 .  |* .                 |
    12       -0.33869    |             *******|  .                 |
    13       -0.10918    |                 .**|  .                 |
    14       -0.07684    |                 .**|  .                 |
    15       -0.02175    |                 .  |  .                 |
    16       -0.13955    |                 ***|  .                 |
    17        0.02589    |                 .  |* .                 |
    18        0.11482    |                 .  |**.                 |
    19       -0.01316    |                 .  |  .                 |
    20       -0.16743    |                 ***|  .                 |
    21        0.13240    |                 .  |***                 |
    22       -0.07204    |                 . *|  .                 |
    23        0.14285    |                 .  |***                 |
    24       -0.06733    |                 . *|  .                 |
    25       -0.10267    |                 .**|  .                 |
    26       -0.01007    |                 .  |  .                 |
    27        0.04378    |                 .  |* .                 |
    28       -0.08995    |                 .**|  .                 |
    29        0.04690    |                 .  |* .                 |
    30       -0.00490    |                 .  |  .                 |
    31       -0.09638    |                 .**|  .                 |
    32       -0.01528    |                 .  |  .                 |
    33        0.01150    |                 .  |  .                 |
    34       -0.01916    |                 .  |  .                 |
    35        0.02303    |                 .  |  .                 |
    36       -0.16488    |                 ***|  .                 |
```

Figure 5.5. PACFs (Nonseasonal Order=1 Seasonal Order=1)

Example 5.2. Model Estimation

After studying the output from Example 5.1 and identifying the ARIMA part of the model as, for example, (0 1 1)(0 1 1) 12, you can replace the IDENTIFY statement with the ARIMA and ESTIMATE statements. The resulting procedure and statements follow:

```
proc x12 data=sales seasons=12 start=jul1972;
   var sales;
   transform power=0;
   arima model=( (0,1,1)(0,1,1) );
   estimate;
   run ;
```

```
                           The X12 Procedure

         Exact ARMA Likelihood Estimation Iteration Tolerances

              Maximum Total ARMA Iterations           200
              Convergence Tolerance               1.0E-05


      Average absolute percentage error in within-sample forecasts:

                      Last year:              2.81
                      Last-1 year:            6.38
                      Last-2 year:            7.69
                      Last three years:       5.63


           Exact ARMA Likelihood Estimation Iteration Summary

              Number of ARMA iterations                  6
              Number of Function Evaluations            19


                Exact ARMA Maximum Likelihood Estimation
                                            Standard
Parameter               Lag    Estimate       Error  t Value  Pr > |t|

Nonseasonal MA            1     0.40181     0.07887     5.09    <.0001
Seasonal MA              12     0.55695     0.07626     7.30    <.0001


                           Estimation Summary

              Number of Residuals                      131
              Number of Parameters Estimated             3
              Variance Estimate                    1.3E-03
              Standard Error Estimate              3.7E-02
              Log likelihood                      244.6965
              Transformation Adjustment          -735.2943
              Adjusted Log likelihood            -490.5978
              AIC                                 987.1956
              AICC (F-corrected-AIC)              987.3845
              Hannan Quinn                        990.7005
              BIC                                 995.8211
```

Figure 5.6. Estimation Data as Printed in the List File

Example 5.3. Seasonal Adjustment

Assuming that the model in Example 5.2 is satisfactory, a seasonal adjustment utilizing forecast extension can be performed by adding the X11 statement to the procedure. By default, the data is forecast one year ahead at the end of the series. The resulting PROC X12 step follows:

```
ods output D8A(match_all=list)=SalesD8A_1;
proc x12 data=sales date=date;
   var sales;
   transform power=0;
   arima model=( (0,1,1)(0,1,1) );
   estimate;
   x11;
   run;
```

```
proc print data=SalesD8A_1;
   title 'Stable Seasonality Test';
   run;

proc print data=SalesD8A_2;
   title 'Nonparametric Stable Seasonality Test';
   run;

proc print data=SalesD8A_3;
   title 'Moving Seasonality Test';
   run;
```

The ODS statement in the preceding example directs output from the D8A table into three data sets: SalesD8A_1, SalesD8A_2, and SalesD8A_3. (match_all=list) is necessary because the three tables associated with table D8A have varying formats.

```
                           The X12 Procedure

                Table D 8.A:  F-tests for seasonality

       Test for the presence of seasonality assuming stability.
                      Sum of                    Mean
                     Squares          DF      Square     F-Value

Between Months      23571.41          11    2142.855    190.9544    **
Residual             1481.28         132    11.22182
Total               25052.69         143

          ** Seasonality present at the 0.1 per cent level.


                  Nonparametric Test for the Presence
                   of Seasonality Assuming Stability
                  Kruskal-
                    Wallis                Probability
                 Statistic          DF          Level

                  131.9546          11           .00%

            Seasonality present at the one percent level.


                       Moving Seasonality Test
                      Sum of                    Mean
                     Squares          DF      Square     F-Value

Between Years       259.2517          10    25.92517    3.370317    **
Error               846.1424         110    7.692204

