



SAS Publishing

TECHNICAL REPORT

SAS/ETS® Software: Changes and Enhancements, Release 8.2



The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/ETS® Software: Changes and Enhancements, Release 8.2*, Cary, NC: SAS Institute Inc., 2001

SAS/ETS® Software: Changes and Enhancements, Release 8.2
Copyright © 2001 by SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-849-2

All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of this software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

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

1st printing, January 2001

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

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

Table of Contents

Part 1. General Information	1
Chapter 1. The SASECRSP Interface Engine	3
Chapter 2. The SASEFAME Interface Engine	29
 Part 2. Procedure Reference	 51
Chapter 3. The ARIMA Procedure	53
Chapter 4. The MODEL Procedure	67
Chapter 5. The VARMAX Procedure	91
Chapter 6. The X12 Procedure	97
 Subject Index	 109
 Syntax Index	 111

Part 1

General Information

Table of Contents

Chapter 1. The SASECRSP Interface Engine	3
Chapter 2. The SASEFAME Interface Engine	29

Part 1. General Information

Chapter 1

The SASECRSP Interface Engine

Chapter Table of Contents

OVERVIEW	5
GETTING STARTED	5
Structure of a SAS Data Set Containing Time Series Data	5
Reading CRSP Data Files	5
Using the SAS DATA Step	6
Using SAS Procedures	6
Using CRSP Date Formats, Informats, and Functions	6
SYNTAX	6
The LIBNAME <i>libref</i> SASECRSP Statement	7
DETAILS	9
The SAS Output Data Set	9
Available Data Sets	9
Understanding CRSP Date Formats, Informats, and Functions	14
EXAMPLES	18
Example 1.1 Extracting a List of PERMNOs Using a RANGE	18
Example 1.2 Extracting All PERMNOs Using a RANGE	20
Example 1.3 Extracting One PERMNO Using No RANGE, Wide Open	22
Example 1.4 Extracting Selected PERMNOs in Range Using INSET= Option	24
Example 1.5 Converting Dates Using the CRSP Date Functions	26
REFERENCES	28

Chapter 1

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. Since CRSP and SAS use three different date representations, you can use the engine-provided CRSP date formats, informats, and functions for flexible seamless programming.

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 CALDT.

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. The CRSP environment variable CRSPDB_SASCAL must be defined to allow the SASECRSP engine access to the CRSPAccess database calendars.

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.

Using CRSP Date Formats, Informats, and Functions

CRSP has historically used two different methods to represent dates, and SAS has used a third. The SASECRSP engine provides 22 functions, 15 informats, and 10 formats to enable you to easily translate the dates from one internal representation to another. See the section “Understanding CRSP Date Formats, Informats, and Functions” on page 14 for details.

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 is daily and 20 monthly. This is a required option.	LIBNAME <i>libref</i> SASECRSP	SETID=
specify the PERMNO of a security to be kept in the SAS data set.	LIBNAME <i>libref</i> SASECRSP	PERMNO=
specify the range of data in format YYYYMMDD using the <i>crsp_begdt</i> and <i>crsp_enddt</i> .	LIBNAME <i>libref</i> SASECRSP	RANGE=
use a SAS data set named <i>setname</i> as input for issues.	LIBNAME <i>libref</i> SASECRSP	INSET=

The LIBNAME *libref* SASECRSP Statement

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

The CRSP environment variable CRSPDB_SASCAL must be defined to allow the SASECRSP engine access to the CRSPAccess database calendars. Often your CRSPDB_SASCAL will be pointing to the same path as your *physical name*. You must use a fully qualified pathname. Your *physical name* must end with either a forward slash if you are on a UNIX system, or a backward slash if you are on WINNT.

For a more complete description of SASECRSP set names, key fields, and date fields, see Table 1.1.

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. The SETID limits the frequency selection of time series that are included in the SAS data set. SETID is a required option for the SASECRSP engine. Depending on your host system, both should end in either a forward slash (UNIX) or a backward slash (WINNT). For more details about the CRSPAccess *crsp_setidnumber*, refer to “Using Multiple Products” in the *CRSPAccess Database Format Installation Guide*. 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, you could use the following statements:

```
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 ending date of the range in

Part 1. General Information

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 monthly data for Microsoft Corporation and for International Business Machine Corporation for the first quarter of 2000, you could 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 CRSPAccess supported 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. For more details about the CRSPAccess keyfields and keyfieldtypes, refer to “Stock and Indices Data Structures” in the *CRSP Data Description Guide*.

The following example might be used to extract the data for a portfolio of four companies to access monthly data where the range of dates is left open:

```
data testin1;
  permno = 10107; output;
  permno = 12490; output;
  permno = 14322; output;
  permno = 25788; output;
run;

LIBNAME mstk sasecrsp 'physical-name'
SETID=20
INSET='testin1';

data a;
  set mstk.prc;
run;

proc print data=a;
run;
```

Suppose you want to restrict the dates for each PERMNO specifically. The example below shows how to specify the INSET= option with the 'setname[,keyfieldname][[,keyfieldtype][[,date1field][[,date2field]]' parameters:

```
data testin2;
    permno = 10107; date1 = 19990101; date2 = 19991231; output;
    permno = 12490; date1 = 19970101; date2 = 19971231; output;
    permno = 14322; date1 = 19990901; date2 = 19991231; output;
    permno = 25788; date1 = 19950101; date2 = 19950331; output;
run;

LIBNAME mstk2 sasecrsp 'physical-name'
    SETID=20
    INSET='testin2,PERMNO,PERMNO,DATE1,DATE2';

data b;
    set mstk2.prc;
run;

proc print data=b;
run;
```

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.

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 1.1 shows the available data sets provided by the SASECRSP interface. Missing values are represented as '.' in the SAS data set. You can see the available data sets in the SAS LIBNAME window of the SAS Display Manager by selecting the SASECRSP libref in the LIBNAME window that you have previously used in your libname statement. You can use PROC PRINT and PROC CONTENTS to print your output data set and its contents. You can view your SAS output observations by double clicking on the desired output data set libref in the libname window of the Display Manager. You can use PROC SQL along with the SASECRSP engine to create a view of your SAS data set.

Part 1. General Information

Table 1.1. Data Sets Available

Dataset	Fields	Label	Type
STKHEAD Header Identification and Summary Data	PERMNO	PERMNO	Numeric
	PERMCO	PERMCO	Numeric
	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 Name History Array	PERMNO	PERMNO	Numeric
	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 Distribution Event Array	PERMNO	PERMNO	Numeric
	DISTCD	Distribution Code	Numeric
	DIVAMT	Dividend Cash Amount	Numeric
	FACPR	Factor to Adjust Price	Numeric
	FACSHR	Factor to Adjust Share	Numeric
	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 Observation Date	Numeric
	SHRENDT	Shares Outstanding Observation End Date	Numeric
	SHRFLG	Shares Outstanding Observation Flag	Numeric

Table 1.1. (continued)

Dataset	Fields	Label	Type
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
PRC Price or Bid/Ask Average Time Series	NSDINX	Nasd Index Code	Numeric
	PERMNO	PERMNO	Numeric
	CALDT	Calendar Trading Date	Numeric
RET Returns Time Series	PRC	Price or Bid/Ask Aver	Numeric
	PERMNO	PERMNO	Numeric
	CALDT	Calendar Trading Date	Numeric
ASKHI Ask or High Time Series	RET	Returns	Numeric
	PERMNO	PERMNO	Numeric
	CALDT	Calendar Trading Date	Numeric
BIDLO Bid or Low Time Series	ASKHI	Ask or High	Numeric
	PERMNO	PERMNO	Numeric
	CALDT	Calendar Trading Date	Numeric
BID Bid Time Series	BIDLO	Bid or Low	Numeric
	PERMNO	PERMNO	Numeric
	CALDT	Calendar Trading Date	Numeric
ASK Ask Time Series	BID	Bid	Numeric
	PERMNO	PERMNO	Numeric
	CALDT	Calendar Trading Date	Numeric
RETX Returns without Dividends Time Series	ASK	Ask	Numeric
	PERMNO	PERMNO	Numeric
	CALDT	Calendar Trading Date	Numeric
SPREAD Spread Between Bid and Ask Time Series	RETX	Returns w/o Dividends	Numeric
	PERMNO	PERMNO	Numeric
	CALDT	Calendar Trading Date	Numeric
	SPREAD	Spread Between Bid Ask	Numeric

Part 1. General Information

Table 1.1. (continued)

Dataset	Fields	Label	Type
ALTPRC	PERMNO	PERMNO	Numeric
Price Alternate	CALDT	Calendar Trading Date	Numeric
Time Series	ALTPRC	Price Alternate	Numeric
VOL	PERMNO	PERMNO	Numeric
Volume Time Series	CALDT	Calendar Trading Date	Numeric
	VOL	Volume	Numeric
NUMTRD	PERMNO	PERMNO	Numeric
Number of Trades	CALDT	Calendar Trading Date	Numeric
Time Series	NUMTRD	Number of Trades	Numeric
ALTPRCDT	PERMNO	PERMNO	Numeric
Alternate Price Date	CALDT	Calendar Trading Date	Numeric
Time Series	ALTPRCDT	Alternate Price Date	Numeric
PORT1	PERMNO	PERMNO	Numeric
Portfolio Data for	CALDT	Calendar Trading Date	Numeric
Portfolio Type 1	PORT1	Portfolio Assignment for Portfolio Type 1	Numeric
	STAT1	Portfolio Statistic for Portfolio Type 1	Numeric
PORT2	PERMNO	PERMNO	Numeric
Portfolio Data for	CALDT	Calendar Trading Date	Numeric
Portfolio Type 2	PORT2	Portfolio Assignment for Portfolio Type 2	Numeric
	STAT2	Portfolio Statistic for Portfolio Type 2	Numeric
PORT3	PERMNO	PERMNO	Numeric
Portfolio Data for	CALDT	Calendar Trading Date	Numeric
Portfolio Type 3	PORT3	Portfolio Assignment for Portfolio Type 3	Numeric
	STAT3	Portfolio Statistic for Portfolio Type 3	Numeric
PORT4	PERMNO	PERMNO	Numeric
Portfolio Data for	CALDT	Calendar Trading Date	Numeric
Portfolio Type 4	PORT4	Portfolio Assignment for Portfolio Type 4	Numeric
	STAT4	Portfolio Statistic for Portfolio Type 4	Numeric
PORT5	PERMNO	PERMNO	Numeric
Portfolio Data for	CALDT	Calendar Trading Date	Numeric
Portfolio Type 5	PORT5	Portfolio Assignment for Portfolio Type 5	Numeric
	STAT5	Portfolio Statistic for Portfolio Type 5	Numeric
PORT6	PERMNO	PERMNO	Numeric
Portfolio Data for	CALDT	Calendar Trading Date	Numeric
Portfolio Type 6	PORT6	Portfolio Assignment	Numeric

Table 1.1. (continued)

Dataset	Fields	Label	Type
	STAT6	for Portfolio Type 6 Portfolio Statistic for Portfolio Type 6	Numeric Numeric Numeric
PORT7 Portfolio Data for Portfolio Type 7	PERMNO CALDT PORT7 STAT7	PERMNO Calendar Trading Date Portfolio Assignment for Portfolio Type 7 Portfolio Statistic for Portfolio Type 7	Numeric Numeric Numeric Numeric Numeric Numeric
PORT8 Portfolio Data for Portfolio Type 8	PERMNO CALDT PORT8 STAT8	PERMNO Calendar Trading Date Portfolio Assignment for Portfolio Type 8 Portfolio Statistic for Portfolio Type 8	Numeric Numeric Numeric Numeric Numeric Numeric
PORT9 Portfolio Data for Portfolio Type 9	PERMNO CALDT PORT9 STAT9	PERMNO Calendar Trading Date Portfolio Assignment for Portfolio Type 9 Portfolio Statistic for Portfolio Type 9	Numeric Numeric Numeric Numeric Numeric Numeric

Understanding CRSP Date Formats, Informats, and Functions

CRSP has historically used two different methods to represent dates, while SAS has used a third. The three formats are SAS Dates, CRSP Dates, and Integer Dates. The SASECRSP engine provides 22 functions, 15 informats, and 10 formats to enable you to easily translate the dates from one internal representation to another. A SASECRSP libname assign must be active to use these date access methods.

SAS dates are internally stored as the number of days since January 1, 1960. The SAS method is an industry standard and provides a great deal of flexibility, including a wide variety of informats, formats, and functions.

CRSP dates are designed to ease time series storage and access. Internally the dates are stored as an offset into an array of trading days. Note that there are five different CRSP trading day calendars: Annual, Quarterly, Monthly, Weekly, and Daily. The CRSP method provides fewer missing values and makes trading period calculations very easy. However, there are also many valid calendar dates that are not available in the CRSP trading calendars, and care must be taken when using other dates.

Integer dates are a way to represent dates that are platform independent and maintain the correct sort order. However, the distance between dates is not maintained.

The best way to illustrate these formats is with some sample data. The table below only shows CRSP Daily and Monthly dates.

Table 1.2. Date Representations for Daily and Monthly Data

Date	SAS Date	CRSP Date (Daily)	CRSP Date (Monthly)	Integer Date
July 31, 1962	942	21	440	19620731
August 31, 1962	973	44	441	19620831
December 30, 1998	14,243	9190	NA*	19981230
December 31, 1998	14,244	9191	877	19981231

* Not available if an exact match is requested.

Having an understanding of the internal differences in representing SAS dates, CRSP dates, and CRSP integer dates will help you use the SASECRSP formats, informats, and functions effectively. Always keep in mind the frequency of the CRSP calendar that you are accessing when you specify a CRSP date.

The CRSP Date Formats

There are two types of formats for CRSP dates, and five frequencies are available for each of the two types. The two types are exact dates (CRSPDT*) and range dates (CRSPDR*), where * can be A for annual, Q for quarterly, M for monthly, W for weekly, or D for daily. The ten formats are: CRSPDTA, CRSPDTQ, CRSPDTM, CRSPDTW, CRSPDTD, CRSPDRA, CRSPDRQ, CRSPDRM, CRSPDRW, and CRSPDRD.

Here are some samples using the monthly and daily calendar as examples. The Annual (CRSPDTA and CRSPDRA), Quarterly (CRSPDTQ and CRSPDRQ), and the Weekly (CRSPDTW and CRSPDRW) work analogously.

Table 1.3. Sample CRSPDT Formats for Daily and Monthly Data

Date	CRSP Date Daily , Monthly	CRSPDTD8. Daily Date	CRSPDRD8. Daily Range	CRSPDTM8. Monthly Date	CRSPDRM8. Monthly Range
July 31,1962	21 , 440	19620731	19620731 +	19620731	19620630- 19620731
August31,1962	44 , 441	19620831	19620831 +	19620831	19620801- 19620831
December30, 1998	9190 , NA *	19981230	19981230 +	NA*	NA*
December31, 1998	9191 , 877	19981231	19981231 +	19981231	19981201- 19981231

+ Daily ranges will look similar to Monthly Ranges if they are Mondays or immediately following a trading holiday.

* When working with exact matches, no CRSP monthly date exists for December 30, 1998.

The @CRSP Date Informats

There are three types of informats for CRSP dates, and five frequencies are available for each of the three types. The three types are exact (@CRSPDT*), range (@CRSPDR*), and backward (@CRSPDB*) dates, where * can be A for annual, Q for quarterly, M for monthly, W for weekly, or D for daily. The fifteen formats are: @CRSPDTA, @CRSPDTQ, @CRSPDTM, @CRSPDTW, @CRSPDTD, @CRSPDRA, @CRSPDRQ, @CRSPDRM, @CRSPDRW, @CRSPDRD, @CRSPDBA, @CRSPDBQ, @CRSPDBM, @CRSPDBW, and @CRSPDBD.

The five CRSPDT* informats find exact matches only. The five CRSPDR* informats look for an exact match, and if it is not found it goes forward, matching the CRSPDR* formats. The five CRSPDB* informats look for an exact match, and if it is not found it goes backward. Here is a sample using only the CRSP monthly calendar as an example, but the daily, weekly, quarterly, and annual frequencies work analogously.

Table 1.4. Sample @CRSP Date Informat Using Monthly Data

Input Date (Integer Date)	CRSP Date CRSPDTM	CRSP Date CRSPDRM	CRSP Date CRSPDBM	CRSPDTM8. Monthly Date	CRSPDRM8. Monthly Range
19620731	440	440	440	19620731	19620630- 19620731
19620815	.(missing)	441	440	See below+	See below*
19620831	441	441	441	19620831	19620801- 19620831

+ If missing, then missing. If 441, then 19620831. If 440, then 19620731.

* If missing, then missing. If 441, then 19620801-19620831. If 440, then 19620630-19620731.

The CRSP Date Functions

There are 22 date functions provided with the SASECRSP engine. These functions are used internally by the engine, but also are available to the end users. There are six groups of functions. The first four have five functions each, one for each CRSP calendar frequency.

Table 1.5. CRSP Date Functions

Function Group	Function Name	Argument One	Argument Two	Return Value
CRSP dates to Integer dates for December 31, 1998				
Annual	crspdcia	74	None	19981231
Quarterly	crspdcicq	293	None	19981231
Monthly	crspdcim	877	None	19981231
Weekly	crspdcicw	1905	None	19981231
Daily	crspdcid	9191	None	19981231
CRSP dates to SAS dates for December 31, 1998				
Annual	crspdcsa	74	None	14,244
Quarterly	crspdcscq	293	None	14,244
Monthly	crspdcsm	877	None	14,244
Weekly	crspdcsw	1905	None	14,244
Daily	crspdcscd	9191	None	14,244
Integer dates to CRSP dates exact is illustrated, but can be forward or backward				
Annual	crspdica	19981231	0	74
Quarterly	crspdicq	19981231	0	293
Monthly	crspdicm	19981231	0	877
Weekly	crspdicw	19981231	0	1905
Daily	crspdicd	19981231	0	9191
SAS dates to CRSP dates exact is illustrated, but can be forward or backward				
Annual	crspdcscsa	14,244	0	74
Quarterly	crspdcscsq	14,244	0	293
Monthly	crspdcscsm	14,244	0	877
Weekly	crspdcscsw	14,244	0	1905
Daily	crspdcscsd	14,244	0	9191

Table 1.5. (continued)

Function Group	Function Name	Argument One	Argument Two	Return Value
Integer dates to SAS dates for December 31, 1998				
Integer to SAS	crspdi2s	19981231	None	14,244
SAS dates to Integer dates for December 31, 1998				
SAS to Integer	crspds2i	14,244	None	19981231

Examples

Example 1.1. Extracting a List of PERMNOs Using a RANGE

This example specifies a list of PERMNOs that are desired from the monthly set of CRSPAccess data. Suppose you want the range of data starting January 1, 1995, and ending June 30, 1996.