        **Moving seasonality present at the one percent level.
```

Figure 5.7. Table D8.A as Printed in the List File

Stable Seasonality Test

Obs	FT_SRC	FT_SS	FT_DF	FT_MS	FT_F	FT_AST
1	Between Months	23571.41	11	2142.855	190.9544	**
2	Residual	1481.28	132	11.22182	.	
3	Total	25052.69	143	.	.	

Nonparametric Stable Seasonality Test

Obs	KW_ST	KW_DF	KW_PR
1	131.9546	11	.00%

Moving Seasonality Test

Obs	FT_SRC	FT_SS	FT_DF	FT_MS	FT_F	FT_AST
1	Between Years	259.2517	10	25.92517	3.370317	**
2	Error	846.1424	110	7.692204	.	

Figure 5.8. Table D8.A as Output in a Data Set

Acknowledgments

The X-12-ARIMA procedure was developed by the Time Series Staff of the Statistical Research Division, U.S. Bureau of the Census.

Brian Monsell is the primary programmer for the U.S. Census Bureau's X-12-ARIMA procedure and has been very helpful in the development of PROC X12.

The version of PROC X12 documented here was produced by converting the U.S. Census Bureau's FORTRAN code to the SAS development language and adding typical SAS procedure syntax. This conversion work was performed by SAS Institute, which resulted in the X12 procedure. Although several features were added during the conversion, credit for the statistical aspects and general methodology of the X12 procedure belongs to the U.S. Bureau of the Census.

The X-12-ARIMA seasonal adjustment program contains components developed from Statistics Canada's X-11-ARIMA program.

References

Box, G. E. P., Jenkins, G. M., and Reinsel, G. C. (1994), *Time Series Analysis: Forecasting and Control*, Third Edition, Englewood Cliffs, NJ: Prentice Hall, Inc.

Dagum, E. B. (1988), *The X-11-ARIMA/88 Seasonal Adjustment Method: Foundations and User's Manual*, Ottawa: Statistics Canada.

Findley, D. F., Monsell, B. C., Bell, W. R., Otto, M. C., and Chen, B. C. (1998), "New Capabilities and Methods of the X-12-ARIMA Seasonal Adjustment Program," *Journal of Business and Economic Statistics*, 16, 127–176 (with Discussion).

Ladiray, D. and Quenneville B. (1999), "Understanding the X11 Method," Working Paper, Time Series Research and Analysis Centre, Statistics Canada, [http://www.census.gov/pub/ts/papers/x11tb_en.ps].

Lothian, J. and Morry M., (1978), "A Test of Quality Control Statistics for the X-11-ARIMA Seasonal Adjustment Program," Research Paper, Seasonal Adjustment and Times Series Staff, Statistics Canada.

Shiskin, J., Young, A. H., and Musgrave, J. C. (1967), "The X-11 Variant of the Census Method II Seasonal Adjustment Program," Technical Paper No. 15, U.S. Department of Commerce, Bureau of the Census.

U.S. Bureau of the Census (1999), *X-12-ARIMA Reference Manual*, U.S. Department of Commerce, Washington, DC, [ftp://ftp.census.gov/pub/ts/x12a/].

U.S. Bureau of the Census (1999), "X-12-ARIMA Seasonal Adjustment Program," [ftp://ftp.census.gov/pub/ts/x12a/].

U.S. Bureau of the Census (1999), *X-12-ARIMA Quick Reference for Unix*, U.S. Department of Commerce, Washington, DC, [ftp://ftp.census.gov/pub/ts/x12a/].

Part 3
Investment Analysis

Table of Contents

Chapter 6
Overview

Chapter Table of Contents

Chapter 6
Overview

About Investment Analysis

The Investment Analysis system is an interactive environment for the time-value of money of a variety of investments:

- Loans
- Savings
- Depreciations
- Bonds
- Generic cashflows

Various analyses are provided to help analyze the value of investment alternatives: time value, periodic equivalent, internal rate of return, benefit-cost ratio, and breakeven analysis.

These analyses can help answer a number of questions you may have about your investments:

- Which option is more profitable or less costly?
- Is it better to buy or rent?
- Are the extra fees for refinancing at a lower interest rate justified?
- What is the balance of this account after saving this amount periodically for so many years?
- How much is legally tax-deductible?
- Is this a reasonable price?

Investment Analysis can be beneficial to users in many industries for a variety of decisions:

- manufacturing: cost justification of automation or any capital investment, replacement analysis of major equipment, or economic comparison of alternative designs
- government: setting funds for services
- finance: investment analysis and portfolio management for fixed-income securities

Starting Investment Analysis

There are two ways to invoke Investment Analysis from the main SAS window. One way is to select **Solutions** → **Analysis** → **Investment Analysis** from the main SAS menu, as displayed in Figure 6.1.

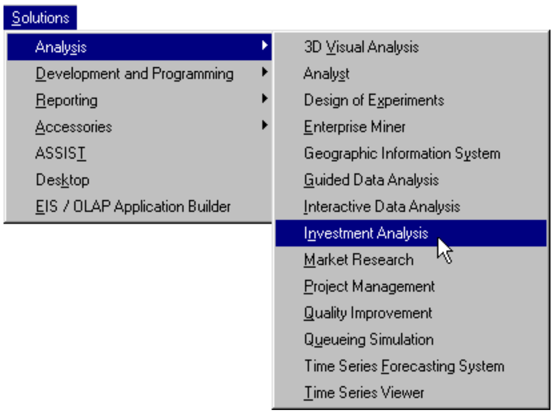

Figure 6.1. Initializing Investment Analysis with the Menu Bar

The other way is to type **INVEST** into the toolbar's command prompt, as displayed in Figure 6.2.

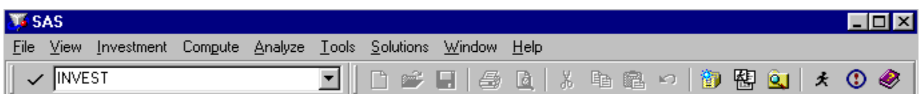

Figure 6.2. Initializing Investment Analysis with the Toolbar

Getting Help

You can get help in Investment Analysis in three ways. One way is to use the Help Menu, as displayed in Figure 6.3. This is the right-most menu item on the menu bar.

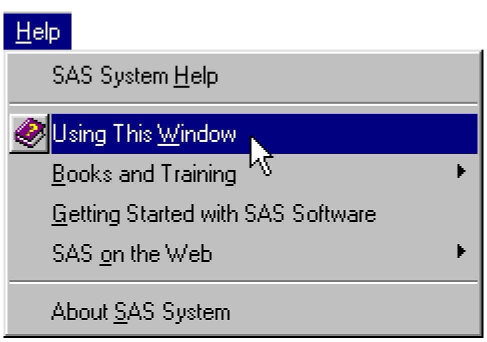

Figure 6.3. The Help Menu

Help buttons, as in Figure 6.4, provide another way to access help. Most dialog boxes provide help buttons in their lower-right corners.

Figure 6.4. A Help Button

Also, the toolbar has a button (see Figure 6.5) that invokes the help system. This is the right-most icon on the toolbar.

Figure 6.5. The Help Icon

Each of these methods invokes a browser that gives specific help for the active window.

Using Help

The chapters pertaining to Investment Analysis in this document typically have a section that introduces you to a menu and summarizes the options available through the menu. Such chapters then have sections titled Task and Dialog Box Guides. The Task section provides a description of how to perform many useful tasks. The Dialog Box Guide lists all dialog boxes pertinent to those tasks and gives a brief description of each element of each dialog box.

Chapter 7
Portfolios

Chapter Table of Contents

Chapter 7
Portfolios

The File Menu

Investment Analysis stores portfolios as catalog entries. Portfolios contain a collection of investments, providing a structure to collect investments with a common purpose or goal (like a retirement or building fund portfolio). It may be advantageous also to collect investments into a common portfolio if they are competing investments you wish to perform a comparative analysis upon. Within this structure you can perform computations and analyses on a collection of investments in a portfolio, just as you would perform them on a single investment.

Investment Analysis provides many tools to aid in your manipulation of portfolios through the **File** menu, shown in Figure 7.1.

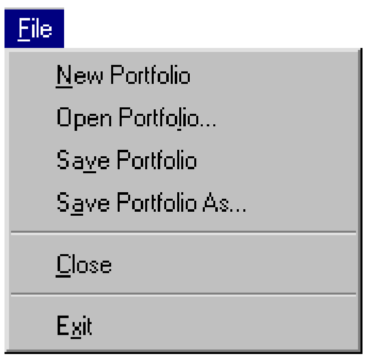

Figure 7.1. File Menu

The **File** menu offers the following items:

New Portfolio creates an empty portfolio with a new name.

Open Portfolio... opens the standard SAS Open dialog box where you select a portfolio to open.

Save Portfolio saves the current portfolio to its current name.

Save Portfolio As... opens the standard SAS Save As dialog box where you supply a new portfolio name for the current portfolio.

Close closes Investment Analysis.

Exit closes SAS (Windows only).

Tasks

Creating a New Portfolio

From the Investment Analysis dialog box, select **File** → **New Portfolio**.

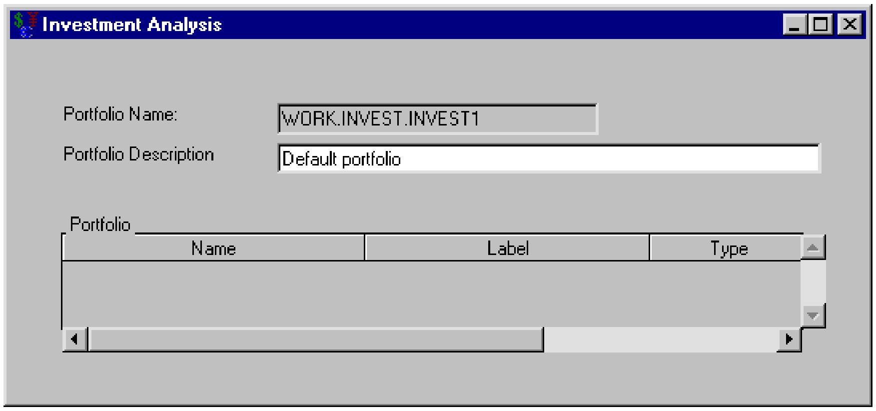

Figure 7.2. Creating a New Portfolio

The **Portfolio Name** will be WORK.INVEST.INVEST1 as displayed in Figure 7.2, unless you have saved a portfolio to that name in the past. In that case some other unused portfolio name is given to the new portfolio.

Saving a Portfolio

From the Investment Analysis dialog box, select **File** → **Save Portfolio**. The portfolio is saved to a catalog-entry with the name in the **Portfolio Name** box.

Opening an Existing Portfolio

From the Investment Analysis dialog box, select **File** → **Open Portfolio...**. This opens the standard SAS Open dialog box. You enter the name of a SAS portfolio to open in the **Entry Name** box. For example, enter SASHELP.INVSAMP.NVST as displayed in Figure 7.3.

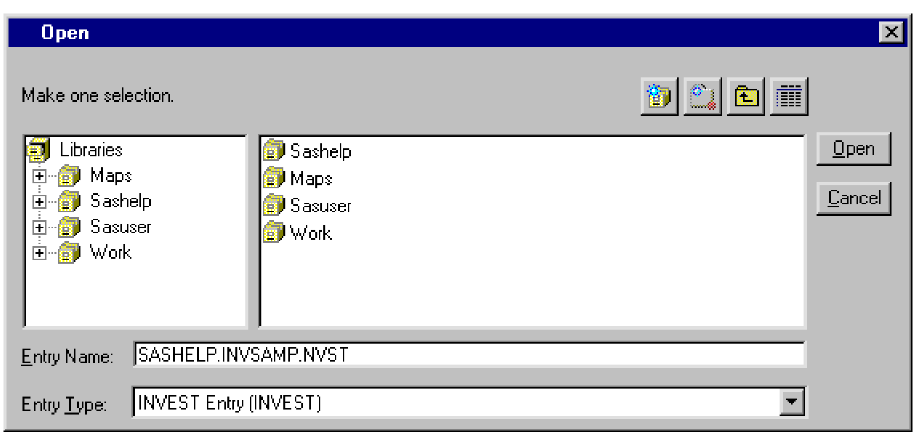

Figure 7.3. Opening an Existing Portfolio

Click **Open** to load the portfolio. The portfolio should look like Figure 7.4.

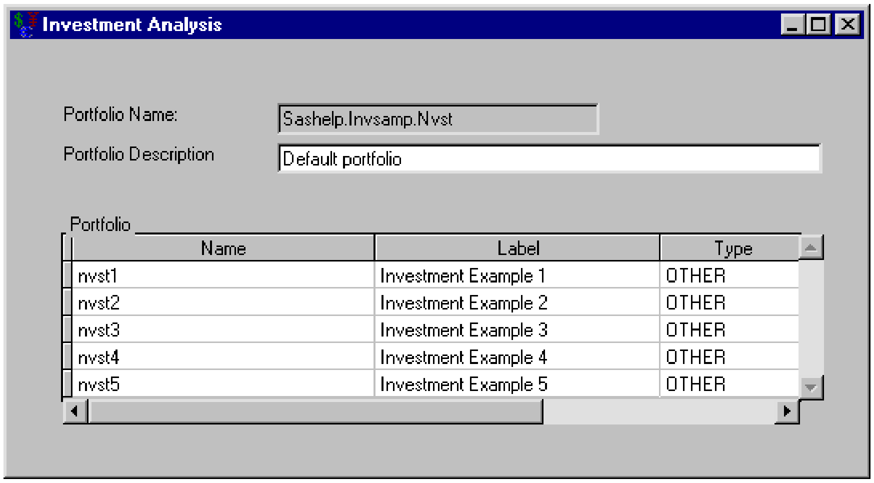

Figure 7.4. The Opened Portfolio

Saving a Portfolio to a Different Name

From the Investment Analysis dialog box, select **File** → **Save Portfolio As...**.

This opens the standard SAS Save As dialog box. You can enter the name of a SAS portfolio into the **Entry Name** box. For example, enter SASUSER.MY_PORTS.PORT1 as in Figure 7.5.

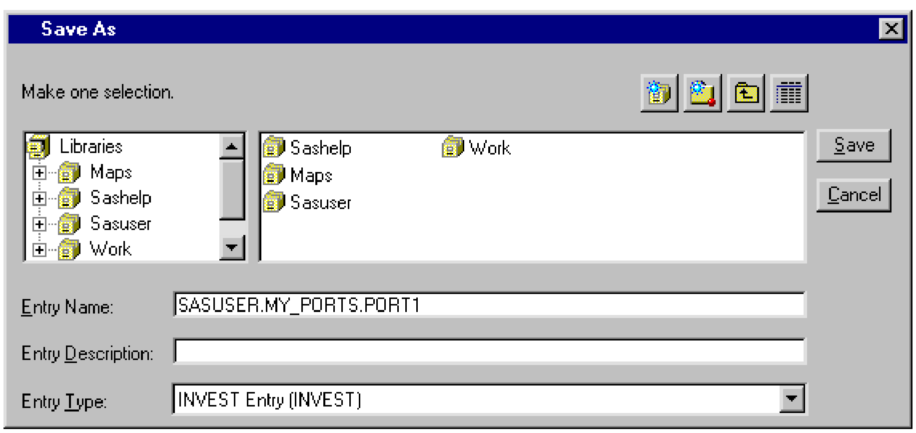

Figure 7.5. Saving a Portfolio to a Different Name

Click **Save** to save the portfolio.

Selecting Investments within a Portfolio

To select a single investment in an opened portfolio, click the investment in the Portfolio area within the Investment Analysis dialog box.

To select a list of adjacent investments, do the following: click the first investment, hold down SHIFT, and click the final investment. Once the list of investment is selected, you may release the SHIFT key. The selected investments will appear highlighted as in Figure 7.6.

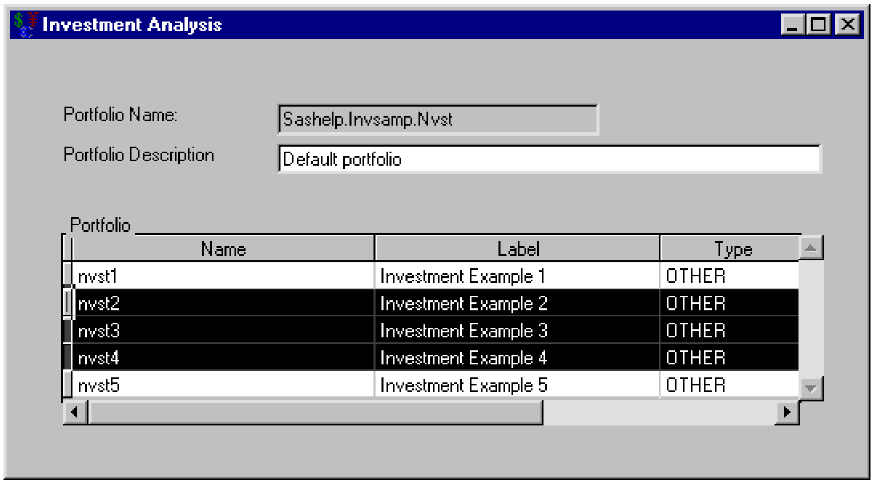

Figure 7.6. Selecting Investments within a Portfolio

Dialog and Utility Guide

Investment Analysis

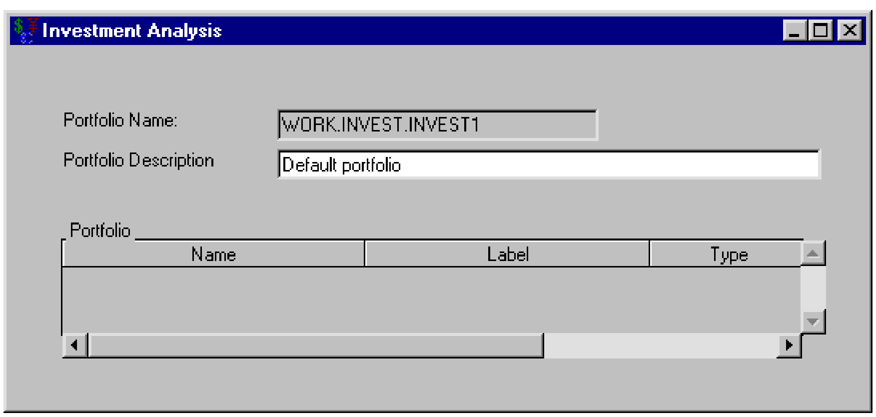

Figure 7.7. Investment Analysis Dialog Box

Investment Portfolio Name holds the name of the portfolio. It is of the form library.catalog_entry.portfolio. The default portfolio name is work.invest.invest1, as in Figure 7.7.

Portfolio Description provides a more descriptive explanation of the portfolio's contents. You can edit this description any time this dialog box is active.

The **Portfolio** area contains the list of investments comprising the particular portfolio. Each investment in the **Portfolio** area displays the following attributes:

Name is the name of the investment. It must be a valid SAS name. It is used to distinguish investments when performing analyses and computations.

Label is a place where you can provide a more descriptive explanation of the investment.

Type is the type of investment, which is fixed when you create the investment. It will be one of the following: LOAN, SAVINGS, DEPRECIATION, BOND, or OTHER.

Additional tools to aid in the management of your portfolio are available by selecting from the menu bar and by right-clicking within the **Portfolio** area.

Menubar Options

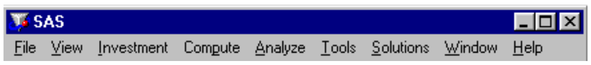

Figure 7.8. The Menu bar

The menu bar (shown in Figure 7.8) provides many tools to aid in the management of portfolios and the investments that comprise them. The following menu items provide functionality particular to Investment Analysis:

File opens and saves portfolios.

Investment creates new investments within the portfolio.

Compute performs constant dollar, after tax, and currency conversion computations on generic cashflows.

Analyze analyzes investments to aid in decision-making.

Tools sets default values of inflation and income tax rates.

Right-Clicking within the Portfolio Area

Figure 7.9. Right-Clicking

After selecting an investment, right-clicking in the **Portfolio** area pops up a menu (see Figure 7.9) that offers the following options:

Edit opens the selected investment within the portfolio.

Duplicate creates a duplicate of the selected investment within the portfolio.

Delete removes the selected investment from the portfolio.

If you wish to perform one of these actions on a collection of investments, you must select a collection of investments (as described in the section "Selecting Investments within a Portfolio" on page 197) before right-clicking.

Chapter 8
Investments

Chapter Table of Contents

Chapter 8
Investments

The Investment Menu

Because there are many types of investments, a tool that manages and analyzes collections of investments must be robust and flexible. Providing specifications for four specific investment types and one generic type, Investment Analysis can model almost any real-world investment.

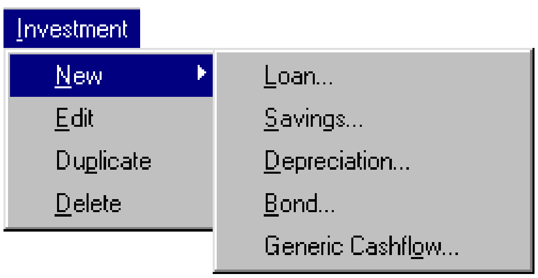

Figure 8.1. Investment Menu

The **Investment** menu, shown in Figure 8.1, offers the following items.

New → **Loan...** opens the Loan dialog box. Loans are useful for acquiring capital to pursue various interests. Available terms include rate adjustments for variable rate loans, initialization costs, prepayments, and balloon payments.

New → **Savings...** opens the Savings dialog box. Savings are necessary when planning for the future, whether for business or personal purposes. Account summary calculations available per deposit include starting balance, deposits, interest earned, and ending balance.

New → **Depreciation...** opens the Depreciation dialog box. Depreciations are prevalent for tax purposes. It is advantageous to deduct as much as possible as early as possible. The available depreciation methods are Straight Line, Sum-of-years Digits, Depreciation Table, and Declining Balance. SAS datasets containing the Modified Accelerated Cost Recovery System rates for various half-year conventions are available to load into the **Depreciation** area. Declining Balance with conversion to Straight Line is also provided.

New → **Bond...** opens the Bond dialog box. Bonds have widely varying terms depending on the issuer. As bond issuers frequently auction their bonds, the ability to price a bond between the issue date and maturity date is desirable. Fixed-coupon bonds may be analyzed for the following: price versus yield-to-maturity, duration, and convexity. These are available at different times in the bond's life.

New → **Generic Cashflow...** opens the Generic Cashflow dialog box. Generic cashflows are the most flexible investments. Only a sequence of date-amount pairs is necessary for specification. You can enter date-amount pairs and load values from SAS datasets to specify any type of investment. You can generate uniform, arithmetic, and geometric cashflows with ease. SAS's forecasting ability is available to forecast future cashflows as well. The new graphical display aids in visualization of the cashflow and enables the user to change the frequency of the cashflow view to aggregate and disaggregate the view.

Edit opens the specification dialog box for an investment selected within the portfolio.

Duplicate creates a duplicate of an investment selected within the portfolio.

Delete removes an investment selected from the portfolio.

If you wish to edit, duplicate, or delete a collection of investments, you must select a collection of investments as described in "Selecting Investments within a Portfolio" on page 197 before performing the menu-option.

Tasks

Loan Tasks

Suppose you want to buy a home that costs $100,000. You can make a down payment of $20,000. Hence, you need a loan of $80,000. You are able to acquire a 30-year loan at 7% interest starting January 1, 2000. Let's use Investment Analysis to specify and analyze this loan.

From the Investment Analysis dialog box, select **Investment** → **New** → **Loan...** from the menu bar to open the Loan dialog box.

Specifying Loan Terms to Create an Amortization Schedule

You must specify the loan before generating the amortization table. To specify the loan, follow these steps:

1. Enter **MORTGAGE** for the **Name**.
2. Enter 80000 for the **Loan Amount**.
3. Enter 7 for the **Initial Rate**.
4. Enter 360 for the **Number of Payments**.
5. Enter 01JAN2000 for the **Start Date**.

Once you have specified the loan, click **Create Amortization Schedule** to generate the amortization schedule displayed in Figure 8.2.

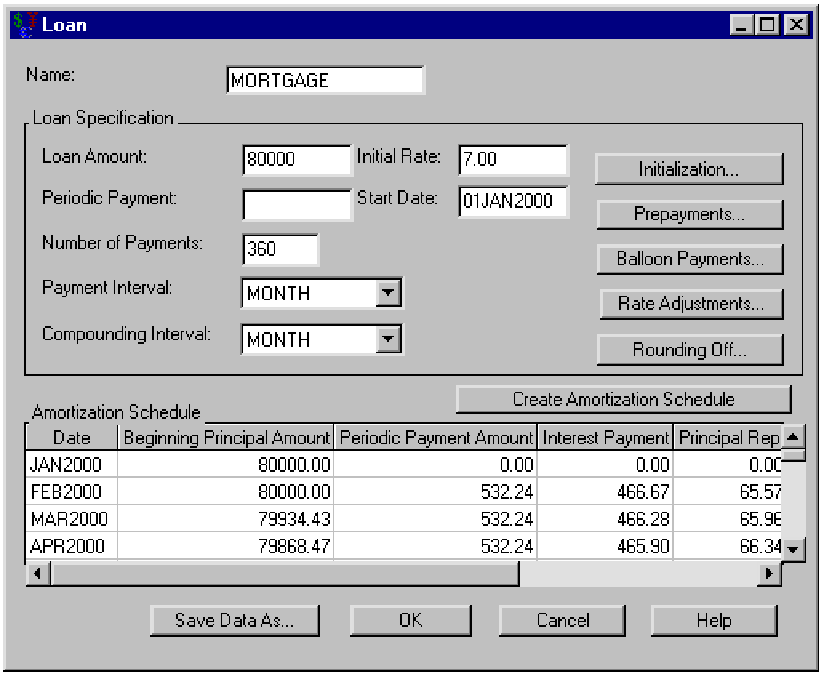

Figure 8.2. Creating an Amortization Schedule

Storing Other Loan Terms

Let's include information concerning the purchase price and downpayment. These terms are not necessary to specify the loan, but it may be advantageous to store such information with the loan.

Consider the loan described in "Loan Tasks" on page 204. From the Loan dialog box (Figure 8.2) click **Initialization...** to open the Loan Initialization Options dialog box. Here you can specify the down payment, initialization costs, and discount points. To specify the down payment, enter 100000 for the **Purchase Price**, as shown in Figure 8.3.

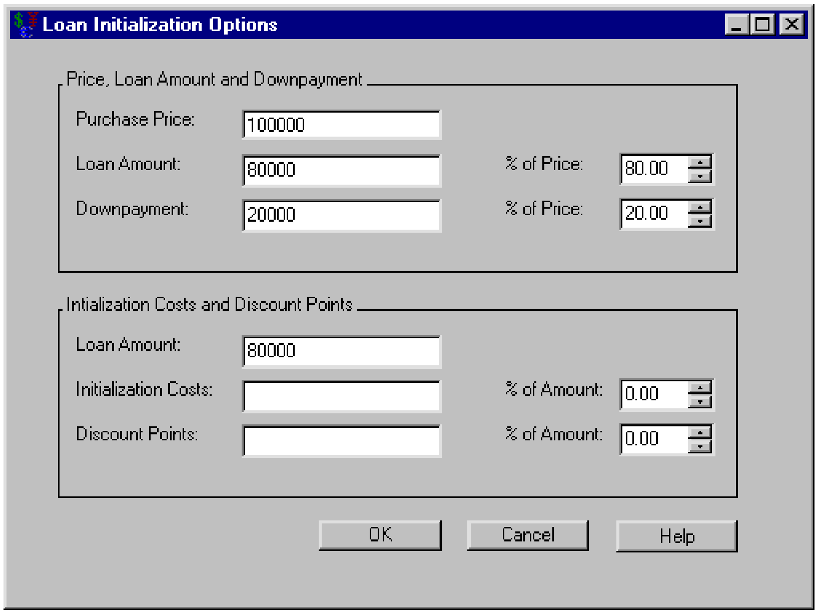

Figure 8.3. Including the Purchase Price

Click **OK** to return to the Loan dialog box.

Adding Prepayments

Now let's observe the effect of prepayments on the loan. Consider the loan described in "Loan Tasks" on page 204. You must pay a minimum of $532.24 each month to keep up with payments. However, let's say you dislike entering this amount in your checkbook. You would rather pay $550.00 to keep the arithmetic simpler. This would constitute a uniform prepayment of $17.76 each month.

From the Loan dialog box, click **Prepayments...** that opens the Loan Prepayments dialog box shown in Figure 8.4.

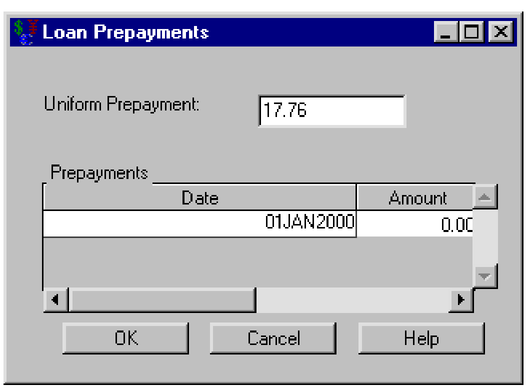

Figure 8.4. Specifying the Loan Prepayments

You can specify an arbitrary sequence of prepayments in the **Prepayments** area. If you want a uniform prepayment, clear the **Prepayments** area and enter the uniform payment amount in the **Uniform Prepayment** box. That amount will be added to each payment until the loan is paid off.

To specify this uniform prepayment, follow these steps:

1. Enter 17.76 for the **Uniform Prepayment**.
2. Click **OK** to return to the Loan dialog box.
3. Click **Create Amortization Schedule**, and the amortization schedule updates, as displayed in Figure 8.5.

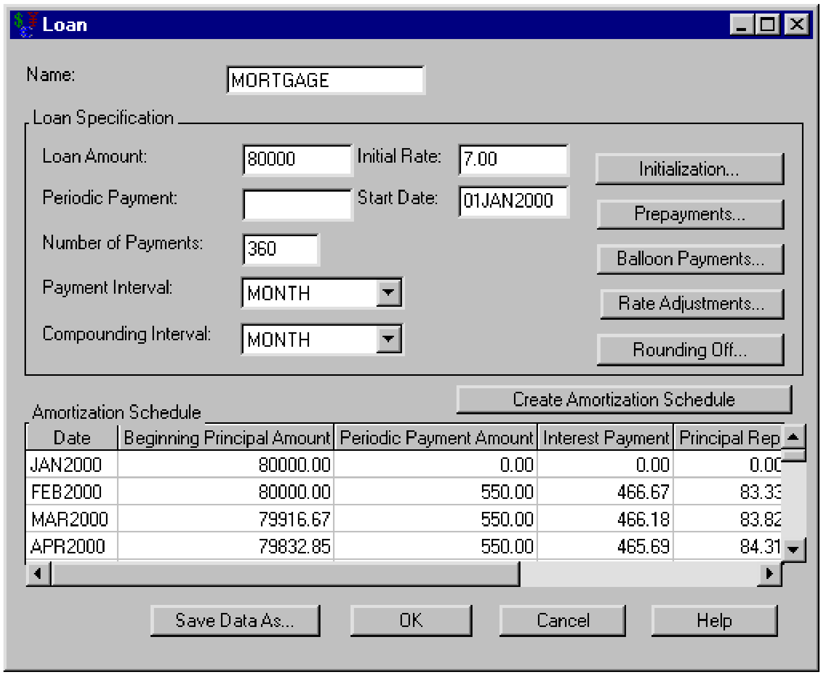

Figure 8.5. The Amortization Schedule with Loan Prepayments

The last payment is on January 2030 without prepayments and February 2027 with prepayment; you would pay the loan off almost three years earlier with the $17.76 prepayments.

To continue this example you must remove the prepayments from the loan specification, following these steps:

1. Return to the Prepayments dialog box from the Loan dialog box by clicking **Prepayments...**.
2. Enter 0 for **Uniform Prepayment**.
3. Click **OK** to return to the Loan dialog box.

Adding Balloon Payments

Consider the loan described in “Loan Tasks” on page 204. Suppose you cannot afford the payments of $532.24 each month. To lessen your monthly payment you could pay balloon payments of $6,000 at the end of 2007 and 2023. You wonder how this would affect your monthly payment.

From the Loan dialog box, click **Balloon Payments...**, which opens the Balloon Payments dialog box shown in Figure 8.6.

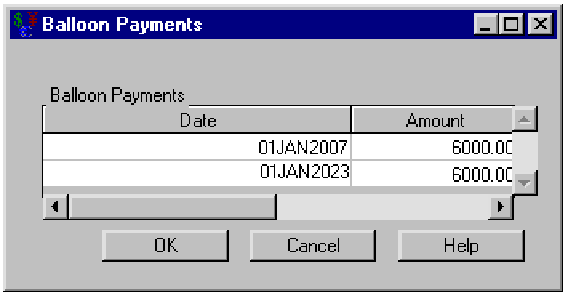

Figure 8.6. Defining Loan Balloon Payments

You can specify an arbitrary sequence of balloon payments by adding date-amount pairs to the **Balloon Payments** area.

To specify these balloon payments, follow these steps:

1. Right-click within the **Balloon Payment** area (which pops up a menu) and release on **New**.
2. Set the pair's **Date** to 01JAN2007.
3. Set its **Amount** to 6000.
4. Right-click within the **Balloon Payment** area (which pops up a menu) and release on **New**.
5. Set the new pair's **Date** to 01JAN2023.
6. Set its **Amount** to 6000.

Click **OK** to return to the Loan dialog box. Click **Create Amortization Schedule**, and the amortization schedule updates. Your monthly payment is now $500.30, a difference of approximately $32 each month.

To continue this example you must remove the balloon payments from the loan specification, following these steps:

1. Return to the Balloon Payments dialog box.
2. Right-click within the **Balloon Payment** area (which pops up a menu) and release on **Clear**.
3. Click **OK** to return to the Loan dialog box.

Handling Rate Adjustments

Consider the loan described in "Loan Tasks" on page 204. Another option for lowering your payments is to get a variable rate loan. You can acquire a three-year-ARM at 6% with a periodic cap of 1% with a maximum of 9%.

From the Loan dialog box, click **Rate Adjustments...** to open the Rate Adjustment Terms dialog box shown in Figure 8.7.

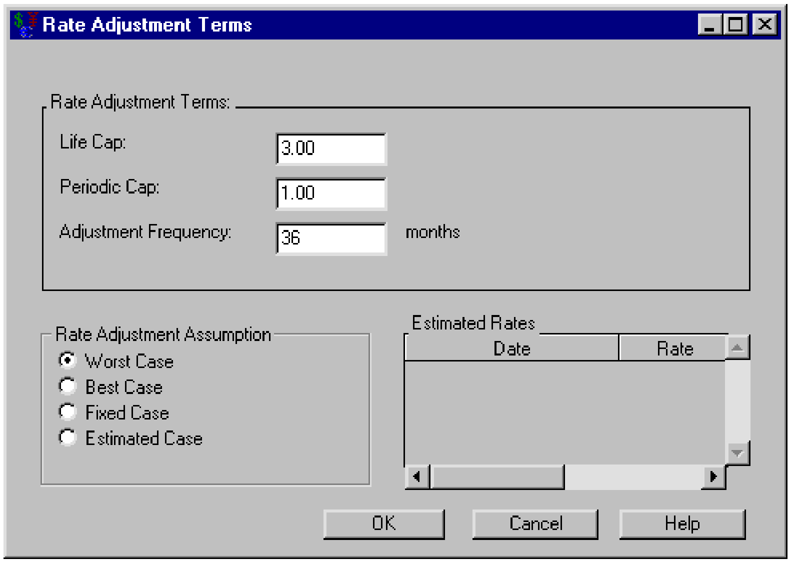

Figure 8.7. Setting the Rate Adjustments

To specify these loan adjustment terms, follow these steps:

1. Enter 3 for the **Life Cap**. The **Life Cap** is the maximum deviation from the **Initial Rate**.
2. Enter 1 for the **Periodic Cap**.
3. Enter 36 for the **Adjustment Frequency**.
4. Confirm that **Worst Case** is selected in the Rate Adjustment Assumption area.
5. Click **OK** to return to the Loan dialog box.
6. Enter 6 for the **Initial Rate**.

Click **Create Amortization Schedule**, and the amortization schedule updates. Your monthly payment drops to $479.64 each month. However, if the worst-case scenario plays out, the payments will increase to $636.84 in nine years. Figure 8.8 displays amortization table information for the final few months under this scenario.

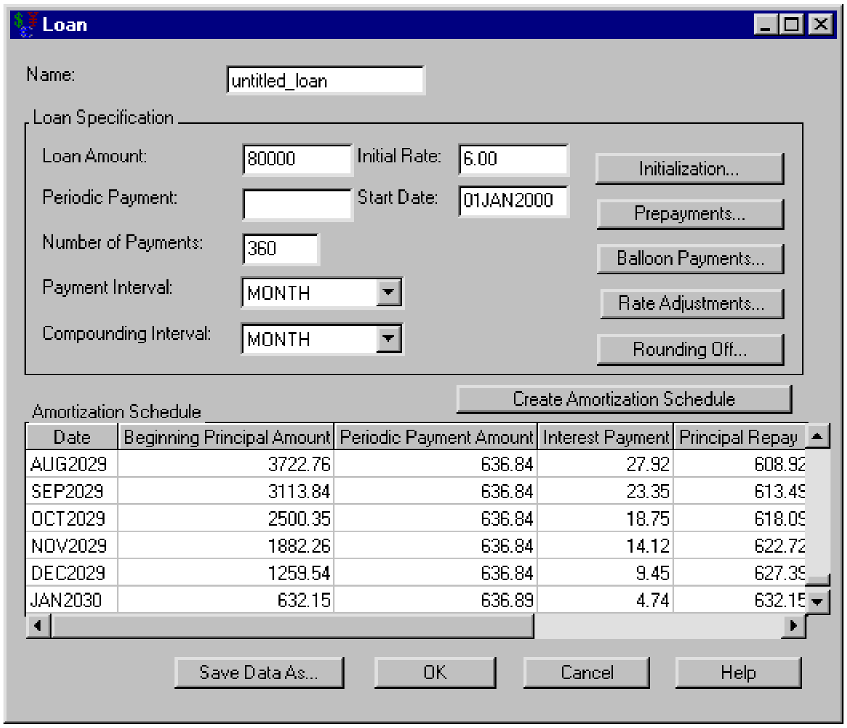

Figure 8.8. The Amortization Schedule with Rate Adjustments

Click **OK** to return to the Investment Analysis dialog box.

Specifying Savings Terms to Create an Account Summary

Suppose you put $500 each month into an account that earns 6% interest for 20 years. What is the balance of the account after those 20 years?

From the Investment Analysis dialog box, select **Investment** → **New** → **Savings...** from the menu bar to open the Savings dialog box.

To specify the savings, follow these steps:

1. Enter **RETIREMENT** for the **Name**.
2. Enter 500 for the **Periodic Deposit**.
3. Enter 240 for the **Number of Deposits**.
4. Enter 6 for the **Initial Rate**.

You must specify the savings before generating the account summary. Once you have specified the savings, click **Create Account Summary** to compute the ending date and balance and to generate the account summary displayed in Figure 8.9.

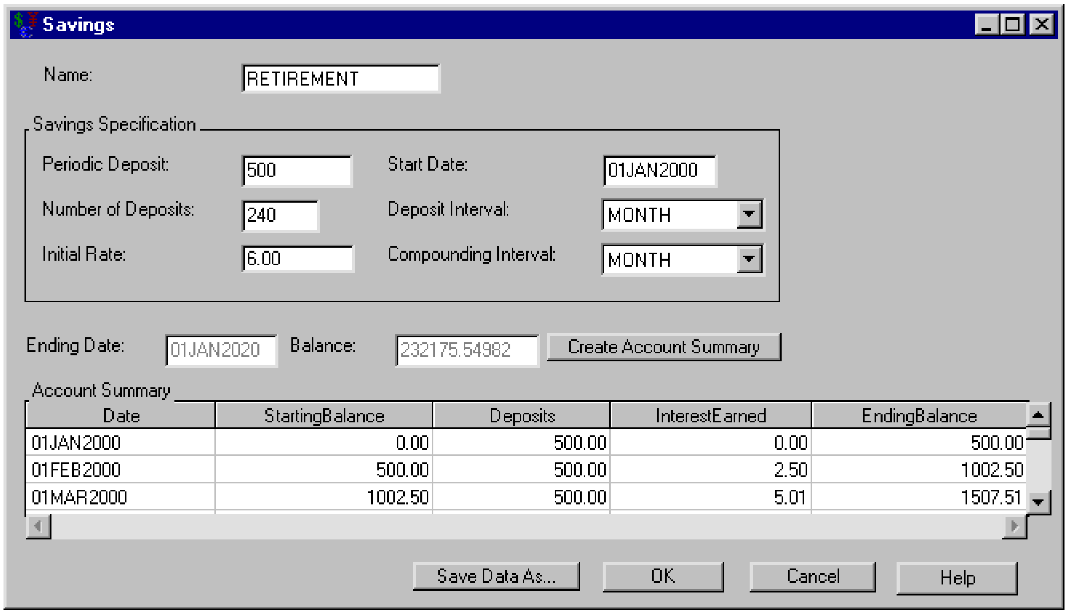

Figure 8.9. Creating an Account Summary

Click **OK** to return to the Investment Analysis dialog box.

Depreciation Tasks

Commercial assets are considered to lose value as time passes. For tax purposes, you want to quantify this loss. This investment structure helps calculate appropriate values.

Suppose you buy a boat that costs $50,000 for commercial fishing that is considered to have a ten-year useful life. How would you depreciate it?

From the Investment Analysis dialog box, select **Investment** → **New** → **Depreciation...** from the menu bar to open the Depreciation dialog box.

Specifying Depreciation Terms to Create a Depreciation Table

To specify the depreciation, follow these steps:

1. Enter **FISHING_BOAT** for the **Name**.
2. Enter 50000 for the **Cost**.
3. Enter 2000 for the **Year of Purchase**.
4. Enter 10 for the **Useful Life**.
5. Enter 0 for the **Salvage Value**.

You must specify the depreciation before generating the depreciation schedule. Once you have specified the depreciation, click **Create Depreciation Schedule** to generate a deprecation schedule like the one displayed in Figure 8.10.

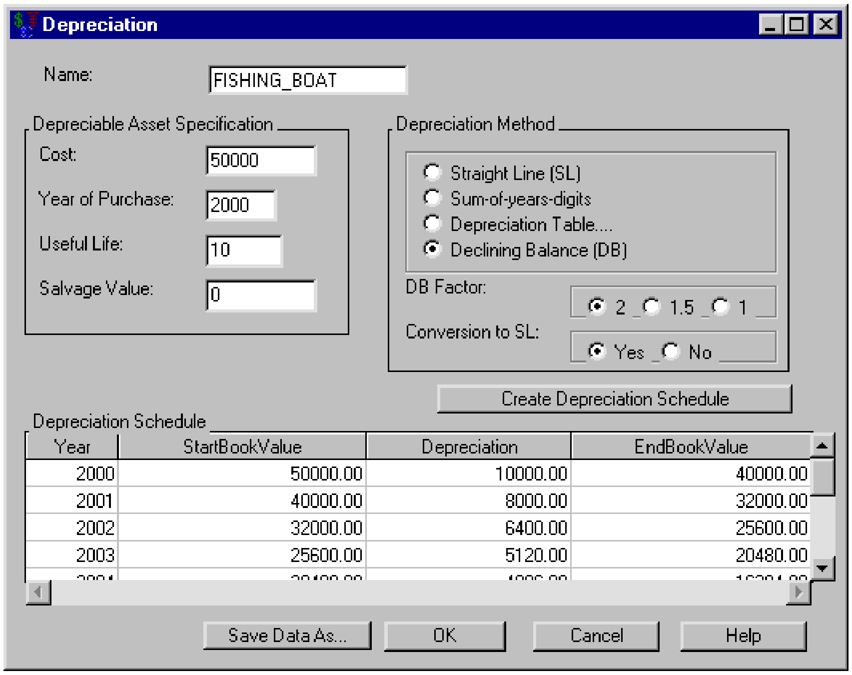

Figure 8.10. Creating a Depreciation Schedule

The default deprecation method is Declining Balance (with Conversion to Straight Line). Try the following methods to see how they each affect the schedule:

- Straight Line
- Sum-of-years Digits
- Declining Balance (without conversion to Straight Line)

It might be useful to compare the value of the boat at 5 years for each method.

A description of these methods is available in "Depreciation Methods" on page 274.

Using the Depreciation Table

Sometimes you want to force the depreciation rates to be certain percentages each year. This option is particularly useful for calculating Modified Accelerated Cost Recovery System (MACRS) Depreciations. The United States' Tax Reform Act of 1986 set depreciation rates for an asset based on an assumed lifetime for that asset. Since these lists of rates are important to many people, Investment Analysis provides SAS Datasets for situations with yearly rates (using the "half-year convention"). Find them at SASHELP.MACRS* where * refers to the class of the property. For example, use SASHELP.MACRS15 for a fifteen-year property. (When using the MACRS with the Tax Reform Act tables, you must set the **Salvage Value** to zero.)

Suppose you want to compute the depreciation schedule for the commercial fishing boat described in "Depreciation Tasks" on page 211. The boat is a ten-year property according to the Tax Reform Act of 1986.

To employ the MACRS depreciation from the Depreciation dialog box, follow these steps:

1. Click **Depreciation Table...** within the **Deprecation Method** area. This opens the Depreciation Table dialog box.
2. Right-click within the **Depreciation** area (which pops up a menu) and select **Load**.
3. Enter SASHELP.MACRS10 for the **Dataset Name**. The dialog box should look like Figure 8.11.

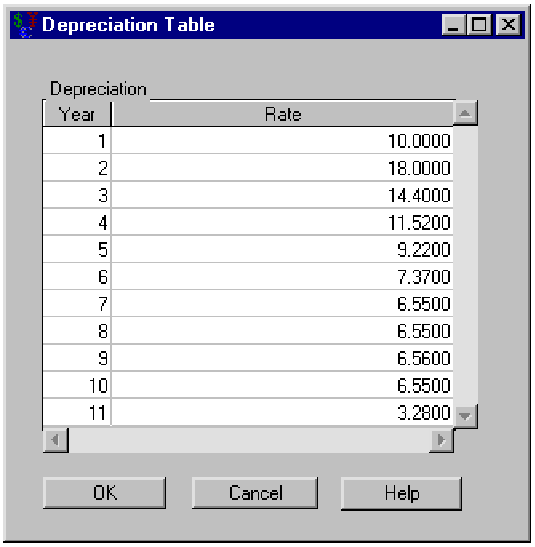

Figure 8.11. MACRS Percentages for a Ten-Year Property

Click **OK** to return to the Depreciation dialog box. Click **Create Depreciation Schedule** and the depreciation schedule fills (see Figure 8.12).

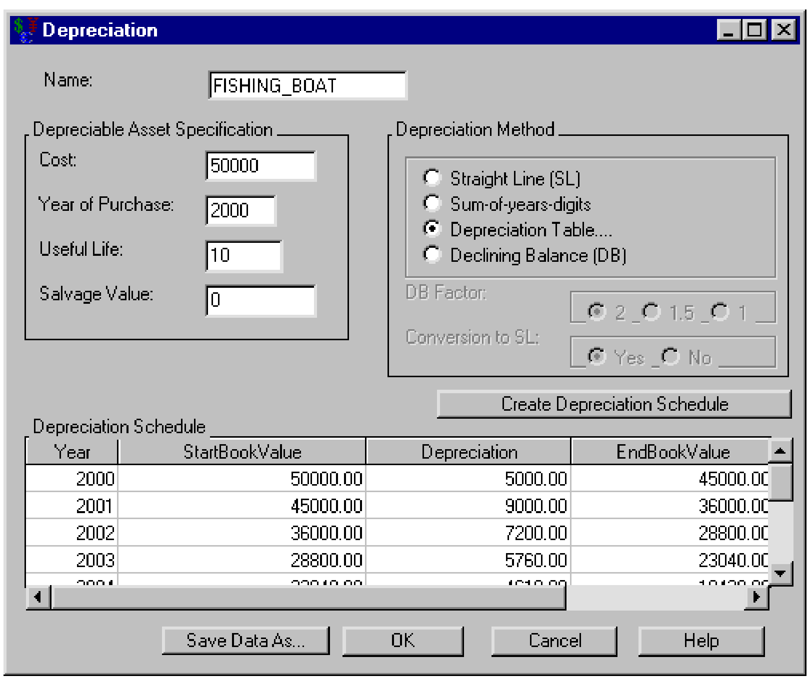

Figure 8.12. Depreciation Table with MACRS10

Note there are eleven entries in this depreciation table. This is because of the half-year convention that enables you to deduct one half of a year the first year which leaves a half year to deduct after the useful life is over.

Click **OK** to return to the Investment Analysis dialog box.

Bond Tasks

Suppose someone offers to sell you a 20-year utility bond. It was issued six years ago. It has a $1,000 face value and pays semi-year coupons at 2%. You can purchase it for $780. Would you be satisfied with this bond if you expect an 8% MARR?

From the Investment Analysis dialog box, select **Investment** → **New** → **Bond...** from the menu bar to open the Bond dialog box.

Specifying Bond Terms

To specify the bond, follow these steps:

1. Enter **UTILITY_BOND** for the **Name**.
2. Enter 1000 for the **Face Value**.
3. Enter 2 for the **Coupon Rate**. The **Coupon Payment** updates to 20.
4. Select SEMIYEAR for **Coupon Interval**.
5. Enter 28 for the **Number of Coupons**. As 14 years remain before the bond matures, the bond still has 28 semiyear coupons to pay. The **Maturity Date** updates.

Computing the Price from Yield

Enter 8 for **Yield** within the **Valuation** area. You see the bond's value would be $666.72 as in Figure 8.13.

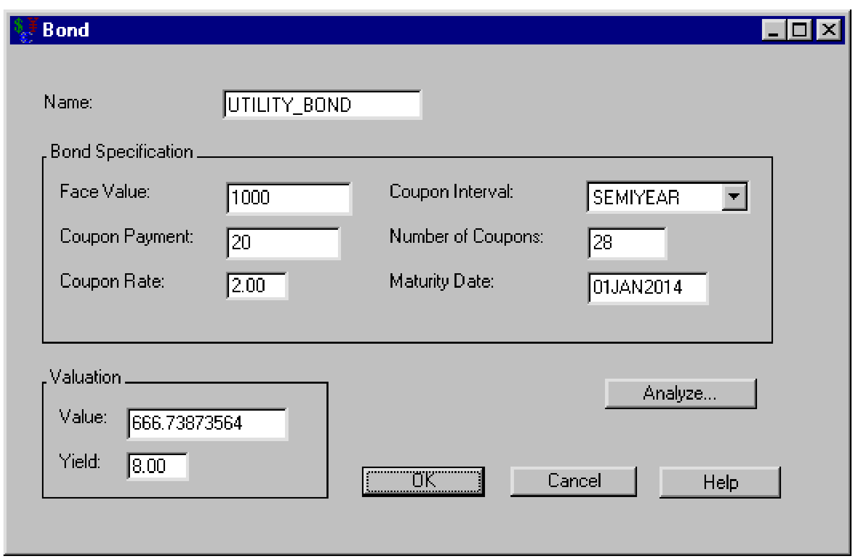

Figure 8.13. Bond Value

Computing the Yield from Price

Now enter 780 for **Value** within the **Valuation** area. You see the yield is only 6.5%, as in Figure 8.14. This is not acceptable if you desire an 8% MARR.

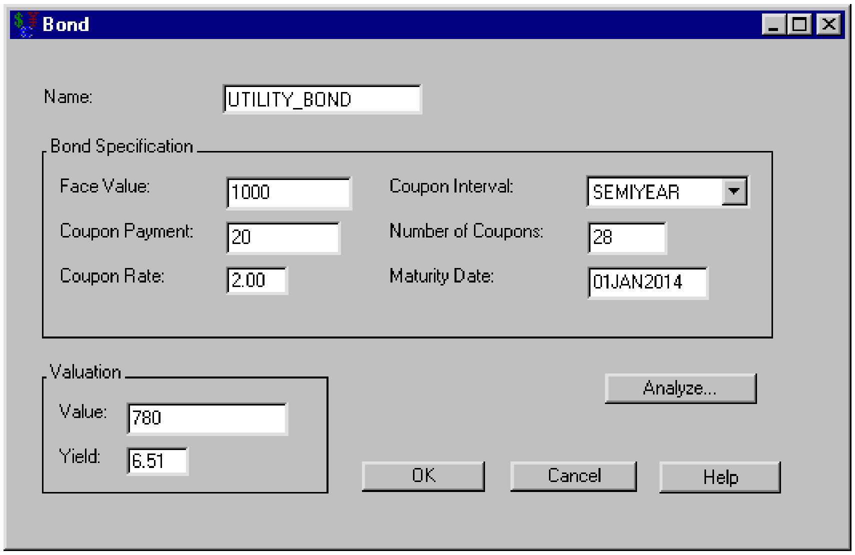

Figure 8.14. Bond Yield

Performing Bond Analysis

To perform bond-pricing analysis, follow these steps:

1. Click **Analyze...** to open the Bond Analysis dialog box.
2. Enter 8.0 as the **Yield to Maturity**.
3. Enter 4.0 as the +/-.
4. Enter 0.5 as the **Increment by**.
5. Enter 780 as the **Reference Price**.
6. Click **Create Bond Valuation Summary**.

The **Bond Valuation Summary** area fills and shows you the different values for various yields as in Figure 8.15.

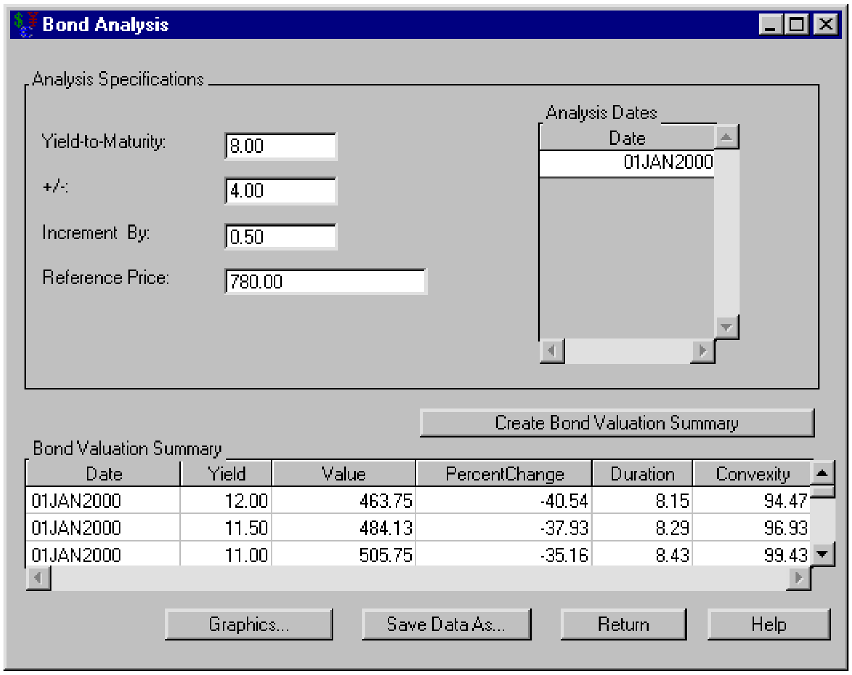

Figure 8.15. Bond Price Analysis

Creating a Price versus Yield-to-Maturity Graph

Click **Graphics...** to open the Bond Price dialog box. This contains the price versus yield-to-maturity graph shown in Figure 8.16.

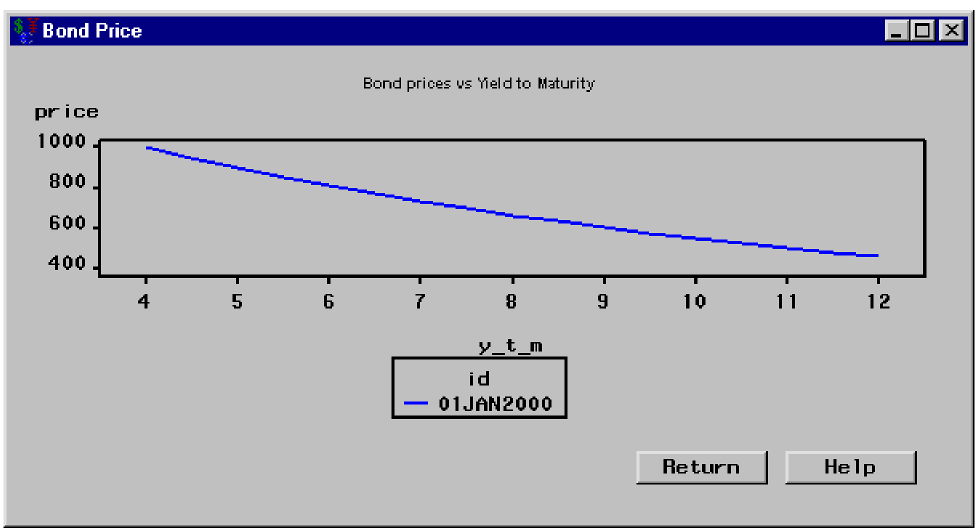

Figure 8.16. Bond Price Graph

Click **Return** to return to the Bond Analysis dialog box. Click **OK** to return to the Bond dialog box. Click **OK** to return to the Investment Analysis dialog box.

Generic Cashflow Tasks

To specify a generic cashflow, you merely define any sequence of date-amount pairs. The flexibility of generic cashflows enables the user to represent economic alternatives/investments that do not fit into loan, savings, depreciation, or bond specifications.

From the Investment Analysis dialog box, select **Investment** → **New** → **Generic Cashflow...** from the menu bar to open the **Generic Cashflow** dialog box. Enter RETAIL for the **Name** as in Figure 8.17.

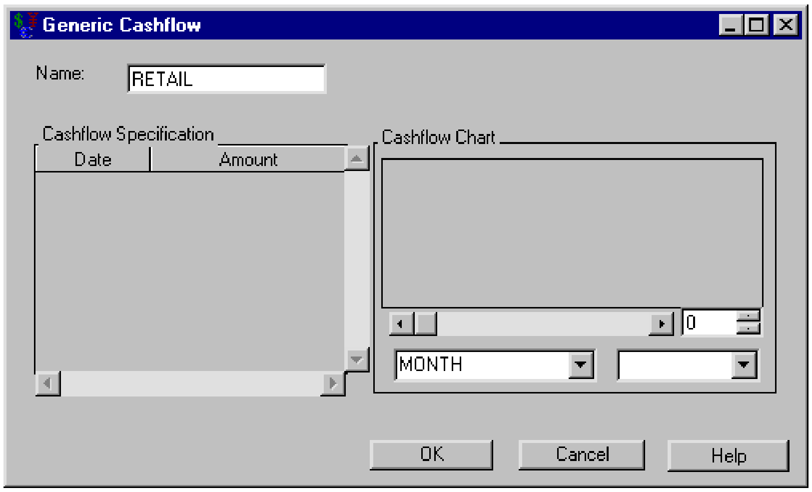

Figure 8.17. Introducing the Generic Cashflow

Right-Clicking within the Cashflow Specification Area

Right-clicking within Generic Cashflow's **Cashflow Specification** area reveals the pop-up menu displayed in Figure 8.18. The menu provides many useful tools to assist you in creating these date-amount pairs.

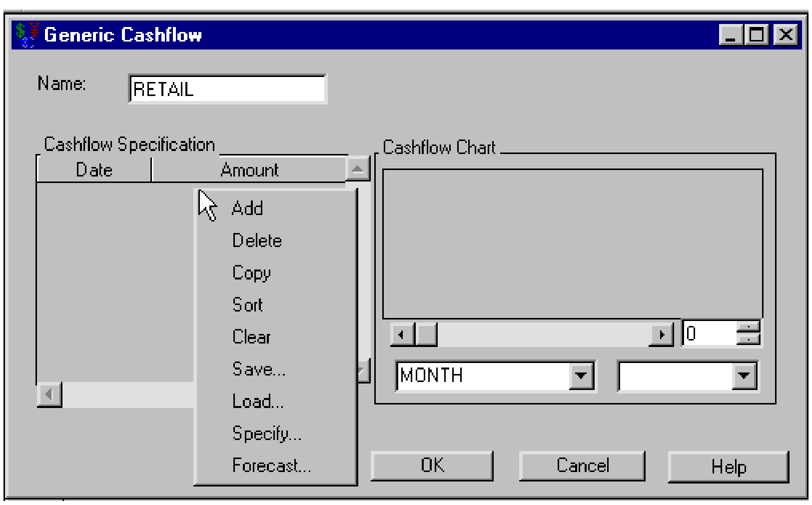

Figure 8.18. Right-Clicking within the Cashflow Specification Area

The following sections describe how to use most of these right-click options. The **Specify...** and **Forecast...** options are described in "Including a Generated Cashflow" and "Including a Forecasted Cashflow".

Adding a New Date-Amount Pair

To add a new date-amount pair manually, follow these steps:

1. Right-click in the **Cashflow Specification** area as shown in Figure 8.18, and release on **Add**.
2. Enter 01JAN01 for the date.
3. Enter 100 for the amount.

Copying a Date-Amount Pair

To copy a selected date-amount pair, follow these steps:

1. Select the pair you just created.
2. Right-click in the **Cashflow Specification** area as shown in Figure 8.18, but this time release on **Copy**.

Sorting All of the Date-Amount Pairs

Change the second date to 01JAN00. Now the dates are unsorted. Right-click in the **Cashflow Specification** area as shown in Figure 8.18, and release on **Sort**.

Deleting a Date-Amount Pair

To delete a selected date-amount pair, follow these steps:

1. Select a date-amount pair.
2. Right-click in the **Cashflow Specification** area as shown in Figure 8.18, and release on **Delete**.

Clearing All of the Date-Amount Pairs

To clear all date-amount pairs, right-click in the **Cashflow Specification** area as shown in Figure 8.18, and release on **Clear**.

Loading Date-Amount Pairs from a Dataset

To load date-amount pairs from a dataset into the **Cashflow Specification** area, follow these steps:

1. Right-click in the **Cashflow Specification** area and release on **Load...**. This opens the Load Dataset dialog box.
2. Enter SASHELP.RETAIL for **Dataset Name**.
3. Click **OK** to return to the Generic Cashflow dialog box.

If there is a **Date** variable in the dataset, Investment Analysis loads it into the list. If there is no **Date** variable, it loads the first available date or datetime-formatted variable. Investment Analysis then searches the dataset for an **Amount** variable to use. If none exists, it takes the first numeric variable that is not used by the **Date** variable.

Saving Date-Amount Pairs to a Dataset

To save date-amount pairs from the **Cashflow Specification** area to a dataset, follow these steps:

1. Right-click in the **Cashflow Specification** area and release on **Save...**. This opens the Save Dataset dialog box.
2. Enter the name of the dataset for **Dataset Name**.
3. Click **OK** to return to the Generic Cashflow dialog box.

Including a Generated Cashflow

To generate date-amount pairs for the **Cashflow Specification** area, follow these steps:

1. Right-click in the **Cashflow Specification** area and release on **Specify...**. This opens the **Flow Specification** dialog box.
2. Select **YEAR** for the **Time Interval**.
3. Enter today's date for the **Starting Date**.
4. Enter 10 for the **Number of Periods**. The **Ending Date** updates.
5. Enter 100 for the level. You can visualize the specification in the Cashflow Chart area (see Figure 8.19).
6. Click **Add** to add the specified cashflow to the list in the Generic Cashflow dialog box. Clicking **Add** also returns you to the Generic Cashflow dialog box.

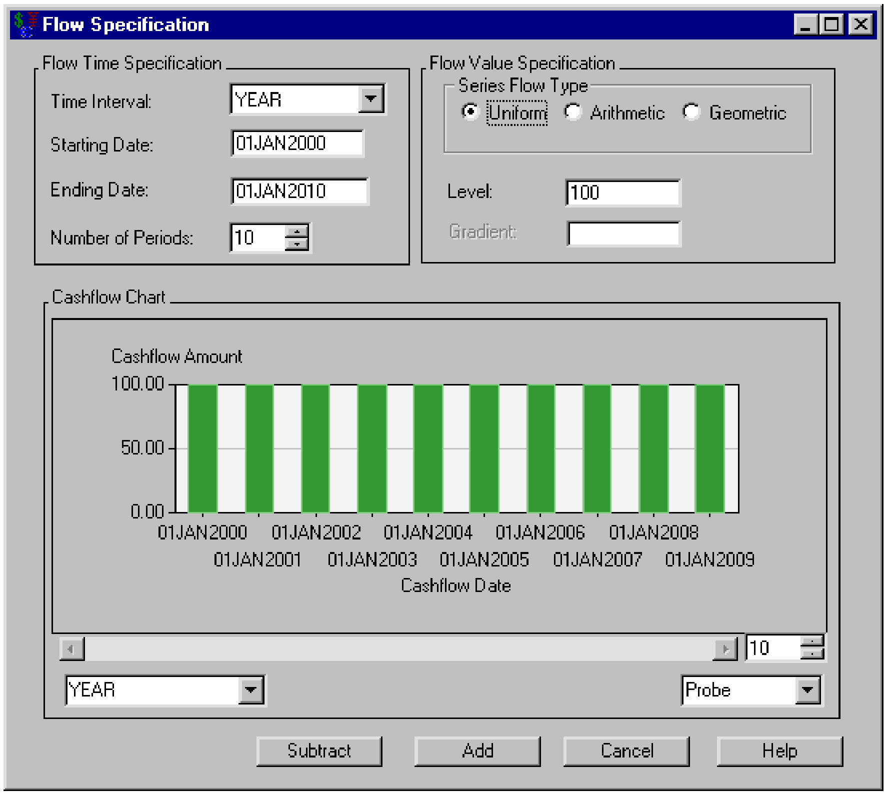

Figure 8.19. Uniform Cashflow Specification

Clicking **Subtract** will subtract the current cashflow from the Generic Cashflow dialog box as it returns you to the Generic Cashflow dialog box.

You can generate arithmetic and geometric specifications by clicking them within the **Series Flow Type** area. However, you must enter a value for the **Gradient**. In both cases the **Level** value is the value of the list at the **Starting Date**. With an arithmetic flow type, entries increment by the value **Gradient** each **Time Interval**. With a geometric flow type, entries increase by the factor **Gradient** each **Time Interval**. Figure 8.20 displays an arithmetic cashflow with a **Level** of 100 and a **Gradient** of 12.

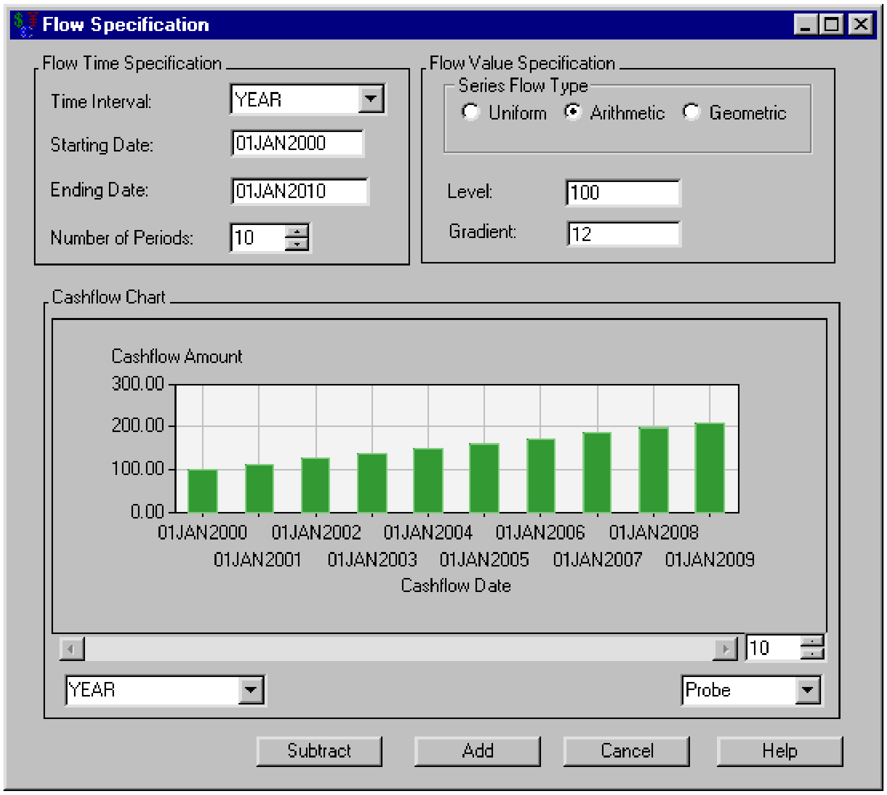