```
title2 'Define a range inside the data range ';
title3 'Valid trading dates (19890131--19981231)';
title4 'My range is ( 19950101-19960630 )';

libname testit1 sasecrsp '/mydata/crspeng/m_sampdata/'
    setid=20
    permno=81871
    permno=82200
    permno=82224
    permno=83435
    permno=83696
    permno=83776
    permno=84788
    range='19950101-19960630';

data a;
    set testit1.ask;
run;

proc print data=a;
run;
```

Output 1.1.1. Printout of the ASK Monthly Time Series Data with RANGE

The SAS System				1
Define a range inside the data range				
Valid trading dates (19890131--19981231)				
My range is (19950101-19960630)				
Obs	PERMNO	CALDT	ASK	
1	81871	19950731	18.25000	
2	81871	19950831	19.25000	
3	81871	19950929	26.00000	
4	81871	19951031	26.00000	
5	81871	19951130	25.50000	
6	81871	19951229	24.25000	
7	81871	19960131	22.00000	
8	81871	19960229	32.50000	
9	81871	19960329	30.25000	
10	81871	19960430	33.75000	
11	81871	19960531	27.50000	
12	81871	19960628	30.50000	
13	82200	19950831	49.50000	
14	82200	19950929	62.75000	
15	82200	19951031	88.00000	
16	82200	19951130	138.50000	
17	82200	19951229	139.25000	
18	82200	19960131	164.25000	
19	82200	19960229	51.00000	
20	82200	19960329	41.62500	
21	82200	19960430	61.25000	
22	82200	19960531	.	
23	82200	19960628	62.50000	
24	82224	19950929	46.50000	
25	82224	19951031	48.50000	
26	82224	19951130	47.75000	
27	82224	19951229	49.75000	
28	82224	19960131	49.00000	
29	82224	19960229	47.00000	
30	82224	19960329	53.00000	
31	82224	19960430	55.50000	
32	82224	19960531	.	
33	82224	19960628	51.00000	
34	83435	19960430	30.25000	
35	83435	19960531	.	
36	83435	19960628	21.00000	
37	83696	19960628	19.12500	

Example 1.2. Extracting All PERMNOs Using a RANGE

If you want all PERMNOs extracted for the ASK time series from the monthly data set, then you would not specify the PERMNO= option.