Figure 8.20. Arithmetic Cashflow Specification

Including a Forecasted Cashflow

To generate date-amount pairs for the **Cashflow Specification** area, follow these steps:

1. Right-click in the **Cashflow Specification** area and release on **Forecast...**. This opens the **Forecast Specification** dialog box.
2. Enter SASHELP.RETAIL as the **Data Set**.
3. Select SALES for the **Analysis Variable**.
4. Click **Compute Forecast** to generate the forecast. You can visualize the forecast in the Cashflow Chart area (see Figure 8.21).
5. Click **Add** to add the forecast to the list in the Generic Cashflow dialog box. Clicking **Add** also returns you to the Generic Cashflow dialog box.

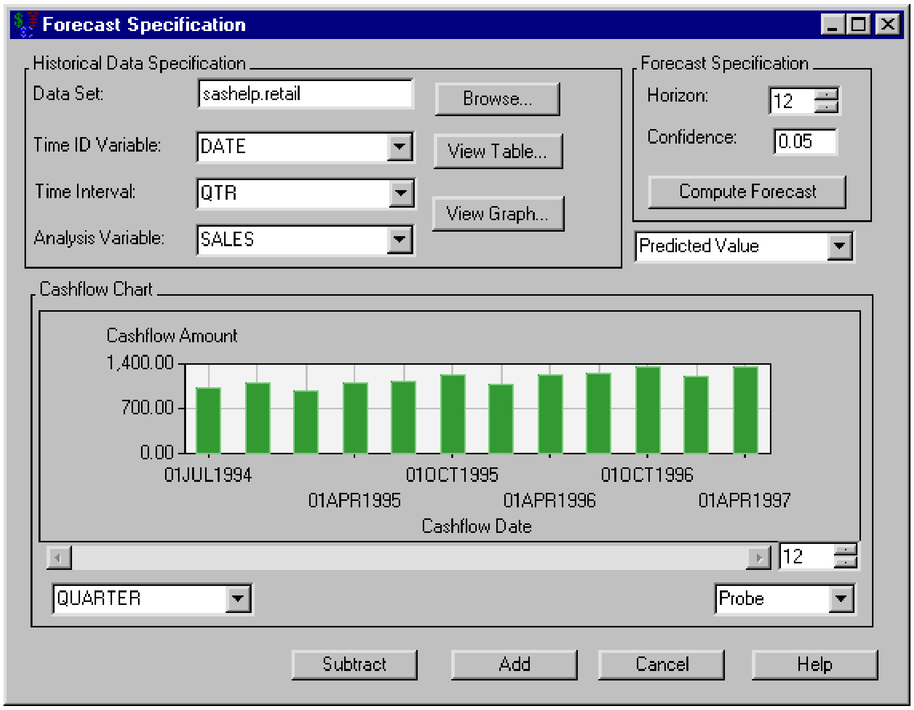

Figure 8.21. Cashflow Forecast

Clicking **Subtract** will subtract the current forecast from the Generic Cashflow dialog box as it returns you to the Generic Cashflow dialog box.

To review the values from the dataset you forecast, click **View Table...** or **View Graph...**.

You can adjust the following values for the dataset you forecast: **Time ID Variable**, **Time Interval**, and **Analysis Variable**.

You can adjust the following values for the forecast: the **Horizon**, the **Confidence**, and choice of predicted value, lower confidence limit, and upper confidence limit.

Using the Cashflow Chart

Three dialog boxes contain the Cashflow Chart to aide in your visualization of cashflows: Generic Cashflow, Flow Specification, and Forecast Specification. Within this chart, you possess the following tools:

You can click on a bar in the plot and view its **Cashflow Date** and **Cashflow Amount**.

You can change the aggregation period of the view with the box in the lower left corner of the Cashflow Chart. You can take the quarterly sales figures from the previous example, select YEAR as the value for this box, and view the annual sales figures. You can change the number in the box to the right of the horizontal scroll bar to alter the number of entries you wish to view. The number in that box must be no greater than the number of entries in the cashflow list. Lessening this number has the effect of zooming in upon a portion of the cashflow. When the number is less than the number of entries in the cashflow list, you can use the scroll bar at the bottom of the chart to scroll through the chart.

Dialog Box Guide

Loan

Selecting **Investment** → **New** → **Loan...** from the Investment Analysis dialog box's menu bar opens the Loan dialog box displayed in Figure 8.22.

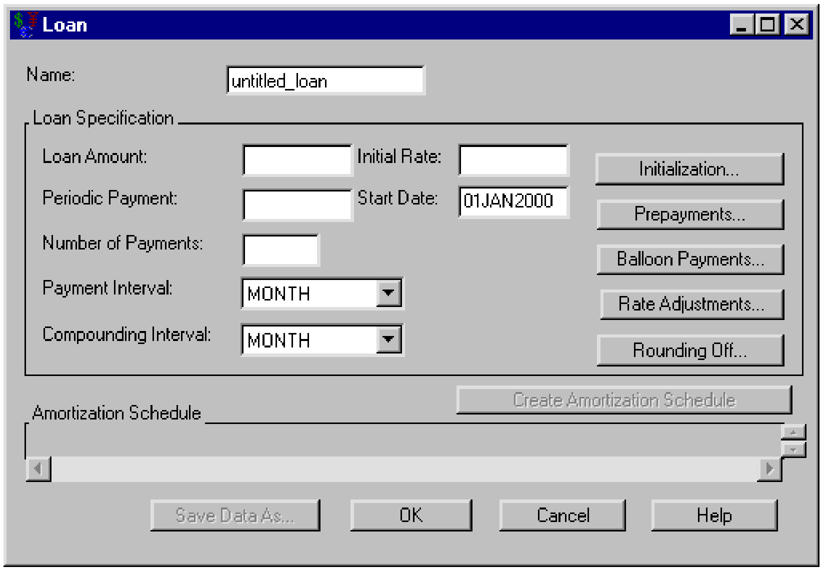

Figure 8.22. Loan Dialog Box

The following items are displayed:

Name holds the name you assign to the loan. You can set the name here or within the **Portfolio** area of the Investment Analysis dialog box. This must be a valid SAS name.

The **Loan Specification** area gives access to the values that define the loan.

Loan Amount holds the borrowed amount.

Periodic Payment holds the value of the periodic payments.

Number of Payments holds the number of payments in loan terms.

Payment Interval holds the frequency of the **Periodic Payment**.

Compounding Interval holds the compounding frequency.

Initial Rate holds the interest rate (a nominal percentage between 0 and 120) you pay on the loan.

Start Date holds the SAS date when the loan is initialized. The first payment is due one **Payment Interval** after this time.

Initialization... opens the Loan Initialization Options dialog box where you can define initialization costs and down-payments relevant to the loan.

Prepayments... opens the Loan Prepayments dialog box where you can specify the SAS dates and amounts of any prepayments.

Balloon Payments... opens the Balloon Payments dialog box where you can specify the SAS dates and amounts of any balloon payments.

Rate Adjustments... opens the Rate Adjustment Terms dialog box where you can specify terms for a variable-rate loan.

Rounding Off... opens the Rounding Off dialog box where you can select the number of decimal places for calculations.

Create Amortization Schedule becomes available when you adequately define the loan within the **Loan Specification** area. Clicking it generates the amortization schedule.

Amortization Schedule fills when you click **Create Amortization Schedule**. The schedule contains a row for the loan's start-date and each payment-date with information about the following:

Date is a SAS date, either the loan's start-date or a payment-date.

Beginning Principal Amount is the balance at that date.

Periodic Payment Amount is the expected payment at that date.

Interest Payment is zero for the loan's start-date; otherwise it holds the interest since the previous date.

Principal Repayment is the amount of the payment that went toward the principal.

Ending Principal is the balance at the end of the payment interval.

Save Data As... becomes available when you generate the amortization schedule. Clicking it opens the Save Output Dataset dialog box where you can save the amortization table (or portions thereof) as a SAS Dataset.

OK returns you to the Investment Analysis dialog box. If this is a new loan specification, clicking **OK** appends the current loan specification to the portfolio. If this is an existing loan specification, clicking **OK** returns the altered loan specification to the portfolio.

Cancel returns you to the Investment Analysis dialog box. If this is a new loan specification, clicking **Cancel** discards the current loan specification. If this is an existing loan specification, clicking **Cancel** discards the current editions.

Loan Initialization Options

Clicking **Initialization...** in the Loan dialog box opens the Loan Initialization Options dialog box displayed in Figure 8.23.

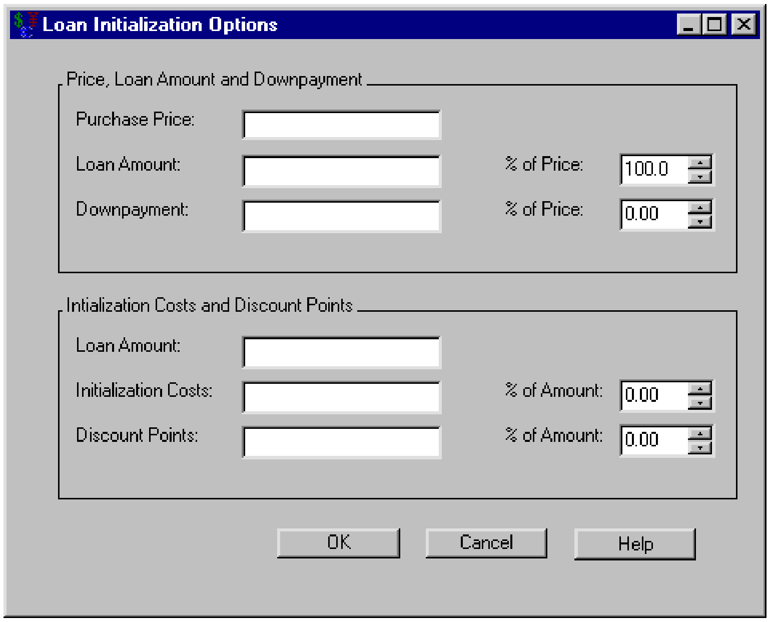

Figure 8.23. Loan Initialization Options Dialog Box

The following items are displayed:

The **Price, Loan Amount and Downpayment** area

> **Purchase Price** holds the actual price of the asset. This value equals the loan amount plus the downpayment.
>
> **Loan Amount** holds the loan amount.
>
> **% of Price** (to the right of **Loan Amount**) updates when you enter the **Purchase Price** and either the **Loan Amount** or **Downpayment**. This holds the percentage of the **Purchase Price** that comprises the **Loan Amount**. Setting the percentage manually causes the **Loan Amount** and **Downpayment** to update.
>
> **Downpayment** holds any downpayment paid for the asset.
>
> **% of Price** (to the right of **Downpayment**) updates when you enter the **Purchase Price** and either the **Loan Amount** or **Downpayment**. This holds the percentage of the **Purchase Price** that comprises the **Downpayment**. Setting the percentage manually causes the **Loan Amount** and **Downpayment** to update.

Initialization Costs and Discount Points area

> **Loan Amount** holds a copy of the **Loan Amount** above.
>
> **Initialization Costs** holds the value of any initialization costs.

% of Amount (to the right of **Initialization Costs**) updates when you enter the **Purchase Price** and either the **Initialization Costs** or **Discount Points**. This holds the percentage of the **Loan Amount** that comprises the **Initialization Costs**. Setting the percentage manually causes the **Initialization Costs** to update.

Discount Points holds the value of any discount points.

% of Amount (to the right of **Discount Points**) updates when you enter the **Purchase Price** and either the **Initialization Costs** or **Discount Points**. This holds the percentage of the **Loan Amount** that comprises the **Discount Points**. Setting the percentage manually causes the **Discount Points** to update.

OK returns you to the Loan dialog box, saving the information that is entered.

Cancel returns you to the Loan dialog box, discarding any editions since you opened the dialog box.

Loan Prepayments

Clicking **Prepayments...** in the Loan dialog box opens the Loan Prepayments dialog box displayed in Figure 8.24.

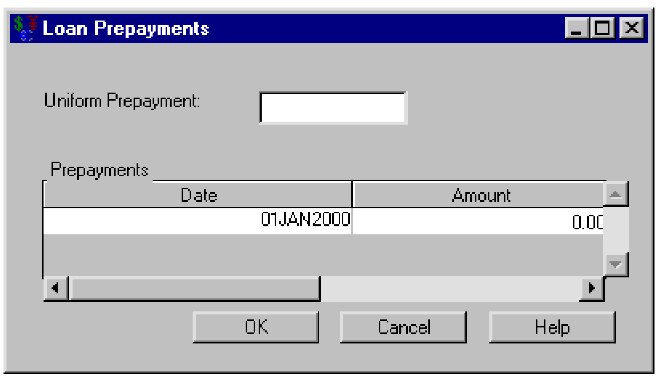

Figure 8.24. Loan Prepayments Dialog Box

The following items are displayed:

Uniform Prepayment holds the value of a regular prepayment concurrent to the usual periodic payment.

Prepayments holds a list of date-amount pairs to accommodate any prepayments. Right-clicking within the **Prepayments** area reveals many helpful tools for managing date-amount pairs.

OK returns you to the Loan dialog box, storing the information entered on the prepayments.

Cancel returns you to the Loan dialog box, discarding any prepayments entered since you opened the dialog box.

Balloon Payments

Clicking **Balloon Payments...** in the Loan dialog box opens the Balloon Payments dialog box displayed in Figure 8.25.

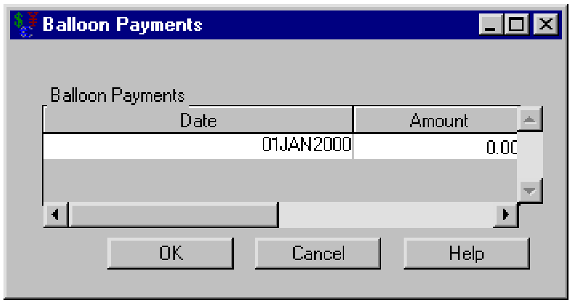

Figure 8.25. Balloon Payments Dialog Box

The following items are displayed:

Balloon Payments holds a list of date-amount pairs to accommodate any balloon payments. Right-clicking within the **Balloon Payments** area reveals many helpful tools for managing date-amount pairs.

OK returns you to the Loan dialog box, storing the information entered on the balloon payments.

Cancel returns you to the Loan dialog box, discarding any balloon payments entered since you opened the dialog box.

Rate Adjustment Terms

Clicking **Rate Adjustments...** in the Loan dialog box opens the Rate Adjustment Terms dialog box displayed in Figure 8.26.

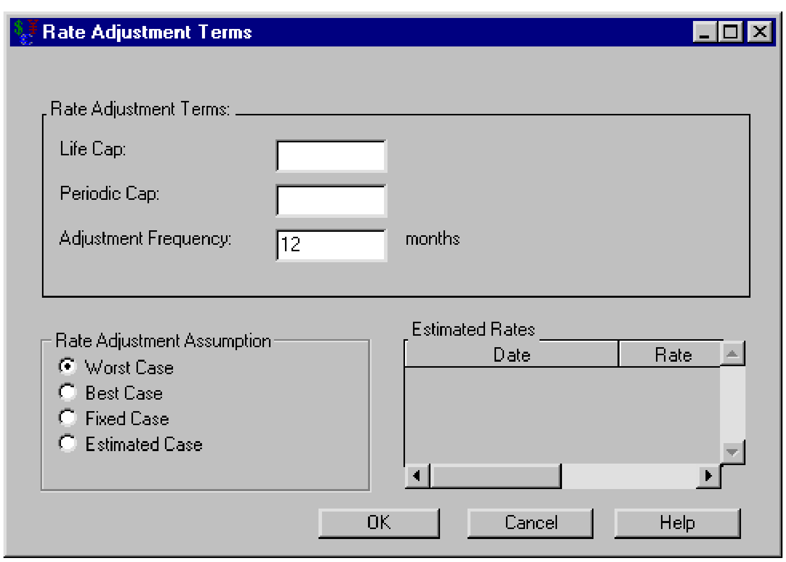

Figure 8.26. Rate Adjustment Terms Dialog Box

The following items are displayed:

The **Rate Adjustment Terms** area

> **Life Cap** holds the maximum deviation from the **Initial Rate** allowed over the life of the loan.
>
> **Periodic Cap** holds the maximum adjustment allowed per adjustment.
>
> **Adjustment Frequency** holds how often (in months) the lender can adjust the interest rate.

The **Rate Adjustment Assumption** determines the scenario the adjustments will take.

> **Worst Case** uses information from the **Rate Adjustment Terms** area to forecast a worst-case scenario.
>
> **Best Case** uses information from the **Rate Adjustment Terms** area to forecast a best-case scenario.
>
> **Fixed Case** specifies a fixed-rate loan.
>
> **Estimated Case** uses information from the **Rate Adjustment Terms** and **Estimated Rate** area to forecast a best-case scenario.

Estimated Rates holds a list of date-rate pairs, where each date is a SAS date and the rate is a nominal percentage between 0 and 120. The **Estimated Case** assumption uses these rates for its calculations. Right-clicking within the **Estimated Rates** area reveals many helpful tools for managing date-rate pairs.

OK returns you to the Loan dialog box, taking rate adjustment information into account.

Cancel returns you to the Loan dialog box, discarding any rate adjustment information provided since opening the dialog box.

Rounding Off

Clicking **Rounding Off...** in the Loan dialog box opens the Rounding Off dialog box displayed in Figure 8.27.

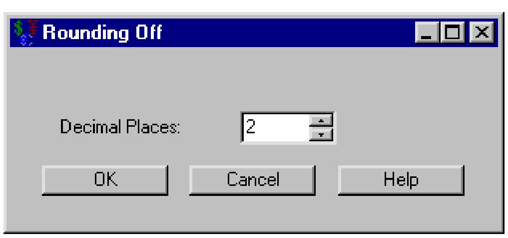

Figure 8.27. Rounding Off Dialog Box

The following items are displayed:

Decimal Places fixes the number of decimal places your results will display.

OK returns you to the Loan dialog box. Numeric values will then be represented with the number of decimals specified in **Decimal Places**.

Cancel returns you to the Loan dialog box. Numeric values will be represented with the number of decimals specified prior to opening this dialog box.

Savings

Selecting **Investment** → **New** → **Savings...** from the Investment Analysis dialog box's menu bar opens the Savings dialog box displayed in Figure 8.28.

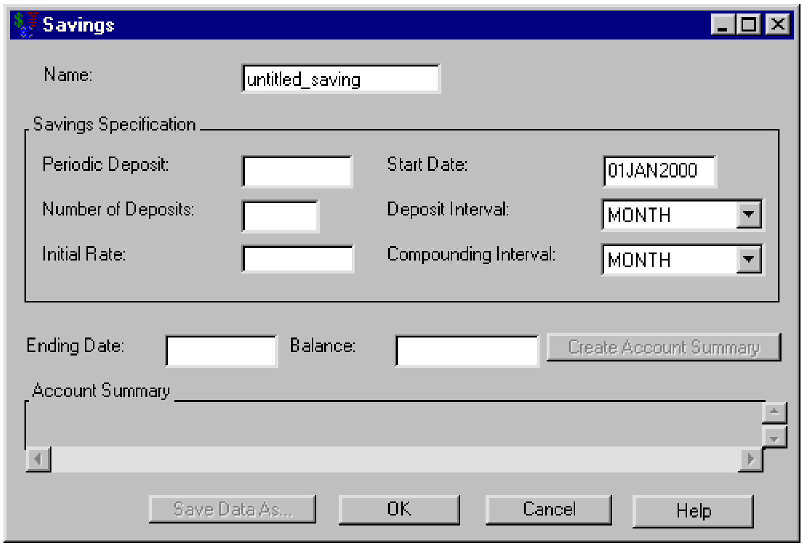

Figure 8.28. Savings Dialog Box

The following items are displayed:

Name holds the name you assign to the savings. You can set the name here or within the **Portfolio** area of the Investment Analysis dialog box. This must be a valid SAS name.

The **Savings Specification** area

> **Periodic Deposit** holds the value of your regular deposits.
>
> **Number of Deposits** holds the number of deposits into the account.
>
> **Initial Rate** holds the interest rate (a nominal percentage between 0 and 120) the savings account earns.
>
> **Start Date** holds the SAS date when deposits begin.
>
> **Deposit Interval** holds the frequency of your **Periodic Deposit**.
>
> **Compounding Interval** holds how often the interest compounds.

Create Account Summary becomes available when you adequately define the savings within the **Savings Specification** area. Clicking it generates the account summary.

Account Summary fills when you click **Create Account Summary**. The schedule contains a row for each deposit-date with information about the following:

Date is the SAS date of a deposit.

Starting Balance is the balance at that date.

Deposits is the deposit at that date.

Interest Earned is the interest earned since the previous date.

Ending Balance is the balance after the payment.

Save Data As... becomes available when you generate an account summary. Clicking it opens the Save Output Dataset dialog box where you can save the account summary (or portions thereof) as a SAS Dataset.

OK returns you to the Investment Analysis dialog box. If this is a new savings, clicking **OK** appends the current savings specification to the portfolio. If this is an existing savings specification, clicking **OK** returns the altered savings to the portfolio.

Cancel returns you to the Investment Analysis dialog box. If this is a new savings, clicking **Cancel** discards the current savings specification. If this is an existing savings, clicking **Cancel** discards the current editions.

Depreciation

Selecting **Investment** → **New** → **Depreciation...** from the Investment Analysis dialog box's menu bar opens the Depreciation dialog box displayed in Figure 8.29.

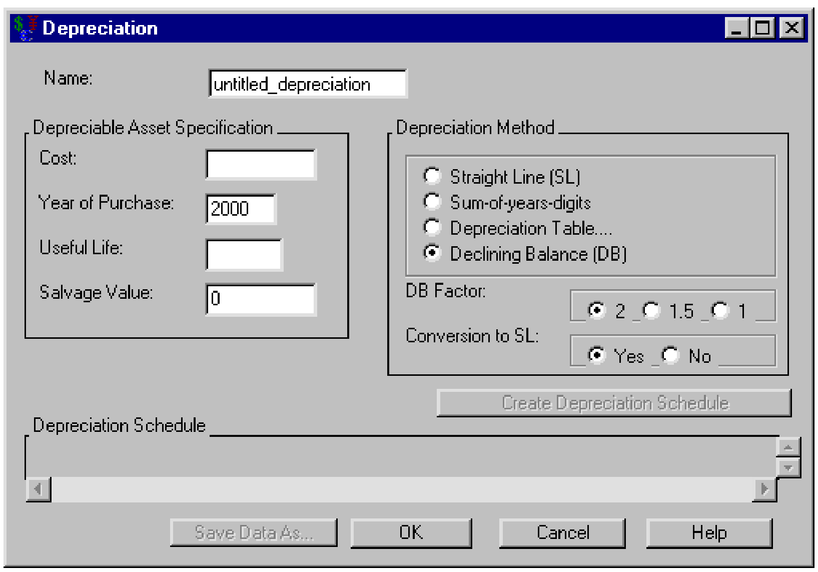

Figure 8.29. Depreciation Dialog Box

The following items are displayed:

Name holds the name you assign to the depreciation. You can set the name here or within the **Portfolio** area of the Investment Analysis dialog box. This must be a valid SAS name.

Depreciable Asset Specification

Cost holds the asset's original cost.

Year of Purchase holds the asset's year of purchase.

Useful Life holds the asset's useful life (in years).

Salvage Value holds the asset's value at the end of its **Useful Life**.

The **Depreciation Method** area holds the depreciation methods available:

- Straight Line
- Sum-of-years Digits
- Depreciation Table...
- Declining Balance
  - DB Factor: choice of 2, 1.5, or 1
  - Conversion to SL: choice of Yes or No

Create Depreciation Schedule becomes available when you adequately define the depreciation within the **Depreciation Asset Specification** area. Clicking the **Create Depreciation Schedule** button then fills the **Depreciation Schedule** area.

Depreciation Schedule fills when you click **Create Depreciation Schedule**. The schedule contains a row for each year. Each row holds:

Year is a year

Start Book Value is the starting book value for that year

Depreciation is the depreciation value for that year

End Book Value is the ending book value for that year

Save Data As... becomes available when you generate the depreciation schedule. Clicking it opens the Save Output Dataset dialog box where you can save the depreciation table (or portions thereof) as a SAS Dataset.

OK returns you to the Investment Analysis dialog box. If this is a new depreciation specification, clicking **OK** appends the current depreciation specification to the portfolio. If this is an existing depreciation specification, clicking **OK** returns the altered depreciation specification to the portfolio.

Cancel returns you to the Investment Analysis dialog box. If this is a new depreciation specification, clicking **Cancel** discards the current depreciation specification. If this is an existing depreciation specification, clicking **Cancel** discards the current editions.

Depreciation Table

Clicking **Depreciation Table...** from **Depreciation Method** area of the Depreciation dialog box opens the Depreciation Table dialog box displayed in Figure 8.30.

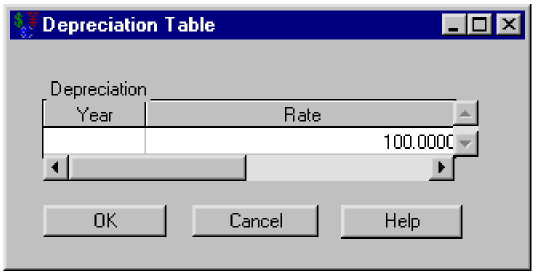

Figure 8.30. Depreciation Table Dialog Box

The following items are displayed:

The **Depreciation** area holds a list of year-rate pairs where the rate is an annual depreciation rate (a percentage between 0% and 100%). Right-clicking within the **Depreciation** area reveals many helpful tools for managing year-rate pairs.

OK returns you to the Depreciation dialog box with the current list of depreciation rates from the **Depreciation** area.

Cancel returns you to the Depreciation dialog box, discarding any editions to the **Depreciation** area since you opened the dialog box.

Bond

Selecting **Investment** → **New** → **Bond...** from the Investment Analysis dialog box's menu bar opens the Bond dialog box displayed in Figure 8.31.

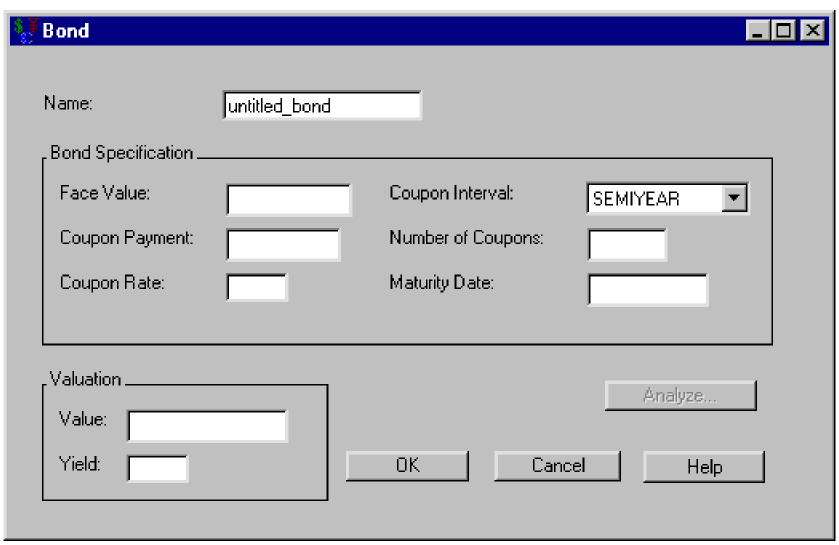

Figure 8.31. Bond

The following items are displayed:

Name holds the name you assign to the bond. You can set the name here or within the **Portfolio** area of the Investment Analysis dialog box. This must be a valid SAS name.

Bond Specification

> **Face Value** holds the bond's value at maturity.
>
> **Coupon Payment** holds the amount of money you receive periodically as the bond matures.
>
> **Coupon Rate** holds the rate (a nominal percentage between 0% and 120%) of the **Face Value** that defines the **Coupon Payment**.
>
> **Coupon Interval** holds how often the bond pays its coupons.
>
> **Number of Coupons** holds the number of coupons before maturity.
>
> **Maturity Date** holds the SAS date when you can redeem the bond for its **Face Value**.

The **Valuation** area becomes available when you adequately define the bond within the **Bond Specification** area. Entering either the **Value** or the **Yield** causes the calculation of the other. If you respecify the bond after performing a calculation here, you must reenter the **Value** or **Yield** value to update the calculation.

> **Value** holds the bond's value if expecting the specified **Yield**.
>
> **Yield** holds the bond's yield if the bond is valued at the amount of **Value**.

You must specify the bond before analyzing it. Once you have specified the bond, clicking **Analyze...** opens the Bond Analysis dialog box where you can compare various values and yields.

OK returns you to the Investment Analysis dialog box. If this is a new bond specification, clicking **OK** appends the current bond specification to the portfolio. If this is an existing bond specification, clicking **OK** returns the altered bond specification to the portfolio.

Cancel returns you to the Investment Analysis dialog box. If this is a new bond specification, clicking **Cancel** discards the current bond specification. If this is an existing bond specification, clicking **Cancel** discards the current editions.

Bond Analysis

Clicking **Analyze...** from the Bond dialog box opens the Bond Analysis dialog box displayed in Figure 8.32.

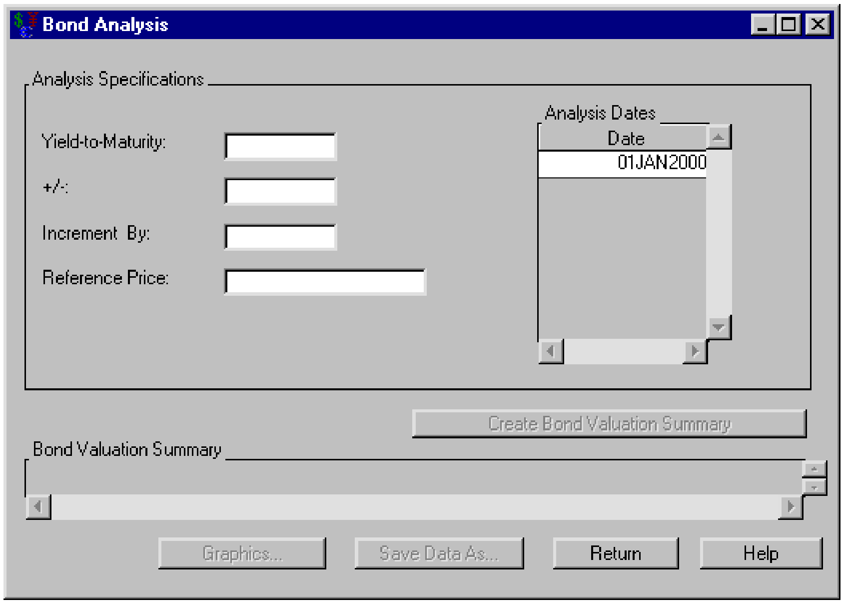

Figure 8.32. Bond Analysis

The following items are displayed:

Analysis Specifications

Yield-to-maturity holds the percentage yield upon which to center the analysis.

+/- holds the maximum deviation percentage from the **Yield-to-maturity** to consider.

Increment by holds the percentage increment by which the analysis is calculated.

Reference Price holds the reference price.

Analysis Dates holds a list of SAS dates for which you perform the bond analysis.

You must specify the analysis before valuing the bond for the various yields. Once you adequately specify the analysis, click **Create Bond Valuation Summary** to generate the bond valuation summary.

Bond Valuation Summary fills when you click **Create Bond Valuation Summary**. The schedule contains a row for each rate with information concerning the following:

Date is the SAS date when the **Value** gives the particular **Yield**.

Yield is the percent yield that corresponds to the **Value** at the given **Date**.

Value is the value of the bond at **Date** for the given **Yield**.

Percent Change is the percent change if the **Reference Price** is specified.

Duration is the duration.

Convexity is the convexity.

Graphics... opens the Bond Price graph representing the price versus yield-to-maturity.

Save Data As... becomes available when you fill the **Bond Valuation Summary** area. Clicking it opens the Save Output Dataset dialog box where you can save the valuation summary (or portions thereof) as a SAS Dataset.

Return takes you back to the Bond dialog box.

Bond Price

Clicking **Graphics...** from the Bond dialog box opens the Bond Price dialog box displayed in Figure 8.33.

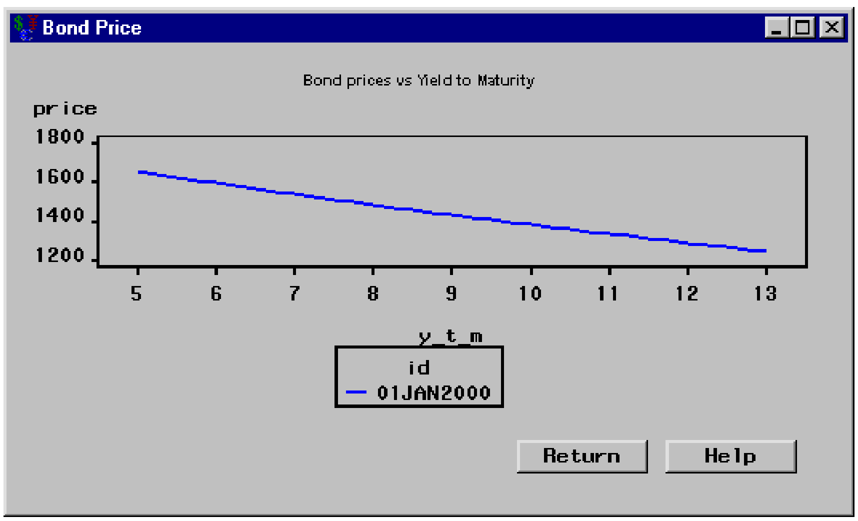

Figure 8.33. Bond Price Graph

It possesses the following item:

Return takes you back to the Bond Analysis dialog box.

Generic Cashflow

Selecting **Investment** → **New** → **Generic Cashflow...** from the Investment Analysis dialog box's menu bar opens the Generic Cashflow dialog box displayed in Figure 8.34.

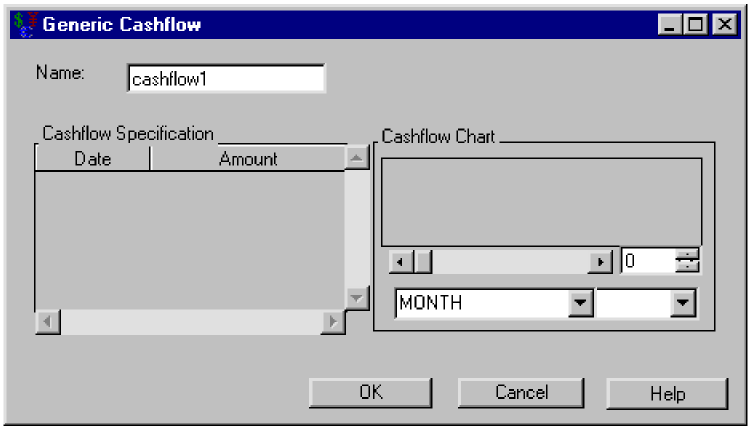

Figure 8.34. Generic Cashflow

The following items are displayed:

Name holds the name you assign to the generic cashflow. You can set the name here or within the **Portfolio** area of the Investment Analysis dialog box. This must be a valid SAS name.

Cashflow Specification holds date-amount pairs corresponding to deposits and withdrawals (or benefits and costs) for the cashflow. Each date is a SAS date. Right-clicking within the **Cashflow Specification** area reveals many helpful tools for managing date-amount pairs.

The **Cashflow Chart** fills with a graph representing the cashflow when the **Cashflow Specification** area is nonempty. The box to the right of the scroll bar controls the number of entries with which to fill the graph. If the number in this box is less than the total number of entries, you can use the scroll bar to view different segments of the cashflow. The left box below the scroll bar holds the frequency for drilling purposes.

OK returns you to the Investment Analysis dialog box. If this is a new generic cashflow specification, clicking **OK** appends the current cashflow specification to the portfolio. If this is an existing cashflow specification, clicking **OK** returns the altered cashflow specification to the portfolio.

Cancel returns you to the Investment Analysis dialog box. If this is a new cashflow specification, clicking **Cancel** discards the current cashflow specification. If this is an existing cashflow specification, clicking **Cancel** discards the current editions.

Right-Clicking within Generic Cashflow's Cashflow Specification Area

Right-click within the **Cashflow Specification** area of the Generic Cashflow dialog box pops up the menu displayed in Figure 8.35.

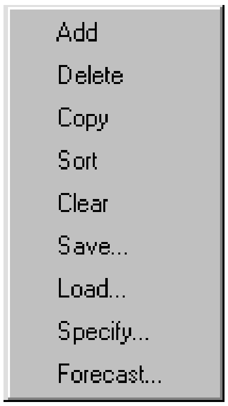

Figure 8.35. Right-Clicking

Add creates a blank pair.

Delete removes the currently highlighted pair.

Copy duplicates the currently selected pair.

Sort arranges the entered pairs in chronological order.

Clear empties the area of all pairs.

Save... opens the Save Dataset dialog box where you can save the entered pairs as a SAS Dataset for later use.

Load... opens the Load Dataset dialog box where you select a SAS Dataset to populate the area.

Specify... opens the **Flow Specification** dialog box where you can generate date-rate pairs to include in your cashflow.

Forecast... opens the **Forecast Specification** dialog box where you can generate the forecast of a dataset to include in your cashflow.

If you wish to perform one of these actions on a collection of pairs, you must select a collection of pairs before right-clicking. To select an adjacent list of pairs, do the following: click the first pair, hold down SHIFT, and click the final pair. Once the list of pairs is selected, you may release the SHIFT key.

Flow Specification

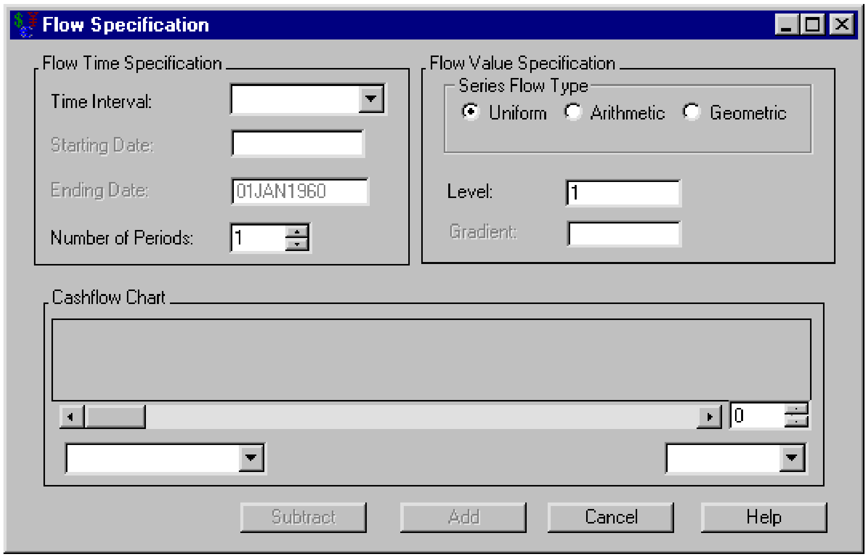

Figure 8.36. Flow Specification

The following items are displayed:

Flow Time Specification

Time Interval holds the uniform frequency of the entries.

You can set the **Starting Date** when you set the **Time Interval**. It holds the SAS date the entries will start.

You can set the **Ending Date** when you set the **Time Interval**. It holds the SAS date the entries will end.

Number of Periods holds the number of entries.

Flow Value Specification

Series Flow Type describes the movement the entries can assume:

- **Uniform** assumes all entries are equal.
- **Arithmetic** assumes the entries begin at **Level** and increase by the value of **Gradient** per entry.
- **Geometric** assumes the entries begin at **Level** and increase by a factor of **Gradient** per entry.

Level holds the starting amount for all **Flow Types**.

You can set the **Gradient** when you select either **Arithmetic** or **Geometric Gradient**. It holds the arithmetic and geometric gradients, respectively, for the **Arithmetic** and **Geometric Flow Types**.

The **Cashflow Chart** fills with a graph displaying the dates and values of the entries when the cashflow entries are adequately defined. The box to the right of the scroll bar controls the number of entries with which to fill the graph. If the number in this box is less than the total number of entries, you can use the scroll bar to view different segments of the cashflow. The left box below the scroll bar holds the frequency for drilling purposes.

Subtract becomes available when the collection of entries is adequately specified. Clicking **Subtract** then returns you to the Generic Cashflow dialog box subtracting the entries from the current cashflow.

Add becomes available when the collection of entries is adequately specified. Clicking **Add** then returns you to the Generic Cashflow dialog box adding the entries to the current cashflow.

Cancel returns you to Generic Cashflow dialog box without editing the cashflow.

Forecast Specification

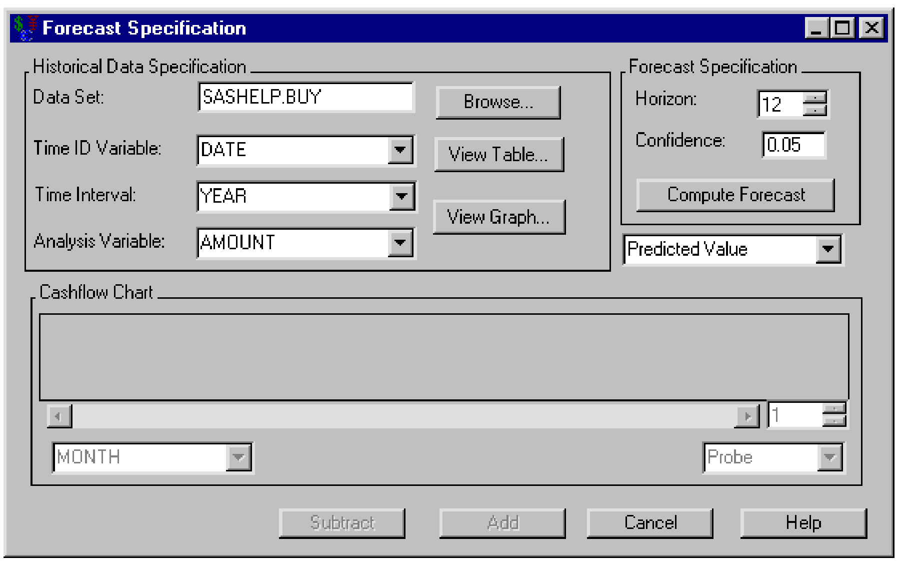

Figure 8.37. Forecast Specification

The following items are displayed:

Historical Data Specification

Data Set holds the name of the dataset to forecast.

Browse... opens the standard SAS **Open** dialog box to help select a dataset to forecast.

Time ID Variable holds the time ID variable to forecast over.

Time Interval fixes the time interval for the **Time ID Variable**.

Analysis Variable holds the data variable upon which to forecast.

View Table... opens a table that displays in a list the contents of the specified dataset.

View Graph... opens the Time Series Viewer that graphically displays the contents of the specified dataset.

Forecast Specification

Horizon holds the number of periods into the future you wish to forecast.

Confidence holds the confidence limit for applicable forecasts.

Compute Forecast fills the **Cashflow Chart** with the forecast.

The box below holds the type of forecast you wish to generate:

- Predicted Value
- Lower Confidence Limit
- Upper Confidence Limit

The **Cashflow Chart** fills when you click **Compute Forecast**. The box to the right of the scroll bar controls the number of entries with which to fill the graph. If the number in this box is less than the total number of entries, you can use the scroll bar to view different segments of the cashflow. The left box below the scroll bar holds the frequency for drilling purposes.

Subtract becomes available when the collection of entries is adequately specified. Clicking **Subtract** then returns you to the Generic Cashflow dialog box subtracting the forecast from the current cashflow.

Add becomes available when the collection of entries is adequately specified. Clicking **Add** then returns you to the Generic Cashflow adding the forecast to the current cashflow.

Cancel returns to Generic Cashflow dialog box without editing the cashflow.

Chapter 9
Computations

Chapter Table of Contents

Chapter 9
Computations

The Compute Menu

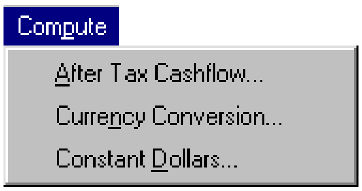

Figure 9.1. The Compute Menu

The **Compute** menu, shown in Figure 9.1, offers the following options to apply to generic cashflows:

Computing an **After Tax Cashflow** is useful when taxes affect investment alternatives differently. Comparing after tax cashflows provide a more accurate determination of the cashflows' profitabilities. You can set default values for income tax rates by selecting **Tools** → **Define Rate** → **Income Tax Rate...** from the Investment Analysis dialog box. This opens the Income Tax Specification dialog box where you can enter the tax rates.

Currency Conversion is necessary when investments are in different currencies. For data concerning currency conversion rates, consider http://dsbb.imf.org/, the International Monetary Fund's Dissemination Standards Bulletin Board.

A **Constant Dollar** (inflation adjusted monetary value) calculation takes a cashflow and inflation information and discounts the cashflow to a level where the buying power of the monetary unit is "constant" over time. Groups quantify inflation (in the form of price indices and inflation rates) for countries and industries by averaging the growth of prices for various products and sectors of the economy. For data concerning price indices, consider the United States Department of Labor at http://www.dol.gov/ and the International Monetary Fund's Dissemination Standards Bulletin Board at http://dsbb.imf.org/. You can set default values for inflation rates by clicking **Tools** → **Define Rate** → **Inflation...** from the Investment Analysis dialog box. This opens the Inflation Specification dialog box where you can enter the inflation rates.

Tasks

The next few sections show how to perform computations for the following situation. Suppose you buy a $10,000 certificate of deposit that pays 12% interest a year for five years. Your earnings are taxed at a rate of 30% federally and 7% locally. Also, you want to transfer all the money to an account in England. British pounds converts to

American dollars at an exchange rate of $1.00 to £0.60. The inflation rate in England is 3%. The instructions in this example assume familiarity with the following:

- The right-clicking options of the **Cashflow Specification** area in the Generic Cashflow dialog box (described in “Right-Clicking within Generic Cashflow’s Cashflow Specification Area” on page 237.)
- The **Save Data As...** button located in many dialog boxes (described in “Saving Output to SAS Datasets” on page 271.)

Taxing a Cashflow

Consider the example described in “The Compute Menu” on page 243. To create the earnings, follow these steps:

1. Select **Investment** → **New** → **Generic Cashflow** to create a generic cashflow.
2. Enter CD_INTEREST for the **Name**.
3. Enter 1200 for each of the five years starting one year from today as displayed in Figure 9.2.
4. Click **OK** to return to the Investment Analysis dialog box.

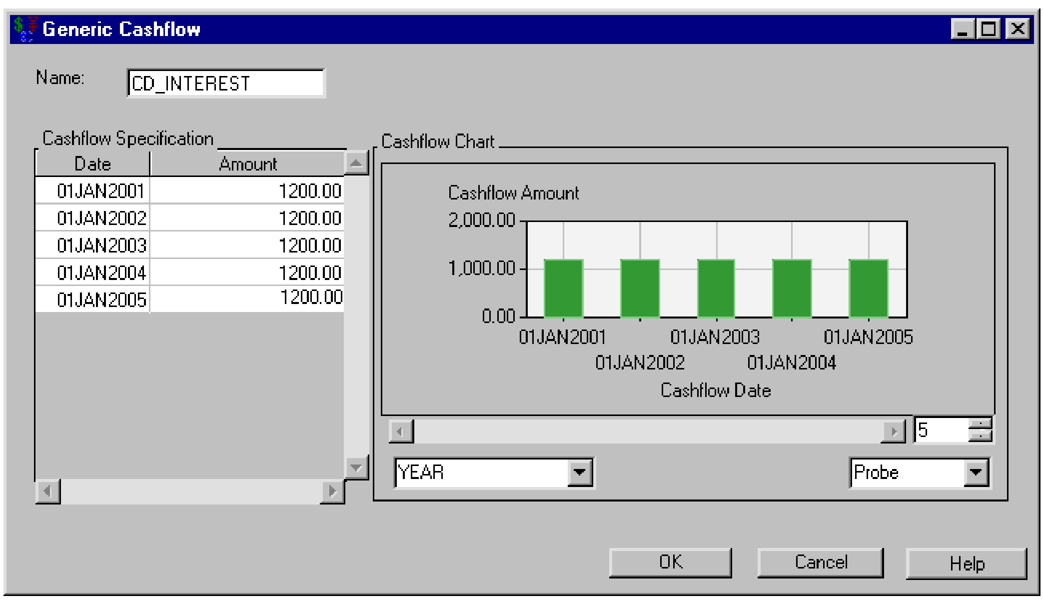

Figure 9.2. Computing the Interest on the CD

To compute the tax on the earnings, follow these steps:

1. Select CD_INTEREST from the **Portfolio** area.
2. Select **Compute** → **After Tax Cashflow** from the pull-down menu.
3. Enter 30 for **Federal Tax**.
4. Enter 7 for **Local Tax**. Note that **Combined Tax** updates.

5. Click **Create After Tax Cashflow** and the **After Tax Cashflow** area fills, as displayed in Figure 9.3.

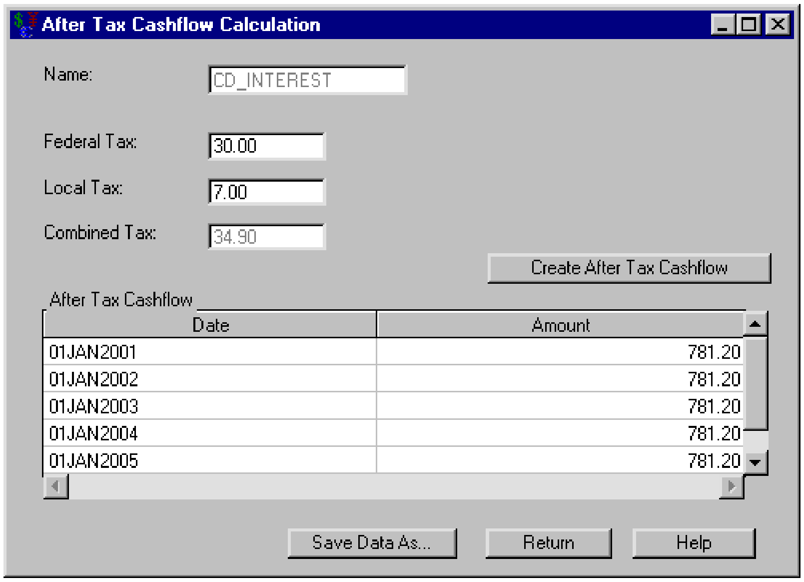

Figure 9.3. Computing the Interest After Taxes

Save the taxed earnings to a dataset named WORK.CD_AFTERTAX. Click **Return** to return to the Investment Analysis dialog box.

Converting Currency

Consider the example described in "The Compute Menu" on page 243. To create the cashflow to convert, follow these steps:

1. Select **Investment** → **New** → **Generic Cashflow...** to open a new generic cashflow.
2. Enter CD_DOLLARS for the **Name**.
3. Load WORK.CD_AFTERTAX into its **Cashflow Specification**.
4. Add -10,000 for today and +10,000 for five years from today to the cashflow as displayed in Figure 9.4.
5. Sort the transactions by date to aid your reading.
6. Click **OK** to return to the Investment Analysis dialog box.

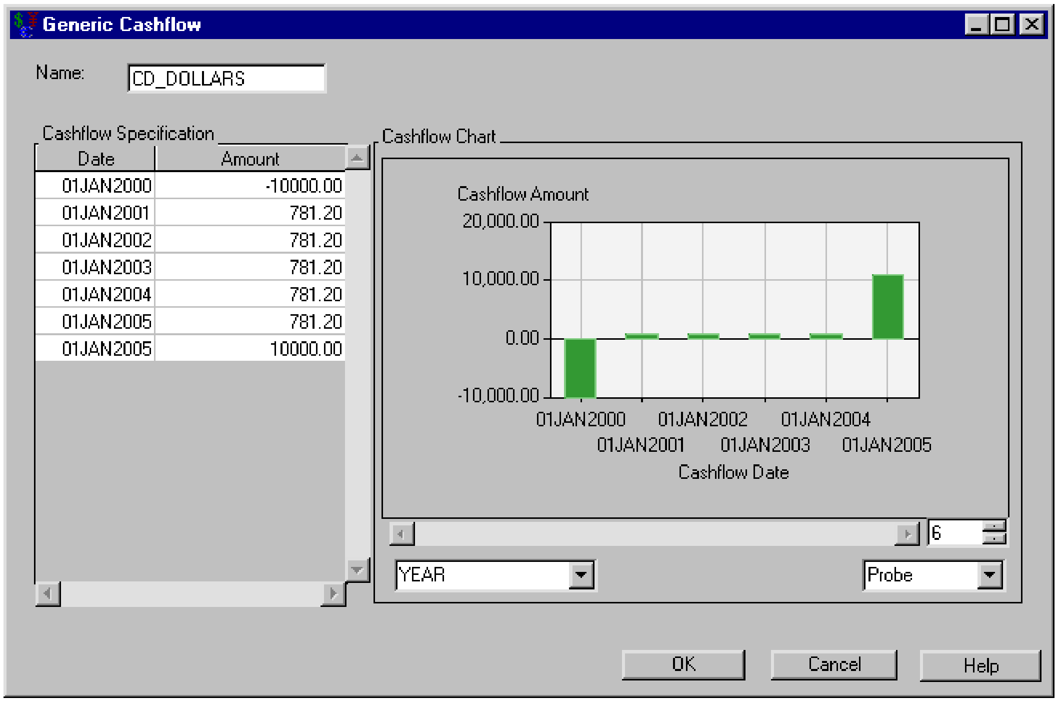

Figure 9.4. The CD in Dollars

To convert from British pounds to American dollars, follow these steps:

1. Select CD_DOLLARS from the portfolio.
2. Select **Compute** → **Currency Conversion** from the pull-down menu. This opens the Currency Conversion dialog box.
3. Select USD for the **From Currency**.
4. Select GBP for the **To Currency**.