```
title2 'Define a range inside the data range ';
title3 'Valid trading dates (19890131--19981231)';
title4 'My range is ( 19950101-19950228 )';

libname testit2 sasecrsp '/mydata/crspeng/m_sampdata/'
    setid=20
    range='19950101-19950228';

data b;
    set testit2.ask;
run;

proc print data=b;
run;
```


Output 1.2.1. Printout of All PERMNOs of ASK Monthly with RANGE

The SAS System				1
Define a range inside the data range				
Valid trading dates (19890131--19981231)				
My range is (19950101-19950228)				
Obs	PERMNO	CALDT	ASK	
1	10078	19950131	32.75000	
2	10078	19950228	32.12500	
3	10104	19950131	42.87500	
4	10104	19950228	31.50000	
5	10107	19950131	59.37500	
6	10107	19950228	63.00000	
7	10155	19950131	2.43750	
8	10155	19950228	3.00000	
9	10837	19950131	3.12500	
10	10837	19950228	3.50000	
11	11042	19950131	21.87500	
12	11042	19950228	23.50000	
13	11081	19950131	42.62500	
14	11081	19950228	41.62500	
15	14593	19950131	40.50000	
16	14593	19950228	39.62500	
17	40484	19950131	35.75000	
18	40484	19950228	38.50000	
19	44194	19950131	14.25000	
20	44194	19950228	15.25000	
21	49606	19950131	139.00000	
22	49606	19950228	129.50000	
23	50156	19950131	44.75000	
24	50156	19950228	41.75000	
25	50404	19950131	18.37500	
26	50404	19950228	20.12500	
27	59248	19950131	16.75000	
28	59248	19950228	16.25000	
29	59328	19950131	69.37500	
30	59328	19950228	79.75000	
31	75030	19950131	21.25000	
32	75030	19950228	22.87500	
33	76535	19950131	2.75000	
34	76535	19950228	2.75000	
35	76829	19950131	21.50000	
36	76829	19950228	25.00000	
37	77352	19950131	15.25000	
38	77352	19950228	16.00000	
39	77418	19950131	54.50000	
40	77418	19950228	71.25000	
41	77882	19950131	6.50000	
42	77882	19950228	7.12500	
43	78083	19950131	32.75000	
44	78083	19950228	34.75000	
45	78117	19950131	39.25000	
46	78117	19950228	39.00000	
47	78987	19950131	22.50000	
48	78987	19950228	25.25000	
49	81181	19950131	3.75000	
50	81181	19950228	3.62500	
51	91708	19950131	.	
52	91708	19950228	.	

Example 1.3. Extracting One PERMNO Using No RANGE, Wide Open

If you want the entire range of available data for one particular PERMNO extracted from the monthly data set, then you would not specify the RANGE= option.

```
title2 'Select only PERMNO = 81871';
title3 'Valid trading dates (19890131--19981231)';
title4 'No range option, leave wide open';

libname testit3 sasecrsp '/mydata/crspeng/m_sampdata/'
    setid=20
    permno=81871;

data c;
    set testit3.ask;
run;

proc print data=c;
run;
```

Output 1.3.1. Printout of PERMNO = 81871 of ASK Monthly without RANGE

The SAS System				1
Select only PERMNO = 81871				
Valid trading dates (19890131--19981231)				
No range option, leave wide open				
Obs	PERMNO	CALDT	ASK	
1	81871	19950731	18.25000	
2	81871	19950831	19.25000	
3	81871	19950929	26.00000	
4	81871	19951031	26.00000	
5	81871	19951130	25.50000	
6	81871	19951229	24.25000	
7	81871	19960131	22.00000	
8	81871	19960229	32.50000	
9	81871	19960329	30.25000	
10	81871	19960430	33.75000	
11	81871	19960531	27.50000	
12	81871	19960628	30.50000	
13	81871	19960731	26.12500	
14	81871	19960830	19.12500	
15	81871	19960930	19.50000	
16	81871	19961031	14.00000	
17	81871	19961129	18.75000	
18	81871	19961231	24.25000	
19	81871	19970131	29.75000	
20	81871	19970228	24.37500	
21	81871	19970331	15.00000	
22	81871	19970430	18.25000	
23	81871	19970530	25.12500	
24	81871	19970630	31.12500	
25	81871	19970731	35.00000	
26	81871	19970829	33.00000	
27	81871	19970930	26.81250	
28	81871	19971031	18.37500	
29	81871	19971128	16.50000	
30	81871	19971231	16.25000	
31	81871	19980130	22.75000	
32	81871	19980227	21.00000	
33	81871	19980331	22.50000	
34	81871	19980430	16.12500	
35	81871	19980529	11.12500	
36	81871	19980630	13.43750	
37	81871	19980731	22.87500	
38	81871	19980831	17.75000	
39	81871	19980930	24.25000	
40	81871	19981030	26.00000	

Example 1.4. Extracting Selected PERMNOs in Range Using INSET= Option

You can select the PERMNOs you want to extract and specify the range of data to extract by defining an INSET such as testin2.

```
title2 'INSET=testin2 uses date ranges along with PERMNOs:';
title3 '10107, 12490, 14322, 25788';
title4 'Begin dates and end dates for each permno are used in the INSET';

data testin2;
    permno = 10107; date1 = 19980731; date2 = 19981231; output;
    permno = 12490; date1 = 19970101; date2 = 19971231; output;
    permno = 14322; date1 = 19950731; date2 = 19960131; output;
    permno = 25778; date1 = 19950101; date2 = 19950331; output;
run;

libname mstk2 sasecrsp '/mydata/crspeng/m_sampdata/' setid=20
    inset='testin2,PERMNO,PERMNO,DATE1,DATE2';

data b;
    set mstk2.prc;
run;