5. Enter 0.60 for the **Exchange Rate**.
6. Click **Apply Currency Conversion** to fill the **Currency Conversion** area as displayed in Figure 9.5.

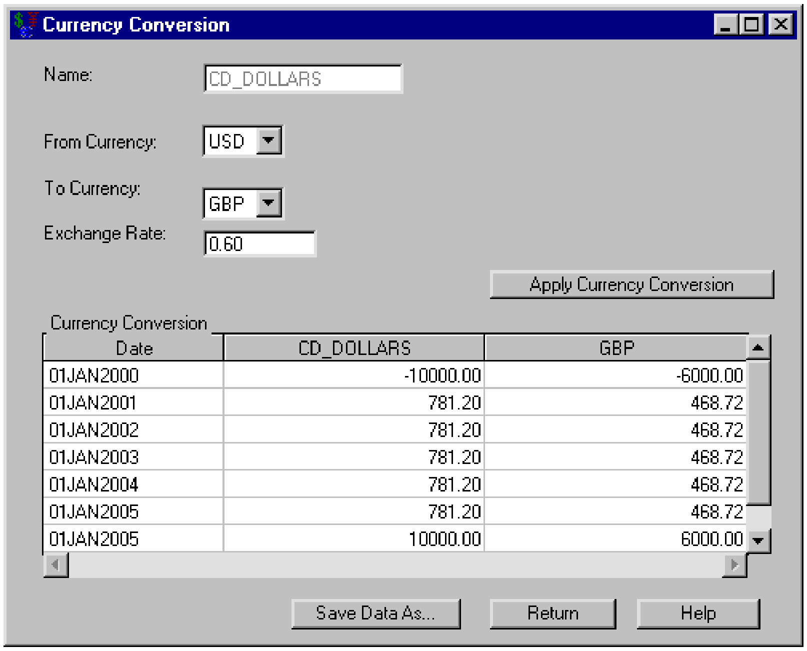

Figure 9.5. Converting the CD to Pounds

Save the converted values to a dataset named WORK.CD_POUNDS. Click **Return** to return to the Investment Analysis dialog box.

Inflating Cashflows

Consider the example described in “The Compute Menu” on page 243. To create the cashflow to deflate, follow these steps:

1. Select **Investment** → **New** → **Generic Cashflow...** to open a new generic cashflow.
2. Enter CD_DEFLATED for **Name**.
3. Load WORK.CD_POUNDS into its **Cashflow Specification** (see Figure 9.6).
4. Click **OK** to return to the Investment Analysis dialog box.

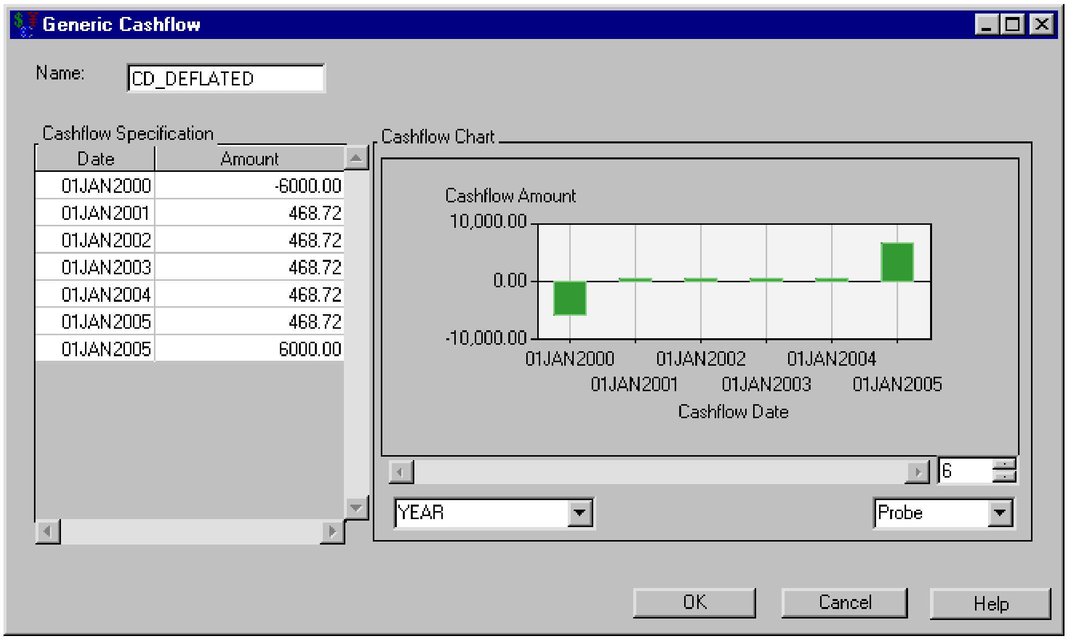

Figure 9.6. The CD before Deflation

To deflate the values, follow these steps:

1. Select CD_DEFLATED from the portfolio.
2. Select **Compute** → **Constant Dollars** from the pull-down menu. This opens the Constant Dollar Calculation dialog box.
3. Clear the **Variable Inflation List** area.
4. Enter 3 for the **Constant Inflation Rate**.
5. Click **Create Constant Dollar Equivalent** to generate a constant dollar equivalent summary (see Figure 9.7).

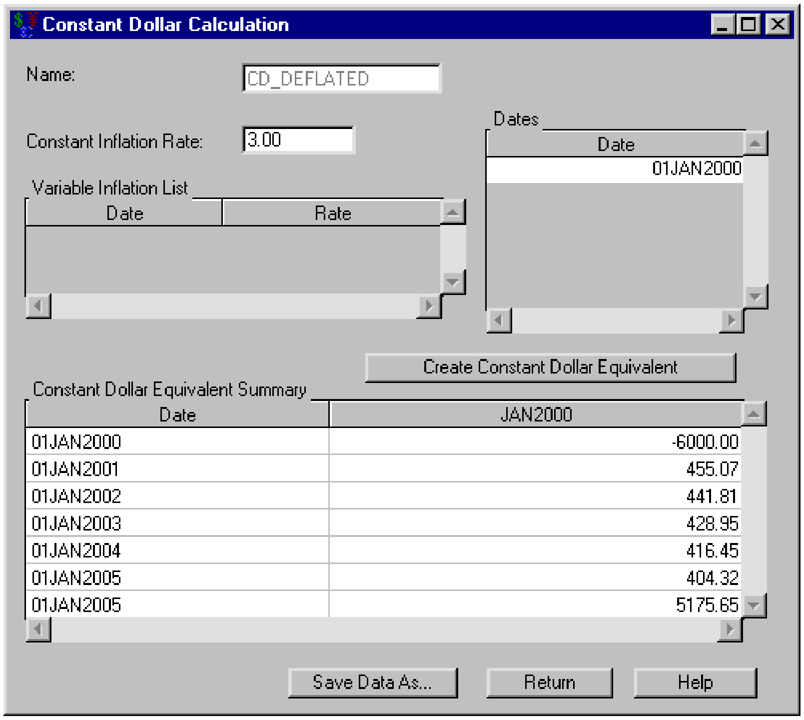

Figure 9.7. CD Values after Deflation

You can save the deflated cashflow to a dataset for use in an internal rate of return analysis or breakeven analysis.

Click **Return** to return to the Investment Analysis dialog box.

Dialog Box Guide

After Tax Cashflow Computation

Having selected a generic cashflow from the Investment Analysis dialog box, to perform an after tax calculation, select **Compute** → **After Tax...** from the Investment Analysis dialog box's menu bar. This opens the After Tax Cashflow Calculation dialog box displayed in Figure 9.8.

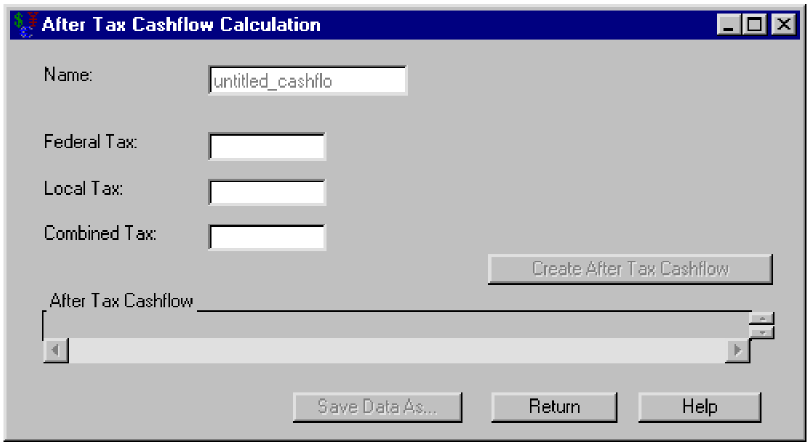

Figure 9.8. After Tax Cashflow Computation Dialog Box

The following items are displayed:

Name holds the name of the investment for which you are computing the after tax cashflow.

Federal Tax holds the federal tax rate (a percentage between 0% and 100%).

Local Tax holds the local tax rate (a percentage between 0% and 100%).

Combined Tax holds the effective tax rate from federal and local income taxes.

Create After Tax Cashflow becomes available when **Combined Tax** is non-empty. Clicking **Create After Tax Cashflow** then fills the **After Tax Cashflow** area.

After Tax Cashflow fills when you click **Create After Tax Cashflow**. It holds a list of date-amount pairs where the amount is the amount retained after taxes for that date.

Save Data As... becomes available when you fill the after tax cashflow. Clicking it opens the Save Output Dataset dialog box where you can save the resulting cashflow (or portions thereof) as a SAS Dataset.

Return returns you to the Investment Analysis dialog box.

Currency Conversion

Having selected a generic cashflow from the Investment Analysis dialog box, to perform a currency conversion, select **Compute** → **Currency Conversion...** from the Investment Analysis dialog box's menu bar. This opens the Currency Conversion dialog box displayed in Figure 9.9.

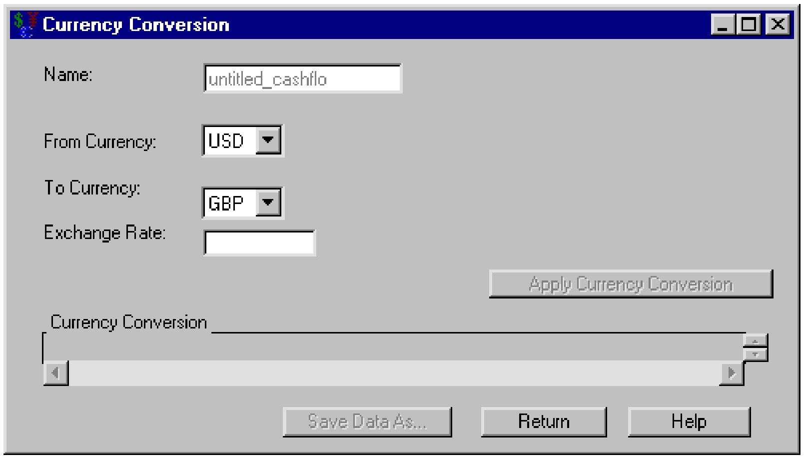

Figure 9.9. Currency Conversion Dialog Box

The following items are displayed:

Name holds the name of the investment to which you are applying the currency conversion.

From Currency holds the name of the currency the cashflow currently represents.

To Currency holds the name of the currency to which you wish to convert.

Exchange Rate holds the rate of exchange between the **From Currency** and the **To Currency**.

Apply Currency Conversion becomes available when you fill **Exchange Rate**. Clicking **Apply Currency Conversion** fills the **Currency Conversion** area.

Currency Conversion fills when you click **Apply Currency Conversion**. The schedule contains a row for each cashflow item with the following information:

- **Date** is a SAS date within the cashflow.
- The **From Currency** value is the amount in the original currency at that date.
- The **To Currency** value is the amount in the new currency at that date.

Save Data As... becomes available when you fill the **Currency Conversion** area. Clicking it opens the Save Output Dataset dialog box where you can save the conversion table (or portions thereof) as a SAS Dataset.

Return returns you to the Investment Analysis dialog box.

Constant Dollar Calculation

Having selected a generic cashflow from the Investment Analysis dialog box, to perform a constant dollar calculation, select **Compute** → **Constant Dollars...** from the Investment Analysis dialog box's menu bar. This opens the Constant Dollar Calculation dialog box displayed in Figure 9.10.

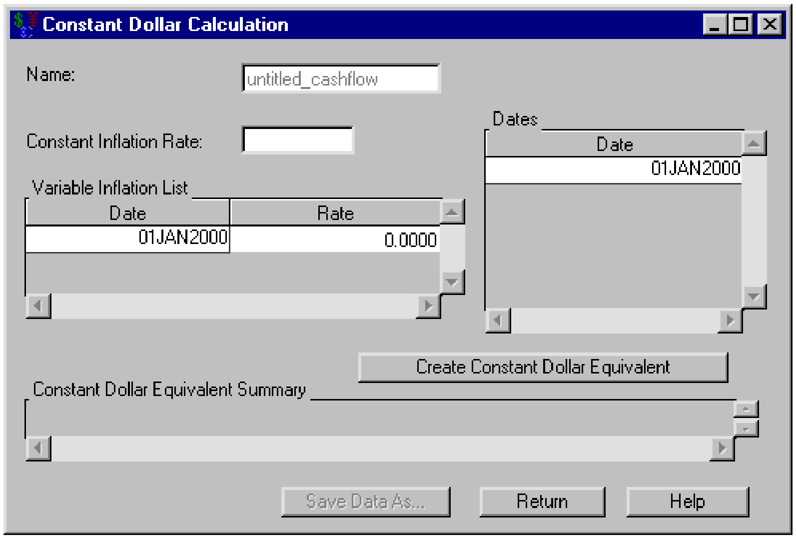

Figure 9.10. Constant Dollar Calculation Dialog Box

The following items are displayed:

Name holds the name of the investment for which you are computing the constant dollars value.

Constant Inflation Rate holds the constant inflation rate (a percentage between 0% and 120%). This value is used if the **Variable Inflation List** area is empty.

Variable Inflation List holds date-rate pairs that describe how inflation varies over time. Each date is a SAS date, and the rate is a percentage between 0% and 120%. Each date refers to when that inflation rate begins. Right-clicking within the **Variable Inflation** area reveals many helpful tools for managing date-rate pairs. If you assume a fixed inflation rate, just insert that rate in **Constant Rate**.

Dates holds the SAS date(s) at which you wish to compute the constant dollar equivalent. Right-clicking within the **Dates** area reveals many helpful tools for managing date lists.

Create Constant Dollar Equivalent becomes available when you enter inflation rate information. Clicking it fills the constant dollar equivalent summary with the computed constant dollar values.

Constant Dollar Equivalent Summary fills with a summary when you click **Create Constant Dollar Equivalent**. The first column lists the dates of the generic cashflow. The second column contains the constant dollar equivalent of the original generic cashflow item of that date.

Save Data As... becomes available when you fill the constant dollar equivalent summary. Clicking it opens the Save Output Dataset dialog box where you can save the Constant Dollar Equivalent Summary (or portions thereof) as a SAS Dataset.

Return returns you to the Investment Analysis dialog box.

Chapter 10
Analyses

Chapter Table of Contents

Chapter 10
Analyses

The Analyze Menu

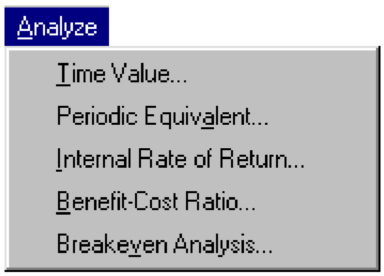

Figure 10.1. Analyze Menu

The **Analyze** menu, shown in Figure 10.1, offers the following options for use on applicable investments:

Time Value analysis involves moving money through time across a defined MARR so that you can compare value at a consistent date. The MARR can be constant or variable over time.

Uniform **Periodic Equivalent** analysis determines the payment needed to convert a cashflow to uniform amounts over time, given a periodicity, a number of periods, and a MARR. This option helps when making comparisons where one alternative is uniform (such as renting) and another is not (such as buying).

The **Internal Rate of Return** of a cashflow is the interest rate that makes the time value equal to 0. This calculation assumes uniform periodicity of the cashflow. It is particularly applicable where the choice of MARR would be difficult.

The **Benefit-Cost Ratio** divides the time value of the benefits by the time value of the costs. For example, governments often use this analysis when deciding whether to commit to a public works project.

Breakeven Analysis computes time values at various MARRs to compare. It is advantageous when it is difficult to determine a MARR. This helps you determine how the cashflow's profitability varies with your choice of MARR. A graph displaying the relationships between time value and MARR is also available.

Tasks

Performing Time Value Analysis

Suppose a rock quarry needs equipment to use the next five years. It has two alternatives:

- A box loader and conveyer system that has a one-time cost of $264,000.
- A two-shovel loader which costs $84,000 but has a yearly operating cost of $36,000. This loader has a service life of three years, which necessitates the purchase of a new loader for the final two years of the rock quarry project. Assume the second loader also costs $84,000 and its salvage value after its two-year service is $10,000. A SAS dataset that describes this is available at SASHELP.ROCKPIT.

You expect a 13% MARR. Which is the better alternative?

To create the cashflows, follow these steps:

1. Create a cashflow with the single amount -264,000. Date the amount 01JAN1998 to be consistent with the dataset you load.
2. Load SASHELP.ROCKPIT into a second cashflow, as displayed in Figure 10.2.

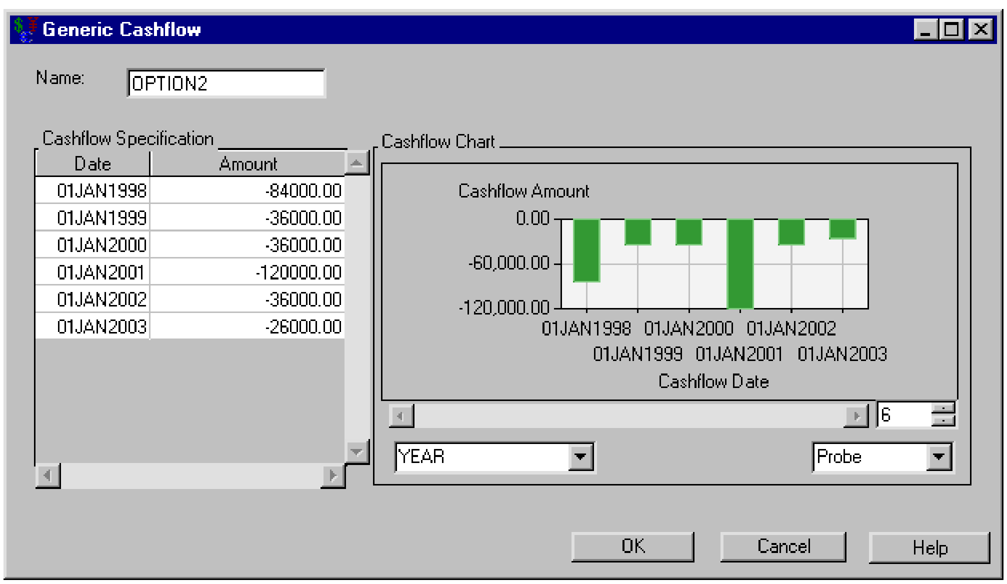

Figure 10.2. The contents of SASHELP.ROCKPIT

To compute the time values of these investments, follow these steps:

1. Select both cashflows.
2. Select **Analyze** → **Time Value...**. This opens the Time Value Analysis dialog box.

3. Enter the date 01JAN1998 into the **Dates** area.
4. Enter 13 for the **Constant MARR**.
5. Click **Create Time Value Summary**.

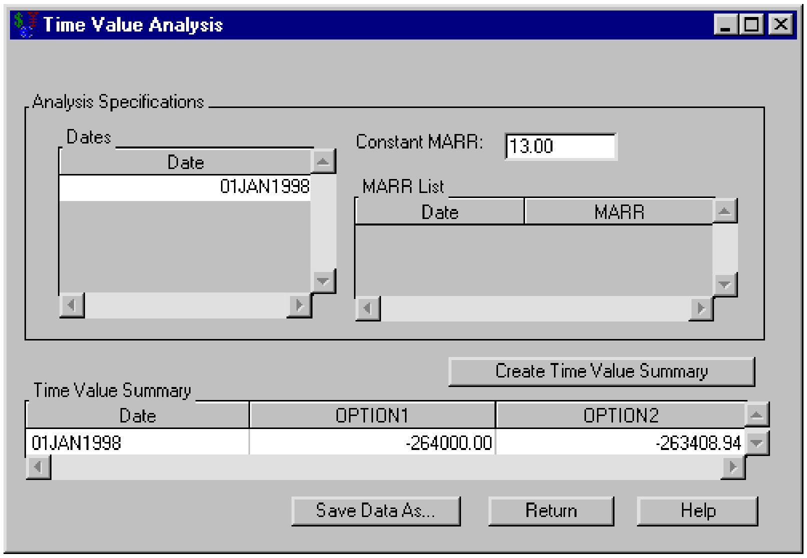

Figure 10.3. Performing the Time Value Analysis

As shown in Figure 10.3, option 1 has a time value of -$264,000.00 naturally on 01JAN1998. However, option 2 has a time value of -$263,408.94, which is slightly less expensive.

Computing an Internal Rate of Return

You are choosing between five investments. A portfolio containing these investments is available at SASHELP.INVSAMP.NVST. Which investments are acceptable if you expect a MARR of 9%?

Open the portfolio SASHELP.INVSAMP.NVST and compare the investments. Note that Internal Rate of Return computations assume regular periodicity of the cashflow. To compute the internal rates of return, follow these steps:

1. Select all five investments.
2. Select **Analyze** → **Internal Rate of Return...**.

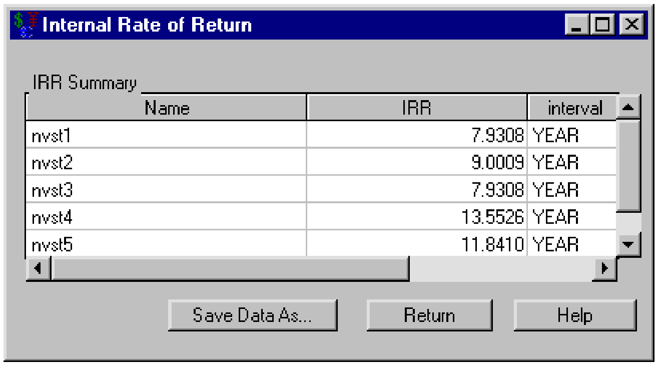

Figure 10.4. Computing an Internal Rate of Return

The results displayed in Figure 10.4 indicate that the internal rates of return for investments 2, 4, and 5 are greater than 9%. Hence, each of these is acceptable.

Performing a Benefit-Cost Ratio Analysis

Suppose a municipality has excess funds to invest. It is choosing between the same investments described in the previous example. Government agencies often compute benefit-cost ratios to decide which investment to pursue. Which is best in this case?

Open the portfolio SASHELP.INVSAMP.NVST and compare the investments.

To compute the benefit-cost ratios, follow these steps:

1. Select all five investments.
2. Select **Analyze** → **Benefit-Cost Ratio...**.
3. Enter 01JAN1996 for the **Date**.
4. Enter 9 for **Constant MARR**.
5. Click **Create Benefit-Cost Ratio Summary** to fill the **Benefit-Cost Ratio Summary** area.

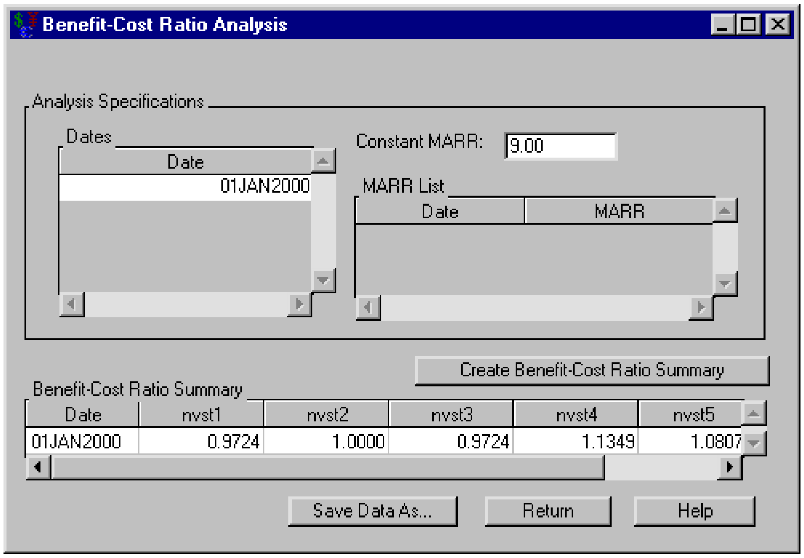

Figure 10.5. Performing a Benefit-Cost Ratio Analysis

The results displayed in Figure 10.5 indicate that investments 2, 4, and 5 have ratios greater than 1. Therefore, each is profitable with a MARR of 9%.

Computing a Uniform Periodic Equivalent

Suppose you need a warehouse for ten years. You have two options:

- Pay rent for ten years at $23,000 per year.
- Build a two-stage facility that you will maintain and which you intend to sell at the end of those ten years.

Datasets describing these scenarios are available in the portfolio SASHELP.INVSAMP.BUYRENT. Which option is more financially sound if you desire a 12% MARR?

Open the portfolio SASHELP.INVSAMP.BUYRENT and compare the options.

To perform the periodic equivalent, follow these steps:

1. Load the portfolio SASHELP.INVSAMP.BUYRENT.
2. Select both cashflows.
3. Select **Analyze** → **Periodic Equivalent...**. This opens the Uniform Periodic Equivalent dialog box.
4. Enter 01JAN1996 for the **Start Date**.
5. Enter 10 for the **Number of Periods**.
6. Select YEAR for the **Interval**.
7. Enter 12 for the **Constant MARR**.
8. Click **Create Time Value Summary**.

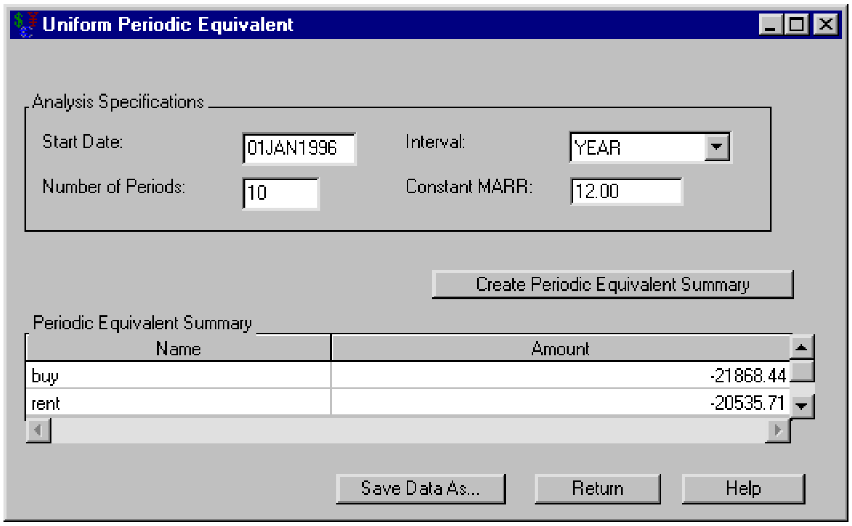

Figure 10.6. Computing a Uniform Periodic Equivalent

Figure 10.6 indicates that to rent costs about $1,300 less each year. Hence, renting is more financially sound. Notice the periodic equivalent for renting is not $23,000. This is because the the $23,000 per year does not account for the MARR.

Performing a Breakeven Analysis

In the previous example you computed the uniform periodic equivalent for a rent-buy scenario. Now let's perform a breakeven analysis to see how the MARR affects the time values.

To perform the breakeven analysis, follow these steps:

1. Select both options.
2. Select **Analyze** → **Breakeven Analysis...**.
3. Enter 01JAN1996 for the **Date**.
4. Enter 12.0 for **Value**.
5. Enter 4.0 for **(+/-)**.
6. Enter 0.5 for **Increment by**.
7. Click **Create Breakeven Analysis Summary** to fill the **Breakeven Analysis Summary** area as displayed in Figure 10.7.

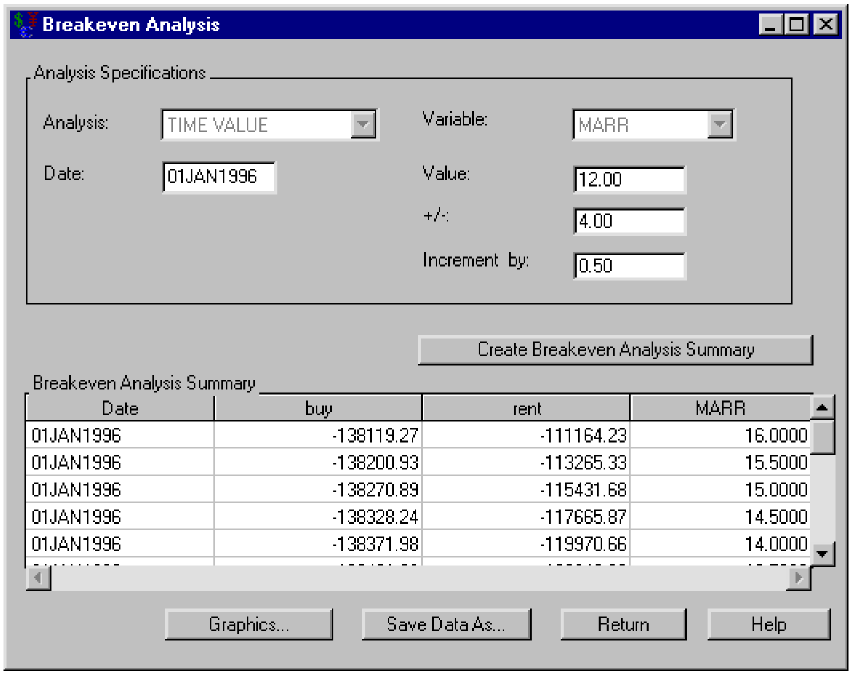

Figure 10.7. Performing a Breakeven Analysis

Click **Graphics...** to view a plot displaying the relationship between time value and MARR.

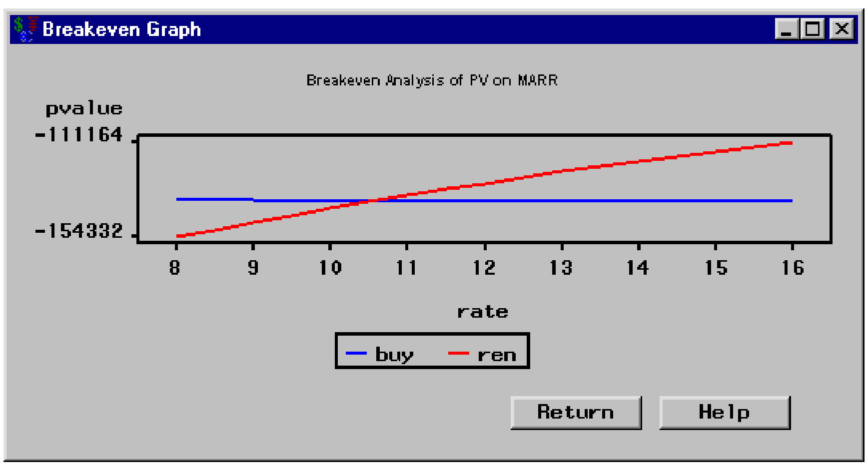

Figure 10.8. Viewing a Breakeven Graph

As shown in Figure 10.8 renting is better if you want a MARR of 12%. However, if your MARR should drop to 10.5%, buying would be better.

With a single investment, knowing where the graph has a time value of 0 tells the MARR when a venture switches from being profitable to a loss. With multiple investments, knowing where the graphs for the various investments cross each other tells at what MARR a particular investment becomes more profitable than another.

Dialog Box Guide

Time Value Analysis

Having selected a generic cashflow from the Investment Analysis dialog box, to perform an time value analysis, select **Analyze** → **Time Value...** from the Investment Analysis dialog box's menu bar. This opens the Time Value Analysis dialog box displayed in Figure 10.9.

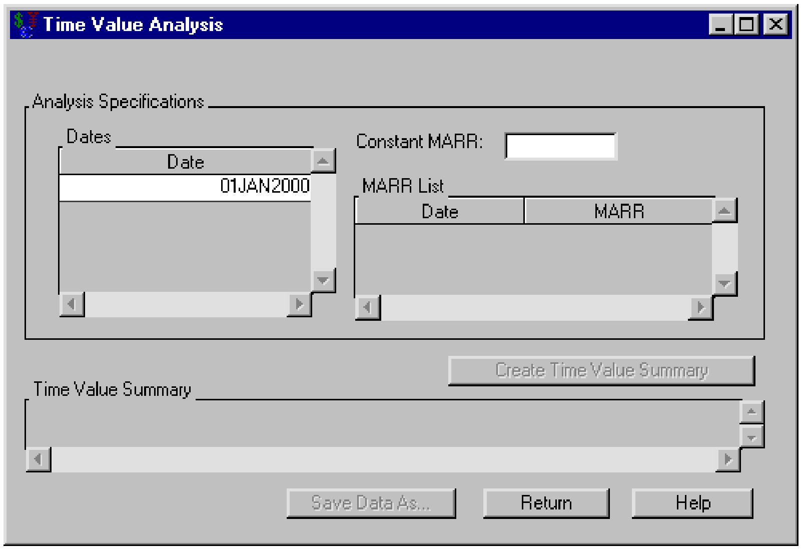

Figure 10.9. Time Value Analysis Dialog Box

The following items are displayed:

Analysis Specifications

> **Dates** holds the list of dates as of which to perform the time value analysis. Right-clicking within the **Dates** area reveals many helpful tools for managing date lists.
>
> **Constant MARR** holds the desired MARR for the time value analysis. This value is used if the **MARR List** area is empty.
>
> **MARR List** holds date-rate pairs that express your desired MARR as it changes over time. Each date refers to when that expected MARR begins. Right-clicking within the **MARR List** area reveals many helpful tools for managing date-rate pairs.

Create Time Value Summary becomes available when you adequately specify the analysis within the **Analysis Specifications** area. Clicking **Create Time Value Summary** then fills the **Time Value Summary** area.

Time Value Summary fills when you click **Create Time Value Summary**. The table contains a row for each date in the **Dates** area. The remainder of each row holds the time values at that date, one value for each investment selected.

Save Data As... becomes available when you fill the time value summary. Clicking it opens the Save Output Dataset dialog box where you can save the **Time Value Summary** (or portions thereof) as a SAS Dataset.

Return takes you back to the Investment Analysis dialog box.

Uniform Periodic Equivalent

Having selected a generic cashflow from the Investment Analysis dialog box, to perform a uniform periodic equivalent, select **Analyze** → **Periodic Equivalent...** from the Investment Analysis dialog box's menu bar. This opens the Uniform Periodic Equivalent dialog box displayed in Figure 10.10.

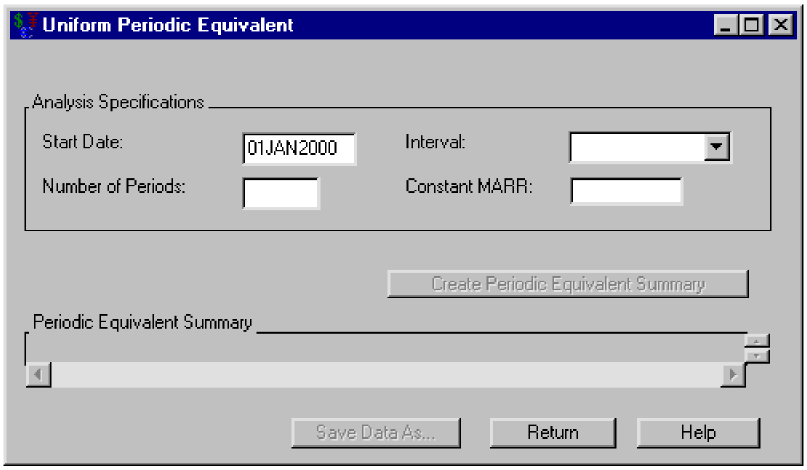

Figure 10.10. Uniform Periodic Equivalent Dialog Box

The following items are displayed:

Analysis Specifications

> **Start Date** holds the date the uniform periodic equivalents begin.
>
> **Number of Periods** holds the number of uniform periodic equivalents.
>
> **Interval** holds how often the uniform periodic equivalents occur.
>
> **Constant MARR** holds the Minimum Attractive Rate of Return.

Create Periodic Equivalent Summary becomes available when you adequately fill the **Analysis Specification** area. Clicking **Create Periodic Equivalent Summary** then fills the periodic equivalent summary.

Periodic Equivalent Summary fills with two columns when you click **Create Periodic Equivalent Summary**. The first column lists the investments selected. The second column lists the computed periodic equivalent amount.

Save Data As... becomes available when you generate the periodic equivalent summary. Clicking it opens the Save Output Dataset dialog box where you can save the Periodic Equivalent Summary (or portions thereof) as a SAS Dataset.

Return takes you back to the Investment Analysis dialog box.

Internal Rate of Return

Having selected a generic cashflow from the Investment Analysis dialog box, to perform an internal rate of return calculation, select **Analyze** → **Internal Rate of Return...** from the Investment Analysis dialog box's menu bar. This opens the Internal Rate of Return dialog box displayed in Figure 10.11.

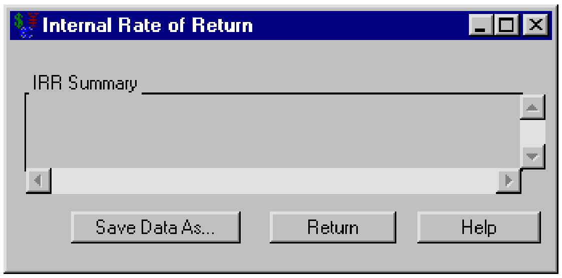

Figure 10.11. Internal Rate of Return Dialog Box

The following items are displayed:

IRR Summary contains a row for each deposit. Each row holds:

> **Name** holds the name of the investment.
>
> **IRR** holds the internal rate of return for that investment.
>
> **interval** holds the interest rate interval for that **IRR**.

Save Data As... Clicking it opens the Save Output Dataset dialog box where you can save the IRR summary (or portions thereof) as a SAS Dataset.

Return takes you back to the Investment Analysis dialog box.

Benefit-Cost Ratio Analysis

Having selected a generic cashflow from the Investment Analysis dialog box, to compute a benefit-cost ratio, select **Analyze** → **Benefit-Cost Ratio...** from the Investment Analysis dialog box's menu bar. This opens the Benefit-Cost Ratio Analysis dialog box displayed in Figure 10.12.

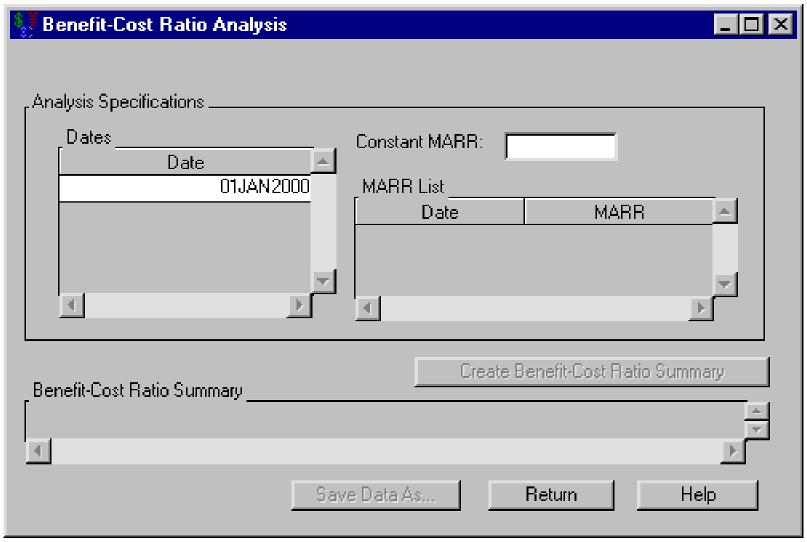

Figure 10.12. Benefit-Cost Ratio Analysis Dialog Box

The following items are displayed:

Analysis Specifications

> **Dates** holds the dates as of which to compute the Benefit-Cost ratios.
>
> **Constant MARR** holds the desired MARR. This value is used if the **MARR List** area is empty.
>
> **MARR List** holds date-rate pairs that express your desired MARR as it changes over time. Each date refers to when that expected MARR begins. Right-clicking within the **MARR List** area reveals many helpful tools for managing date-rate pairs.

Create Benefit-Cost Ratio Summary becomes available when you adequately specify the analysis. Clicking **Create Benefit-Cost Ratio Summary** fills the benefit-cost ratio summary.

Benefit-Cost Ratio Summary fills when you click **Exchange the Rates**. The area contains a row for each date in the **Dates** area. The remainder of each row holds the benefit-cost ratios at that date, one value for each investment selected.

Save Data As... becomes available when you generate the benefit-cost ratio summary. Clicking it opens the Save Output Dataset dialog box where you can save the Benefit-Cost Summary (or portions thereof) as a SAS Dataset.

Return takes you back to the Investment Analysis dialog box.

Breakeven Analysis

Having selected a generic cashflow from the Investment Analysis dialog box, to perform a breakeven analysis, select **Analyze** → **Breakeven Analysis...** from the Investment Analysis dialog box's menu bar. This opens the Breakeven Analysis dialog box displayed in Figure 10.13.

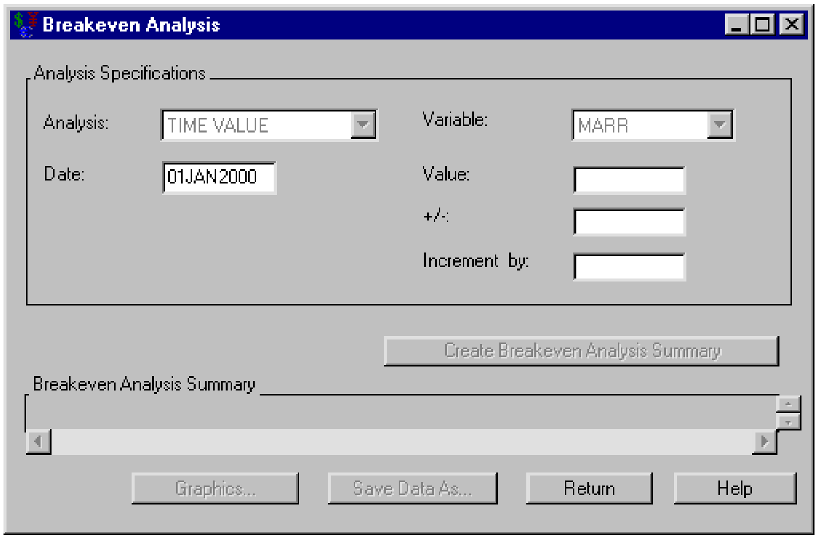

Figure 10.13. Breakeven Analysis Dialog Box

The following items are displayed:

Analysis Specification

Analysis holds the analysis type. Only Time Value is currently available.

Date holds the date for which you perform this analysis.

Variable holds the variable upon which the breakeven analysis will vary. Only MARR is currently available.

Value holds the desired rate upon which to center the analysis.

+/- holds the maximum deviation from the **Value** to consider.

Increment by holds the increment by which the analysis is calculated.

Create Breakeven Analysis Summary becomes available when you adequately specify the analysis. Clicking **Create Breakeven Analysis Summary** then fills the **Breakeven Analysis Summary** area.

Breakeven Analysis Summary fills when you click **Create Breakeven Analysis Summary**. The schedule contains a row for each MARR and date.

Graphics... becomes available when you fill the **Breakeven Analysis Summary** area. Clicking it opens the Breakeven Graph graph representing the time value versus MARR.

Save Data As... becomes available when you generate the breakeven analysis summary. Clicking it opens the Save Output Dataset dialog box where you can save the Breakeven Analysis Summary (or portions thereof) as a SAS Dataset.

Return takes you back to the Investment Analysis dialog box.

Breakeven Graph

Suppose you perform a breakeven analysis in the Breakeven Analysis dialog box. Once you create the breakeven analysis summary, you can click the **Graphics...** button to open the Breakeven Graph dialog box displayed in Figure 10.14.

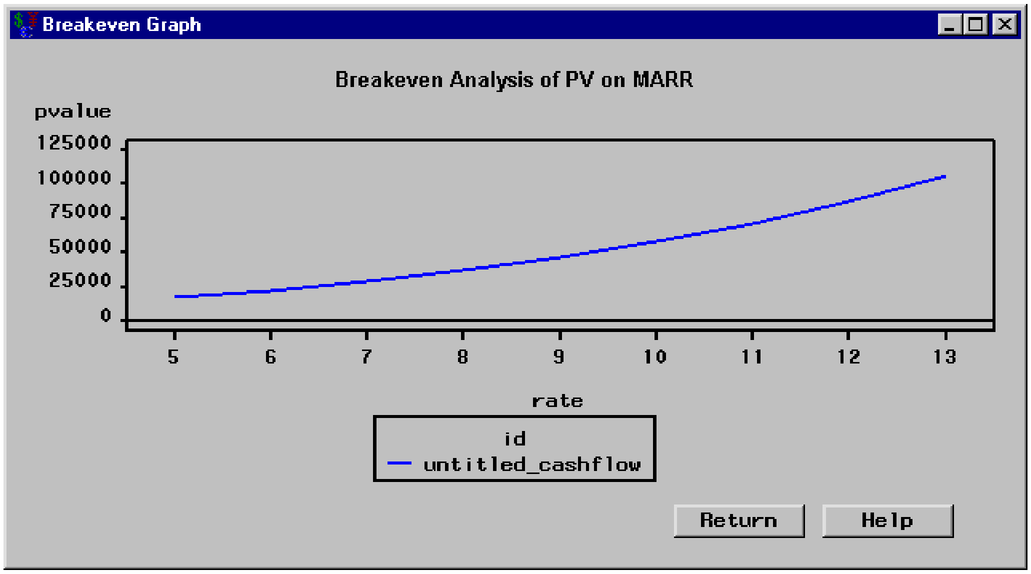

Figure 10.14. Breakeven Graph Dialog Box

The following item is displayed:

Return takes you back to the Breakeven Analysis dialog box.

Chapter 11
Details

Chapter Table of Contents

Chapter 11
Details

Investments and Datasets

Investment Analysis provides tools to assist you in moving data between SAS datasets and lists you can use within Investment Analysis.

Saving Output to SAS Datasets

Many investment specifications have a button that reads **Save Data As...**. Clicking that button opens the Save Output Dataset dialog box (see Figure 11.1). This dialog box enables you to save all or part of the area generated by the specification.

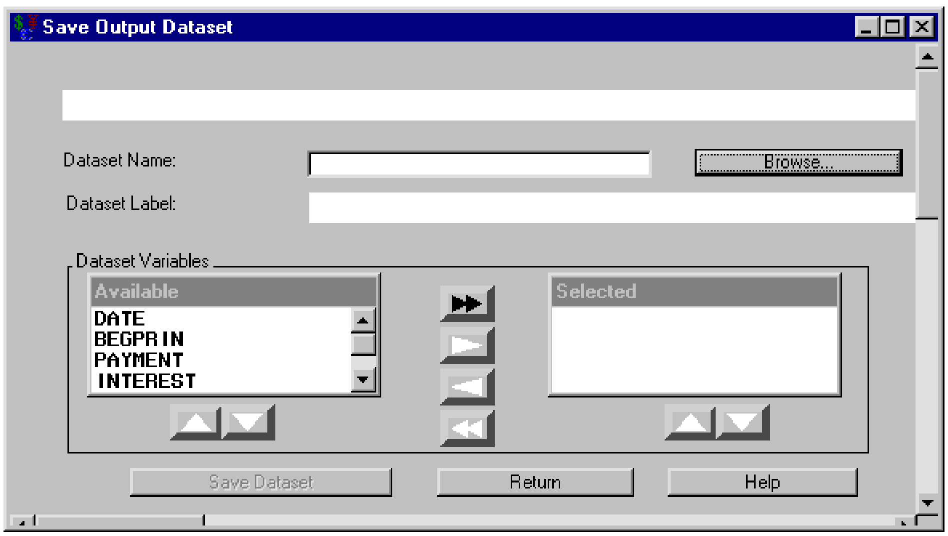

Figure 11.1. Saving to a Dataset

The following items are displayed:

Dataset Name holds the SAS dataset name to which you wish to save.

Browse... opens the standard SAS **Open** dialog box, which enables you to select an existing dataset to overwrite.

Dataset Label holds the dataset's label.

Dataset Variables organizes variables. The varables listed in the **Selected** area will be included in the dataset.

- You can select variables one at a time, by clicking the single right-arrow after each selection to move it to the **Selected** area.
- If the desired dataset has many variables you wish to save, it may be simpler to follow these steps:
  1. Click the double right arrow to select all available variables.
  2. Remove any unwanted variable by selecting it from the **Selected** area and clicking the single left arrow.
- The double left arrow removes all selected variables from the proposed dataset.
- The up and down arrows below the **Available** and **Selected** boxes enable you to scroll up and down the list of variables in their respective boxes.

Save Dataset attempts to save the dataset. If the Dataset name exists, you are asked if you want to replace the dataset, append to the existing Dataset, or cancel the current save attempt. You then return to this dialog box ready to create another dataset to save.

Return takes you back to the specification dialog box.

Loading a SAS Dataset into a List

Right-click in the area you wish to load the list and release on **Load...**. This opens the Load Dataset dialog box (see Figure 11.2).

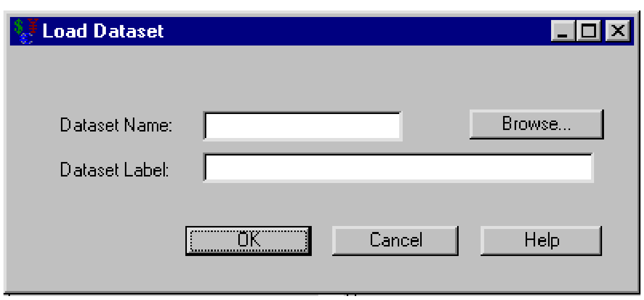

Figure 11.2. Load Dataset Dialog Box

The following items are displayed:

Dataset Name holds the name of the SAS dataset you wish to load.

Browse... opens the standard SAS **Open** dialog box, which aids in finding a dataset to load. If there is a **Date** variable in the dataset, Investment Analysis loads it into the list. If there is no **Date** variable, it loads the first available time-formatted variable. If an amount or rate variable is needed, Investment Analysis searches the dataset for a **Amount** or **Rate** variable to use. Otherwise it takes the first numeric variable that is not used by the **Date** variable.

Dataset Label holds a dataset label.

OK attempts to load the dataset specified in **Dataset Name**. If the specified dataset exists, clicking **OK** returns you to the calling dialog box with the selected dataset filling the list. If the specified dataset does not exist and you click **OK**, you receive an error message and no dataset is loaded.

Cancel returns you to the calling dialog box without loading a dataset. To load values from a dataset into a list, follow these steps:

Saving Data from a List to a SAS Dataset

Right-click in the area you wish to hold the list, and release on **Save...**. This opens the Save Dataset dialog box.

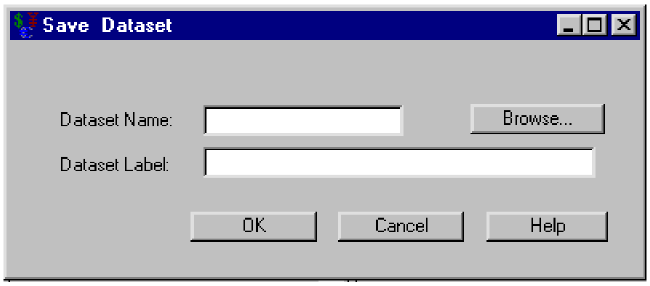

Figure 11.3. Save Dataset Dialog Box

The following items are displayed:

Dataset Name holds the SAS dataset name to which you wish to save.

Browse... opens the standard SAS **Save As** dialog box, which enables you to find an existing dataset to overwrite.

Dataset Label holds a user-defined description to be saved as the label of the dataset.

OK saves the current data to the dataset specified in **Data set Name**. If the specified dataset does not already exist, clicking **OK** saves the dataset and returns you to the calling dialog box. If the specified dataset does already exist, clicking **OK** warns you and enables you to replace the old dataset with the new dataset or cancel the save attempt.

Cancel aborts the save process. Clicking **Cancel** returns you to the calling dialog box without attempting to save.

Right Mouse Button Options

A pop-up menu often appears when you right-click within table editors. The menus offer tools to aid in the management of the table's entries. Most table editors provide the following options.

Figure 11.4. Right-Clicking Options

Add creates a blank row.

Delete removes any currently selected row.

Copy duplicates the currently selected row.

Sort arranges the rows in chronological order according to the date variable.

Clear empties the table of all rows.

Save... opens the Save Dataset dialog box where you can save the all rows to a SAS Dataset for later use.

Load... opens the Load Dataset dialog box where you select a SAS Dataset to fill the rows.

If you wish to perform one of these actions on a collection of rows, you must select a collection of rows before right-clicking. To select an adjacent list of rows, do the following: click the first pair, hold down SHIFT, and click the final pair. Once the list of rows is selected, you may release the SHIFT key.

Depreciation Methods

Suppose an asset's price is $20,000 and it has a salvage value of $5,000 in five years. The following sections describe various methods to quantify the depreciation.

Straight Line (SL)

This method assumes a constant depreciation value per year.

Assuming the price of a depreciating asset is P and its salvage value after N years is S,

$$\text{Annual Depreciation} = \frac{P-S}{N}$$

For our example, the annual depreciation would be

$$\frac{\$20,000 - \$5,000}{5} = \$3,000$$

Sum-of-years Digits

An asset often loses more of its value early in its lifetime. A method that exhibits this dynamic is desirable.

Assume an asset depreciates from price P to salvage value S in N years. First compute the value: sum-of-years = $1 + 2 + \cdots + N$. The depreciation for the years after the asset's purchase is:

Table 11.1. Sum-of-years General Example

year number	annual depreciation
first	$\frac{N}{\mathrm{sum-of-years}}(P - S)$
second	$\frac{N-1}{\mathrm{sum-of-years}}(P - S)$
third	$\frac{N-2}{\mathrm{sum-of-years}}(P - S)$
$\vdots$	$\vdots$
final	$\frac{1}{\mathrm{sum-of-years}}(P - S)$

For the ith year of the asset's use this equation generalizes to

$$\text{Annual Depreciation} = \frac{N+1-i}{\mathrm{sum-of-years}}(P - S)$$

For our example, $N = 5$ and the sum of years is $1 + 2 + 3 + 4 + 5 = 15$. The depreciation during the first year is

$$(\$20,000 - \$5,000)\frac{5}{15} = \$5,000$$

Table 11.2 describes how Declining Balance would depreciate the asset.

Table 11.2. Sum-of-years Example

Year	Depreciation	Year-end Value
1	$(\$20,000 - \$5,000)\frac{5}{15} = \$5,000$	$\$15,000.00$
2	$(\$20,000 - \$5,000)\frac{4}{15} = \$4,000$	$\$11,000.00$
3	$(\$20,000 - \$5,000)\frac{3}{15} = \$3,000$	$\$8,000.00$
4	$(\$20,000 - \$5,000)\frac{2}{15} = \$2,000$	$\$6,000.00$
5	$(\$20,000 - \$5,000)\frac{1}{15} = \$1,000$	$\$5,000.00$

And as expected, the value after N years is S.

$$\begin{aligned}\text{Value after 5 years} &= P \text{ - (5 years' depreciation)}\\ &= P \text{ - } \left(\tfrac{5}{10}(P-S)+\tfrac{4}{10}(P-S)+\tfrac{3}{10}(P-S)+\right.\\ &\qquad \left.\tfrac{2}{10}(P-S)+\tfrac{1}{10}(P-S)\right)\\ &= P-(P-S)\\ &= S\end{aligned}$$

Declining Balance (DB)

Recall that the Straight Line method assumes a constant depreciation value. Conversely, the Declining Balance method assumes a constant depreciation rate per year. And like the Sum-of-years method, more depreciation tends to occur earlier in the asset's life.

Assume the price of a depreciating asset is P and its salvage value after N years is S. You could assume the asset depreciates by a factor of $\frac{1}{N}$ (or a rate o f $\frac{100}{N}\%$). This method is known as Single Declining Balance. In an equation this looks like:

$$\text{Annual Depreciation} = \tfrac{1}{N} \text{ Previous year's value}$$

So for our example, the depreciation during the first year is

$$\frac{\$20,000}{5} = \$4,000$$

Table 11.3 describes how Declining Balance would depreciate the asset.

Table 11.3. Declining Balance Example

Year	Depreciation	Year-end Value
1	$\frac{\$20{,}000.00}{5} = \$4,000.00$	$\$16,000.00$
2	$\frac{\$16{,}000.00}{5} = \$3,200.00$	$\$12,800.00$
3	$\frac{\$12{,}800.00}{5} = \$2,560.00$	$\$10,240.00$
4	$\frac{\$10{,}240.00}{5} = \$2,560.00$	$\$8,192.00$
5	$\frac{\$12{,}800.00}{5} = \$2,560.00$	$\$6,553.60$

DB Factor

You could also accelerate the depreciation by increasing the factor (and hence the rate) at which depreciation occurs. Other commonly accepted depreciation rates are $\frac{200}{N}\%$ (called Double Declining Balance as the depreciation factor becomes $\frac{2}{N}$) and $\frac{150}{N}\%$. Investment Analysis enables you to choose between these three types for Declining Balance: 2 (with $\frac{200}{N}\%$ depreciation), 1.5 (with $\frac{150}{N}\%$), and 1 (with $\frac{100}{N}\%$).

Declining Balance and the Salvage Value

The Declining Balance method assumes that depreciation is faster earlier in an asset's life; this is what you wanted. But notice the final value is greater than the salvage value. Even if the salvage value were greater than \$6,553.60, the final year-end value would not change. The salvage value never enters the calculation, so there is no way for the salvage value to force the depreciation to assume its value. Newnan and Lavelle (1998) describe two ways to adapt the Declining Balance method to assume the salvage value at the final time. One way is as follows:

Suppose you call the depreciated value after i years $V(i)$. This sets $V(0) = P$ and $V(N) = S$.

- If $V(N) > S$ according to the usual calculation for $V(N)$, redefine $V(N)$ to equal S.
- If $V(i) < S$ according to the usual calculation for $V(i)$ for some i (and hence for all subsequent $V(i)$ values), you can redefine all such $V(i)$ to equal S.

This alteration to Declining Balance forces the depreciated value of the asset after N years to be S and keeps $V(i)$ no less than S.

Conversion to SL

The second (and preferred) way to force Declining Balance to assume the salvage value is by Conversion to Straight Line. If $V(N) > S$, the first way redefines $V(N)$ to equal S; you can think of this as converting to the Straight Line method for the last timestep.

If the $V(N)$ value supplied by DB is appreciably larger than S, then the depreciation in the final year would be unrealistically large. An alternate way is to compute the DB and SL step at each timestep and take whichever step gives a larger depreciation (unless DB drops below the salvage value).

Once SL assumes a larger depreciation, it continues to be larger over the life of the asset. This forces the value at the final time to equal the salvage value as SL forces this. As an algorithm, this looks like