proc print data=b;
run;
```

Output 1.4.1. Printout of PRC Monthly Time Series Using INSET= Option

The SAS System				1
Second INSET=testin2 uses date ranges along with PERMNOs:				
10107, 12490, 14322, 25788				
Begin dates and end dates for each permno are used in the INSET				
Obs	PERMNO	CALDT	PRC	
1	10107	19980731	109.93750	
2	10107	19980831	95.93750	
3	10107	19980930	110.06250	
4	10107	19981030	105.87500	
5	10107	19981130	122.00000	
6	10107	19981231	138.68750	
7	12490	19970131	156.87500	
8	12490	19970228	143.75000	
9	12490	19970331	137.25000	
10	12490	19970430	160.50000	
11	12490	19970530	86.50000	
12	12490	19970630	90.25000	
13	12490	19970731	105.75000	
14	12490	19970829	101.37500	
15	12490	19970930	106.00000	
16	12490	19971031	98.50000	
17	12490	19971128	109.50000	
18	12490	19971231	104.62500	
19	14322	19950731	32.62500	
20	14322	19950831	32.37500	
21	14322	19950929	36.87500	
22	14322	19951031	34.00000	
23	14322	19951130	39.37500	
24	14322	19951229	39.00000	
25	14322	19960131	41.50000	
26	25778	19950131	49.87500	
27	25778	19950228	57.25000	
28	25778	19950331	59.37500	

Example 1.5. Converting Dates Using the CRSP Date Functions

This example shows how to use the CRSP date functions and formats. The CRSPDT and SASDT formats are used for all the crspdt variables, while the YYMMDD format is used for the sasdt variables.

```
title2 'OUT= Data Set';
title3 'CRSP Functions for sasecrsp';
title4 'Always assign the libname sasecrsp first';

libname mstk sasecrsp '/mydata/crspeng/m_sampdata/' setid=20;

data a (keep = crspdt crspdt2 crspdt3
           sasdt sasdt2 sasdt3
           intdt intdt2 intdt3);
  format crspdt crspdt2 crspdt3 crspdt8.;
  format sasdt sasdt2 sasdt3 yyymmdd6.;
  format intdt intdt2 intdt3 8.;
  format exact 2.;
  crspdt = 1;
  sasdt = '2jul1962'd;
  intdt = 19620702;
  exact = 0;

/* Call the CRSP date to Integer function*/
  intdt2 = crspdcid(crspdt);

/* Call the SAS date to Integer function*/
  intdt3 = crspds2i(sasdt);

/* Call the Integer to Crsp date function*/
  crspdt2 = crspdicd(intdt,exact);

/* Call the Sas date to Crsp date conversion function*/
  crspdt3 = crspds2d(sasdt,exact);

/* Call the CRSP date to SAS date conversion function*/
  sasdt2 = crspdc2d(crspdt);

/* Call the Integer to Sas date conversion function*/
  sasdt3 = crspdi2s(intdt);
run;

proc print;
run;

title2 'Proc CONTENTS showing formats for sasecrsp';
proc contents data=a;
run;
```

Output 1.5.1. Printout of Date Conversions Using the CRSP Date Functions

```

                                The SAS System                                1
                                OUT= Data Set
                                CRSP Functions for sasecrsp
                                Always assign the libname sasecrsp first

Obs crspdt   crspdt2  crspdt3   sasdt sasdt2 sasdt3   intdt   intdt2   intdt3
1  19620702 19620702 19620702 620702 620702 620702 19620702 19620702 19620702

Proc CONTENTS showing formats for sasecrsp
CRSP Functions for sasecrsp
Always assign the libname sasecrsp first

The CONTENTS Procedure

Data Set Name: WORK.A                      Observations:      1
Member Type:  DATA                      Variables:         9
Engine:       V8                        Indexes:           0
Created:      13:15 Monday, November 27, 2000 Observation Length: 72
Last Modified: 13:15 Monday, November 27, 2000 Deleted Observations: 0
Protection:                                     Compressed:      NO
Data Set Type:                                   Sorted:         NO
Label:

-----Engine/Host Dependent Information-----

Data Set Page Size:      8192
Number of Data Set Pages: 1
First Data Page:        1
Max Obs per Page:       113
Obs in First Data Page: 1
Number of Data Set Repairs: 0
File Name:               /tmp/SAS_work67560000352F/a.sas7bdat
Release Created:         8.0202M0
Host Created:            HP-UX
Inode Number:            414
Access Permission:       rw-r--r--
Owner Name:              saskff
File Size (bytes):       16384

-----Alphabetic List of Variables and Attributes-----

#   Variable   Type   Len   Pos   Format
-----
1   crspdt     Num    8     0   CRSPD8.
2   crspdt2    Num    8     8   CRSPD8.
3   crspdt3    Num    8    16   CRSPD8.
7   intdt      Num    8    48   8.
8   intdt2     Num    8    56   8.
9   intdt3     Num    8    64   8.
4   sasdt      Num    8    24   YYMDD6.
5   sasdt2     Num    8    32   YYMDD6.
6   sasdt3     Num    8    40   YYMDD6.

```

References

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

Chapter 2

The SASEFAME Interface Engine

Chapter Table of Contents

OVERVIEW	31
GETTING STARTED	31
Using Remote FAME Data Access	31
SYNTAX	32
The LIBNAME <i>libref</i> SASEFAME Statement	32
EXAMPLES	33
Example 2.1 Remote FAME Access, Using SAS CONNECT	33
Example 2.2 Remote FAME Access, Using FAME CHLI	43
REFERENCES	49

Chapter 2

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

Using Remote FAME Data Access

There are two ways to access your remote FAME data bases. The first way is to use SAS/CONNECT to submit a remote SAS session to your FAME server, and then use PROC DOWNLOAD to bring your results back to your SAS client. For an illustration, see Example 2.1. You can also refer to *Communications Access Methods for SAS/CONNECT and SAS/SHARE Software* for more about SAS/CONNECT.

The second way uses an experimental feature of the SASEFAME interface where the FAME CHLI communicates with your remote FAME server using the frdb_m port number and the node name of your FAME master server in your *libref*. For an illustration, see Example 2.2. You can also refer to “Starting the Master Server” in the *Guide to FAME Database Servers*.

Syntax

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

Description	Statement	Option
specifies the FAME frequency and the FAME technique.	LIBNAME <i>libref</i> 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 <i>libref</i> SASEFAME	WILDCARD=

The LIBNAME *libref* SASEFAME Statement

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

If you are accessing a remote FAME database using FAME CHLI, you can use the following syntax for *physical name* :

'#port number\@hostname physical path name'

Examples

Example 2.1. Remote FAME Access, Using SAS CONNECT

Suppose you are running SAS in a client/server environment and have SAS/CONNECT capability allowing you access to your FAME server. You could access your FAME remote data by signing on to your FAME server from your client session and doing a remote submit to access your FAME data. You could then use PROC DOWNLOAD to bring your remote data into the local client SAS session.

If you start your program by signing on to the remote server and checking your FAME environment variable definitions, your code might look like this:

```
options validvarname=any;
/* VALIDVARNAME set to ANY allows
   glue character in the time series names */

%let remnode=mysunbox.unx.sas.com;
/* name the sastcpd service for the connection */
signon remnode.shr2;
rsubmit;

%let FAME=%sysget(FAME);
%put(&FAME);

options validvarname=any;
libname lib1 sasefame "/usr/local/famelib/util"
              convert=(frequency=annual technique=constant)
              wildcard="?";
```

Here is the corresponding segment of the SAS log showing the remote sign-on process and the definition of the FAME environment variable on the remote host, as well as the warning generated when you use the VALIDVARNAME option allowing special characters in your SAS variable names:

Output 2.1.1. SAS Log of the SAS/CONNECT Remote Access to FAME Data

```

1          The SAS System
NOTE: Copyright (c) 1999-2000 by SAS Institute Inc., Cary, NC, USA.
NOTE: SAS (r) Proprietary Software Release 8.2 (TS02.02M0D05092000)
      Licensed to SAS Institute Inc., Site 0000000001.
NOTE: This session is executing on the HP-UX B.10.20 platform.

      options validvarname=any;
WARNING: VALIDVARNAME=ANY is an experimental option. The DATA Step and
the SQL procedure have been tested for use with this
option. Other use of this option may cause undetected errors.
2
3          %let remnode=mysunbox.unx.sas.com;
4
5
6          /* use this one on the AIX box
7          filename rlink '/users/myuser/connect/sunbox.scr';
8          */
9          /* use this one on H8X */
10         filename rlink '/u/myuser/connect/sunbox.scr';
11
12         /* options remote=remnode comamid=tcp; */
13         signon remnode.shr2;
NOTE: Remote signon to REMNODE.SHR2 commencing (SAS Release 8.02.02M0D050900).
SESSION ESTABLISHED
TRACE:      Stop;
NOTE: End of script file trace.
NOTE: Copyright (c) 1999-2000 by SAS Institute Inc., Cary, NC, USA.
NOTE: SAS (r) Proprietary Software Release 8.1 (TS01.01M0P05082000)
      Licensed to SAS Institute Y2K Testing, Site 0000000001.
NOTE: This session is executing on the SunOS 5.6 platform.
This message is contained in the SAS news file, and is presented upon
initialization. Edit the files "news" in the "misc/base" directory to
display site-specific news and information in the program log.
The command line option "-nonews" will prevent this display.
NOTE: SAS initialization used:
      real time          0.26 seconds
      cpu time           0.25 seconds

NOTE: Remote signon to REMNODE.SHR2 complete.
14         rsubmit;

NOTE: Remote submit to REMNODE.SHR2 commencing.
1         %let FAME=%sysget(FAME);
2         %put(&FAME);
(/usr/local/famelib80)
3
4         options mprint validvarname=any;
WARNING: VALIDVARNAME=ANY is an experimental option. The DATA Step and
the SQL procedure have been tested for use with this option.
Other use of this option may cause undetected errors.
5
NOTE: Libref LIB1 was successfully assigned as follows:
Engine:          FAMECHLI
Physical Name:   /usr/local/famelib/util

```

Next, you would use your libname assignment to access the remote data like this:

```

data oecd1;
    set lib1.oecd1;
run;

title2 'OECD1: PRINT Procedure';

```

```

proc print data=oecl1(obs=10);
run;

title2 'OECD1: CONTENTS Procedure';
proc contents data=oecl1;
run;

proc sort data=oecl1;
  by date;
run;

title2 'OECD1: MEANS Procedure';
proc means data=oecl1 sum;
  by date;
  var 'fin.herd'n;
run;

proc download inlib=work /* remote SASWORK */
  outlib=work; /* local SASWORK */
run;

proc datasets lib=work;
  contents data=_ALL_;
run;

endrsubmit;
signoff;

/* NOW the local work has the fame data */
proc datasets lib=work;
  contents data=_ALL_;
run;

```

Here is the remainder of the segment of the SAS log showing the download process and the remote sign-off along with the PROC DATASETS work directory results:

Output 2.1.2. SAS Log of the SAS/CONNECT Remote Access to FAME Data

```
NOTE: Download in progress from data=WORK.OECD1 to out=WORK.OECD1
NOTE: 2744 bytes were transferred at 52406 bytes/second.
NOTE: The data set WORK.OECD1 has 7 observations and 49 variables.
NOTE: Downloaded 7 observations of 49 variables.
NOTE: The data set WORK.OECD1 has 7 observations and 49 variables.
```

```
NOTE: PROCEDURE DOWNLOAD used:
      real time          2.71 seconds
      cpu time           0.04 seconds
```

```
275 proc datasets lib=work;
```

```
      -----Directory-----
Libref:          WORK
Engine:          V8
Physical Name:    /usr/tmp/SAS_work3EEB00002CC4_sunbox
File Name:        /usr/tmp/SAS_work3EEB00002CC4_sunbox
Inode Number:     212060504
Access Permission: rwxrwxrwx
Owner Name:       myuser
```

```
      -----Directory-----
File Size (bytes): 398
```

			File	
#	Name	Memtype	Size	Last Modified
1	OECD1	DATA	40960	10MAY2000:13:19:58

```
277 contents data=_ALL_;
```

```
278 run;
```

```
NOTE: Remote submit to REMNODE.SHR2 complete.
```

```
15 signoff;
```

```
NOTE: Remote signoff from REMNODE.SHR2 commencing.
```

```
NOTE: The PROCEDURE DATASETS printed pages 43-48.
```

```
NOTE: PROCEDURE DATASETS used:
```

```
      real time          0.17 seconds
      cpu time           0.09 seconds
```

```
NOTE: Remote signoff from REMNODE.SHR2 complete.
```

```
16
```

```
17 proc datasets lib=work;
```

```
      -----Directory-----
Libref:          WORK
Engine:          V8
```

```
      -----Directory-----
Physical Name:    /tmp/SAS_workDF8500003D1D
File Name:        /tmp/SAS_workDF8500003D1D
Inode Number:     1089
Access Permission: rwxr-xr-x
Owner Name:       myuser
File Size (bytes): 1024
```

			File	
#	Name	Memtype	Size	Last Modified
1	OECD1	DATA	40960	10MAY2000:13:20:03

```
      /* NOW the local work has the fame data */
```

```
18
```

```
19 contents data=_ALL_;
```

```
20
```

```
run;
```

```
21
```

```
22 endsas;
```


Output 2.1.3. Printout of the SAS/CONNECT Remote Access to FAME Data

```

***fameproc: Using FAME with a multitude of SAS Procs*** 1
OECD1: PRINT Procedure

Obs DATE   AUS.      AUT.      BEL.      CAN.
DIRDES AUS.HERD DIRDES AUT.HERD DIRDES BEL.HERD DIRDES CAN.HERD

1 1985      .      .      360.900  5990.60  334.700 14936.00 1345.90 1642
2 1986 680.400  881.70  .      .      341.300 15459.80 1435.30 1755
3 1987 725.800  983.80  .      .      369.200 16614.30 1486.50 1849
4 1988 750.000 1072.90  .      .      374.000 16572.70 1589.60 2006
5 1989      .      .      .      .      .      18310.70 1737.00 2214
6 1990      .      .      .      .      .      18874.20 1859.20 2347
7 1991      .      .      .      .      .      1959.60 2488