```
V(0) = P;
for i=1 to N
   if DB step > SL step from (i,V(i))
      take a DB step to make V(i);
   else
      break;
for j = i to N
   take a SL step to make V(j);
```

The MACRS discussed in Depreciation Table... is actually a variation on the Declining Balance with conversion to Straight Line method.

Comparison of Depreciation Methods

Table 11.4 displays the depreciation for four depreciation methods.

Table 11.4. Comparison of Depreciation Methods

year	sbvalue	deprectn	ebvalue
1999	20000.00	3000.00	17000.00
2000	17000.00	3000.00	14000.00
2001	14000.00	3000.00	11000.00
2002	11000.00	3000.00	8000.00
2003	8000.00	3000.00	5000.00

Straight Line

year	sbvalue	deprectn	ebvalue
1999	20000.00	5000.00	15000.00
2000	15000.00	4000.00	11000.00
2001	11000.00	3000.00	8000.00
2002	8000.00	2000.00	6000.00
2003	6000.00	1000.00	5000.00

Sum-of-years Digits

year	sbvalue	deprectn	ebvalue
1999	20000.00	4000.00	16000.00
2000	16000.00	6400.00	9600.00
2001	9600.00	3840.00	5760.00
2002	5760.00	2304.00	3456.00
2003	3456.00	2304.00	1152.00
2004	1152.00	1152.00	0.00

Depreciation Table

year	sbvalue	deprectn	ebvalue
1999	20000.00	8000.00	12000.00
2000	12000.00	4800.00	7200.00
2001	7200.00	2200.00	5000.00
2002	5000.00	0.00	5000.00
2003	5000.00	0.00	5000.00

Declining Balance

- Under Depreciation Table, realize a 5-year class MACRS Depreciation actually lasts 6 years.
- The Declining Balance method is Double Declining Balance with conversion to Straight Line.

For further reference, consider Newnan and Lavelle (1998). They offer explanations and graphs for the individual depreciation methods. They also analyze the differences between the various methods.

Rate Information

The Tools Menu

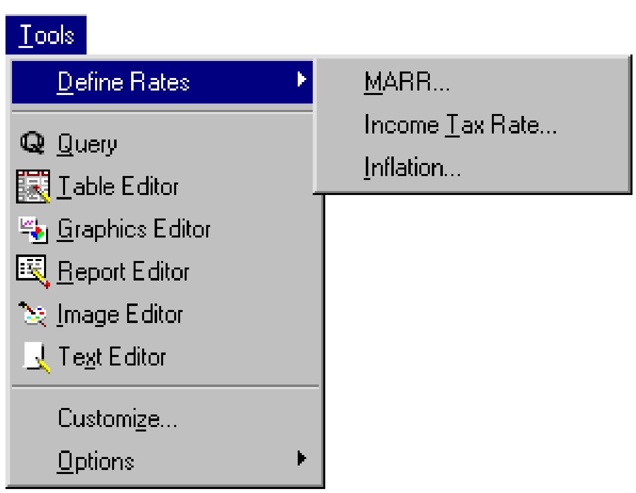

Figure 11.5. The Tools Menu

The **Tools** → **Define Rates** menu offers the following options:

MARR... opens the Minumum Attractive Rate of Return (MARR) dialog box.

Income Tax Rate... opens the Income Tax Specification dialog box.

Inflation... opens the Inflation Specification dialog box.

Dialog Box Guide

Minimum Attractive Rate of Return (MARR)

Selecting **Tools** → **Define Rates** → **MARR** from the Investment Analysis dialog box menu bar opens the MARR dialog box displayed in Figure 11.6.

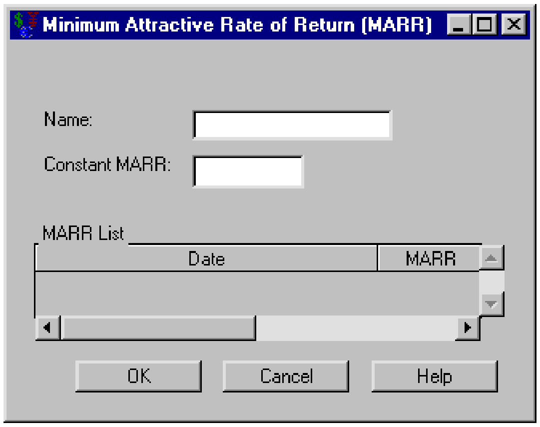

Figure 11.6. MARR Dialog Box

Name holds the name you assign to the MARR specification. This must be a valid SAS name.

Constant MARR holds the numeric value you choose to be the constant MARR. This value is used if the **MARR List** table editor is empty.

MARR List holds date-MARR pairs where the date refers to when the particular MARR value begins. Each date is a SAS date.

OK returns you to the Investment Analysis dialog box. Pressing it causes the MARR specification above to be assumed when you do not specify MARR rates in a dialog box that needs MARR rates.

Cancel returns you to the Investment Analysis dialog box, discarding any work done in the MARR dialog box.

Income Tax Specification

Selecting **Tools** → **Define Rates** → **Income Tax Rate** from the Investment Analysis dialog box menu bar opens the Income Tax Specification dialog box displayed in Figure 11.7.

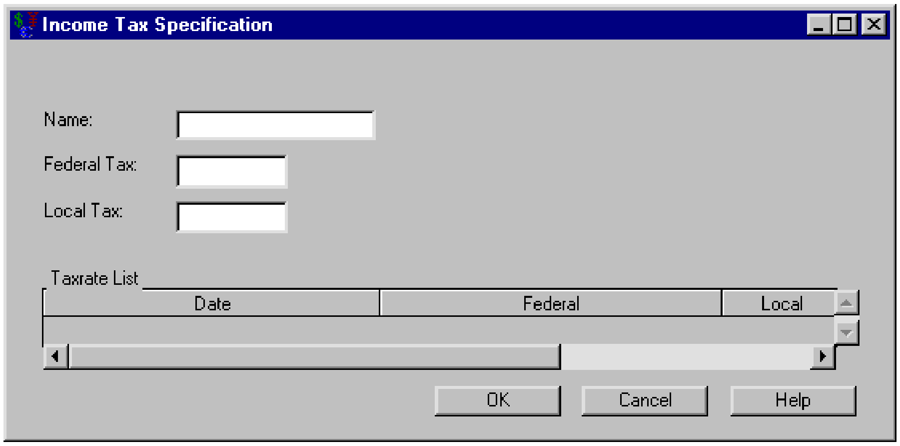

Figure 11.7. Income Tax Specification Dialog Box

Name holds the name you assign to the Income Tax specification. This must be a valid SAS name.

Federal Tax holds the numeric value you desire to be the constant Federal Tax.

Local Tax holds the numeric value you desire to be the constant Local Tax.

Taxrate List holds date-Income Tax triples where the date refers to when the particular Income Tax value begins. Each date is a SAS date, and the value is a percentage between 0% and 100%.

OK returns you to the Investment Analysis dialog box. Clicking it causes the income tax specification above to be the default income tax rates when using the After Tax Cashflow Calculation dialog box.

Cancel returns you to the Investment Analysis dialog box, discarding any editions since this dialog box was opened.

Inflation Specification

Selecting **Tools** → **Define Rates** → **Inflation** from the Investment Analysis dialog box menu bar opens the Inflation Specification dialog box displayed in Figure 11.8.

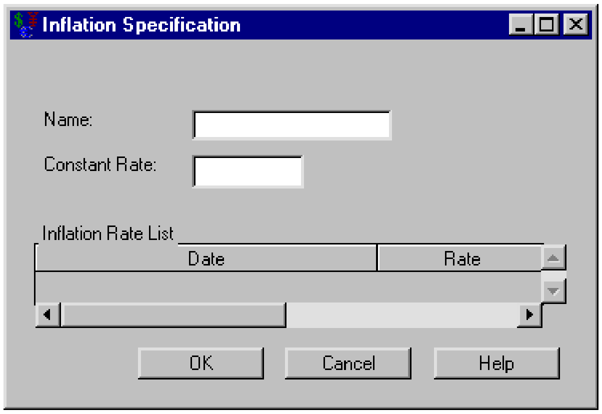

Figure 11.8. Inflation Specification Dialog Box

Name holds the name you assign to the Inflation specification. This must be a valid SAS name.

Constant Rate holds the numeric value you desire to be the constant inflation rate. This value is used if the **Inflation Rate List** table editor is empty.

Inflation Rate List holds date-rate pairs where the date refers to when the particular inflation rate begins. Each date is a SAS date and the rate is a percentage between 0% and 120%.

OK returns you to the Investment Analysis dialog box. Pressing it causes the inflation specification above to be assumed when you use the Constant Dollar Calculation dialog box and do not specify inflation rates.

Cancel returns you to the Investment Analysis dialog box, discarding any editions since this dialog box was opened.

Reference

Newnan, Donald G. and Lavelle, Jerome P. (1998), *Engineering Economic Analysis,* Austin, Texas: Engineering Press.

Subject Index

Syntax Index

Your Turn

If you have comments or suggestions about *SAS/ETS® Software: Changes and Enhancements, Release 8.1*, please send them to us on a photocopy of this page or send us electronic mail.

For comments about this book, please return the photocopy to

SAS Institute
Publications Division
SAS Campus Drive
Cary, NC 27513
email: yourturn@sas.com

For suggestions about the software, please return the photocopy to

SAS Institute
Technical Support Division
SAS Campus Drive
Cary, NC 27513
email: suggest@sas.com

*Welcome * Bienvenue * Willkommen * Yohkoso * Bienvenido*

SAS® Institute Publishing Is Easy to Reach

Visit our Web page located at www.sas.com/pubs

You will find product and service details, including

- **sample chapters**
- **tables of contents**
- **author biographies**
- **book reviews**

Learn about

- **regional user-group conferences**
- **trade-show sites and dates**
- **authoring opportunities**
- **custom textbooks**

Explore all the services that SAS Institute Publishing has to offer!

Your Listserv Subscription Automatically Brings the News to You

Do you want to be among the first to learn about the latest books and services available from SAS Institute Publishing? Subscribe to our listserv **newdocnews-l** and, once each month, you will automatically receive a description of the newest books and which environments or operating systems and SAS release(s) that each book addresses.

To subscribe,

1. Send an e-mail message to **listserv@vm.sas.com.**
2. Leave the "Subject" line blank.
3. Use the following text for your message:

   subscribe NEWDOCNEWS-L *your-first-name your-last-name*

   For example: subscribe NEWDOCNEWS-L John Doe

Create Customized Textbooks Quickly, Easily, and Affordably

SelecText® offers instructors at U.S. colleges and universities a way to create custom textbooks for courses that teach students how to use SAS software.

For more information, see our Web page at **www.sas.com/selectext**, or contact our SelecText coordinators by sending e-mail to **selectext@sas.com**.

You're Invited to Publish with SAS Institute's User Publishing Program

If you enjoy writing about SAS software and how to use it, the User Publishing Program at SAS Institute offers a variety of publishing options. We are actively recruiting authors to publish books, articles, and sample code. Do you find the idea of writing a book or an article by yourself a little intimidating? Consider writing with a co-author. Keep in mind that you will receive complete editorial and publishing support, access to our users, technical advice and assistance, and competitive royalties. Please contact us for an author packet. E-mail us at **sasbbu@sas.com** or call 919-677-8000, then press 1-6479. See the SAS Institute Publishing Web page at **www.sas.com/pubs** for complete information.

See *Observations*®, Our Online Technical Journal

Feature articles from *Observations®: The Technical Journal for SAS® Software Users* are now available online at **www.sas.com/obs**. Take a look at what your fellow SAS software users and SAS Institute experts have to tell you. You may decide that you, too, have information to share. If you are interested in writing for *Observations*, send e-mail to **sasbbu@sas.com** or call 919-677-8000, then press 1-6479.

Book Discount Offered at SAS Public Training Courses!

When you attend one of our SAS Public Training Courses at any of our regional Training Centers in the U.S., you will receive a 15% discount on book orders that you place during the course. Take advantage of this offer at the next course you attend!

SAS Institute
SAS Campus Drive
Cary, NC 27513-2414
Fax 919-677-4444

E-mail: sasbook@sas.com
Web page: www.sas.com/pubs
To order books, call Fulfillment Services at 800-727-3228*
For other SAS Institute business, call 919-677-8000*

* **Note:** Customers outside the U.S. should contact their local SAS office.