Obs CHE.      DEU.      DNK.      ESP.
DIRDES CHE.HERD DIRDES DEU.HERD DIRDES DNK.HERD DIRDES ESP.HERD

1      .      .      2702.10 6695.70 191.300 1873.70 335.500 31987.0
2 366.600  900      .      7120.00 211.700 2123.00 354.800 36778.0
3      .      .      3365.60 8338.50 232.700 2372.00 409.900 43667.0
4 632.100 1532 3538.60 8780.00 258.100 2662.00 508.200 55365.5
5      .      1648 3777.20 9226.60 284.800 2951.00 623.600 69270.5
6      .      .      2953.30 9700.00  .      .      723.600 78848.0
7      .      .      .      .      .      .      .      89908.0

Obs FIN.      FRA.      GBR.      GRC.
DIRDES FIN.HERD DIRDES FRA.HERD DIRDES GBR.HERD DIRDES GRC.HERD

1 183.700 1097.00 2191.60 15931.00 2068.40 1174.00  .      .
2 202.000 1234.00 2272.90 17035.00 2226.10 1281.00 44.600 3961.00
3 224.300 1401.30 2428.70 18193.00 2381.10 1397.20  .      .
4 247.700 1602.00 2573.50 19272.00 2627.00 1592.00 60.600 6674.50
5 259.700 1725.50 2856.50 21347.80 2844.10 1774.20 119.800 14485.20
6 271.000 1839.00 3005.20 22240.00  .      .      .      .
7      .      .      .      .      .      .      .      .

Obs IRL.      ISL.      ITA.      JPN.
DIRDES IRL.HERD DIRDES ISL.HERD DIRDES ITA.HERD DIRDES JPN.HERD

1 39.2000 28.3707 7.1000 268.300 1344.90 1751008 8065.70 1789780
2 46.5000 35.0000  .      .      1460.60 2004453 8290.10 1832575
3 48.1000 36.0380 7.8000 420.400 1674.40 2362102 9120.80 1957921
4 49.6000 37.0730  .      .      1861.50 2699927 9657.20 2014073
5 50.2000 39.0130 10.3000 786.762 1968.00 2923504 10405.90 2129372
6 51.7000  .      11.0000 902.498 2075.00 3183071  .      2296992
7      .      .      11.8000 990.865 2137.80 3374000  .      .

Obs NLD.      NOR.      NZL.      PRT.
DIRDES NLD.HERD DIRDES NOR.HERD DIRDES NZL.HERD DIRDES PRT.HERD

1 798.400 2032 209.000 1802.50 59.2000 80.100  .      .
2 836.000 2093  .      .      .      .      76.700 5988.90
3 886.500 2146 250.000 2165.80  .      .      .      .
4 883.000 2105  .      .      .      .      111.500 10158.20
5 945.000 2202 308.900 2771.40 78.7000 143.800  .      .
6      .      .      .      .      .      .      .      .
7      .      .      352.000 3100.00  .      .      .      .

Obs SWE.      TUR.      USA.      YUG.
DIRDES SWE.HERD DIRDES TUR.HERD DIRDES USA.HERD DIRDES YUG.HERD

1 840.10 6844 144.800 22196 14786.00 14786.00 175.900 1.87
2      .      .      136.400 26957 16566.90 16566.90 208.600 4.10
3 1016.90 8821 121.900 32309 18326.10 18326.10 237.000 10.21
4      .      .      174.400 74474 20246.20 20246.20 233.000 29.81
5 1076.00 11104 212.300 143951 22159.50 22159.50 205.100 375.22
6      .      .      .      .      23556.10 23556.10  .      2588.50
7      .      .      .      .      24953.80 24953.80  .      .

```

Output 2.1.4. PROC CONTENTS of the Remote FAME Data on the SUN Server
Node

```
***fameproc: Using FAME on Remote Server ***
      OECD1: CONTENTS Procedure
      The CONTENTS Procedure

Data Set Name: WORK.OECD1              Observations:      7
Member Type:   DATA                   Variables:        49
Engine:        V8                      Indexes:          0
Created:       13:19 Wednesday, May 10, 2000      Observation Length: 392
Last Modified: 13:19 Wednesday, May 10, 2000      Deleted Observations: 0
Protection:                                Compressed:      NO
Data Set Type:                                Sorted:         NO
Label:

      -----Engine/Host Dependent Information-----

Data Set Page Size:      32768
Number of Data Set Pages: 1
First Data Page:        1
Max Obs per Page:       83
Obs in First Data Page: 7
Number of Data Set Repairs: 0
File Name:               /usr/tmp/SAS_work3EEB00002CC4_sunbox/oecd1.sas7bdat
Release Created:         8.0101M0
Host Created:            SunOS
Inode Number:            211927040
Access Permission:       rw-rw-rw-
Owner Name:              myuser
File Size (bytes):       40960
```

Output 2.1.5. PROC CONTENTS of the Remote FAME Data on the SUN Server Node

***fameproc: Using FAME on Remote Server ***							
OECD1: CONTENTS Procedure							
The CONTENTS Procedure							
-----Alphabetic List of Variables and Attributes-----							
#	Variable	Type	Len	Pos	Format	Informat	Label
2	AUS.DIRDES	Num	8	8			
3	AUS.HERD	Num	8	16			
4	AUT.DIRDES	Num	8	24			
5	AUT.HERD	Num	8	32			
6	BEL.DIRDES	Num	8	40			
7	BEL.HERD	Num	8	48			
8	CAN.DIRDES	Num	8	56			
9	CAN.HERD	Num	8	64			
10	CHE.DIRDES	Num	8	72			
11	CHE.HERD	Num	8	80			
1	DATE	Num	8	0	YEAR4.	4.	Date of Observation
12	DEU.DIRDES	Num	8	88			
13	DEU.HERD	Num	8	96			
14	DNK.DIRDES	Num	8	104			
15	DNK.HERD	Num	8	112			
16	ESP.DIRDES	Num	8	120			
17	ESP.HERD	Num	8	128			
18	FIN.DIRDES	Num	8	136			
19	FIN.HERD	Num	8	144			
20	FRA.DIRDES	Num	8	152			
21	FRA.HERD	Num	8	160			
22	GBR.DIRDES	Num	8	168			
23	GBR.HERD	Num	8	176			
24	GRC.DIRDES	Num	8	184			
25	GRC.HERD	Num	8	192			
26	IRL.DIRDES	Num	8	200			
27	IRL.HERD	Num	8	208			
28	ISL.DIRDES	Num	8	216			
29	ISL.HERD	Num	8	224			
30	ITA.DIRDES	Num	8	232			
31	ITA.HERD	Num	8	240			
32	JPN.DIRDES	Num	8	248			
33	JPN.HERD	Num	8	256			
34	NLD.DIRDES	Num	8	264			
35	NLD.HERD	Num	8	272			
36	NOR.DIRDES	Num	8	280			
37	NOR.HERD	Num	8	288			
38	NZL.DIRDES	Num	8	296			
39	NZL.HERD	Num	8	304			
40	PRT.DIRDES	Num	8	312			
41	PRT.HERD	Num	8	320			
42	SWE.DIRDES	Num	8	328			
43	SWE.HERD	Num	8	336			
44	TUR.DIRDES	Num	8	344			
45	TUR.HERD	Num	8	352			
46	USA.DIRDES	Num	8	360			
47	USA.HERD	Num	8	368			
48	YUG.DIRDES	Num	8	376			
49	YUG.HERD	Num	8	384			

Output 2.1.6. PROC MEANS and PROC TABULATE of Remote FAME Data

fameproc: Using FAME with a multitude of SAS Procs	
OECD1: MEANS Procedure	
----- Date of Observation=1985 -----	
The MEANS Procedure	
Analysis Variable : FIN.HERD	
Sum	

1097.00	

----- Date of Observation=1986 -----	
Analysis Variable : FIN.HERD	
Sum	

1234.00	

----- Date of Observation=1987 -----	
Analysis Variable : FIN.HERD	
Sum	

1401.30	

----- Date of Observation=1988 -----	
Analysis Variable : FIN.HERD	
Sum	

1602.00	

----- Date of Observation=1989 -----	
Analysis Variable : FIN.HERD	
Sum	

1725.50	

----- Date of Observation=1990 -----	
The MEANS Procedure	
Analysis Variable : FIN.HERD	
Sum	

1839.00	

----- Date of Observation=1991 -----	
Analysis Variable : FIN.HERD	
Sum	

.	

OECD1: TABULATE Procedure	

FIN.HERD	

Sum	

8898.80	

Output 2.1.7. PROC CONTENTS of the Remote FAME Data on the HP-UX Client Node

The DATASETS Procedure			
Data Set Name:	WORK.OECD1	Observations:	7
Member Type:	DATA	Variables:	49
Engine:	V8	Indexes:	0
Created:	13:20 Wednesday, May 10, 2000	Observation Length:	392
Last Modified:	13:20 Wednesday, May 10, 2000	Deleted Observations:	0
Protection:		Compressed:	NO
Data Set Type:		Sorted:	YES
Label:			
-----Engine/Host Dependent Information-----			
Data Set Page Size:	32768		
Number of Data Set Pages:	1		
First Data Page:	1		
Max Obs per Page:	83		
Obs in First Data Page:	7		
Number of Data Set Repairs:	0		
File Name:	/tmp/SAS_workDF8500003D1D/oecd1.sas7bdat		
Release Created:	8.0202M0		
Host Created:	HP-UX		
Inode Number:	1452		
Access Permission:	rw-r--r--		
Owner Name:	myuser		
File Size (bytes):	40960		

Part 1. General Information

Output 2.1.8. PROC CONTENTS of the Remote FAME Data on the HP-UX Client Node

fameproc: Using FAME with a multitude of SAS Procs							1
OECD1: CONTENTS Procedure							
The CONTENTS Procedure							
-----Alphabetic List of Variables and Attributes-----							
#	Variable	Type	Len	Pos	Format	Informat	Label
2	AUS.DIRDES	Num	8	8			
3	AUS.HERD	Num	8	16			
4	AUT.DIRDES	Num	8	24			
5	AUT.HERD	Num	8	32			
6	BEL.DIRDES	Num	8	40			
7	BEL.HERD	Num	8	48			
8	CAN.DIRDES	Num	8	56			
9	CAN.HERD	Num	8	64			
10	CHE.DIRDES	Num	8	72			
11	CHE.HERD	Num	8	80			
1	DATE	Num	8	0	YEAR4.	4.	Date of Observation
12	DEU.DIRDES	Num	8	88			
13	DEU.HERD	Num	8	96			
14	DNK.DIRDES	Num	8	104			
15	DNK.HERD	Num	8	112			
16	ESP.DIRDES	Num	8	120			
17	ESP.HERD	Num	8	128			
18	FIN.DIRDES	Num	8	136			
19	FIN.HERD	Num	8	144			
20	FRA.DIRDES	Num	8	152			
21	FRA.HERD	Num	8	160			
22	GBR.DIRDES	Num	8	168			
23	GBR.HERD	Num	8	176			
24	GRC.DIRDES	Num	8	184			
25	GRC.HERD	Num	8	192			
26	IRL.DIRDES	Num	8	200			
27	IRL.HERD	Num	8	208			
28	ISL.DIRDES	Num	8	216			
29	ISL.HERD	Num	8	224			
30	ITA.DIRDES	Num	8	232			
31	ITA.HERD	Num	8	240			
32	JPN.DIRDES	Num	8	248			
33	JPN.HERD	Num	8	256			
34	NLD.DIRDES	Num	8	264			
35	NLD.HERD	Num	8	272			
36	NOR.DIRDES	Num	8	280			
37	NOR.HERD	Num	8	288			
38	NZL.DIRDES	Num	8	296			
39	NZL.HERD	Num	8	304			
40	PRT.DIRDES	Num	8	312			
41	PRT.HERD	Num	8	320			
42	SWE.DIRDES	Num	8	328			
43	SWE.HERD	Num	8	336			
44	TUR.DIRDES	Num	8	344			
45	TUR.HERD	Num	8	352			
46	USA.DIRDES	Num	8	360			
47	USA.HERD	Num	8	368			
48	YUG.DIRDES	Num	8	376			
49	YUG.HERD	Num	8	384			
-----Sort Information-----							
Sortedby:				DATE			
Validated:				YES			
Character Set:				ASCII			

Example 2.2. Remote FAME Access, Using FAME CHLI

Suppose you are running FAME in a client/server environment and have FAME CHLI capability allowing you access to your FAME server. You could access your FAME remote data by specifying the port number of the tcpip service that is defined for your frdb_m and the node name of your FAME master server in your physical path. In the following example, the FAME server node name is booker, and the port number is 5555, which was designated in the FAME master command. Refer to “Starting the Master Server” in the *Guide to FAME Database Servers* for more about starting your FAME master server.

```
/* DRIECON Database, Using FAME with REMOTE ACCESS VIA CHLI */
/*****/

libname test1 sasefame '#5555\@booker \ $FAME/util';
data a;
    set test1.driecon;
run;

proc contents data=a;
run;
proc means data=a n;
run;
```

Your log will show the FAME databases that are read in for conversion:

Output 2.2.1. SAS Log Segment of the FAME CHLI Remote Access to FAME Data

```

22          /* DRIECON Database, Using FAME with REMOTE ACCESS VIA HLI */
23          /*****
24
25          libname test1 sasefame '#5555\@booker \$FAME/util';
WARNING: Remote access using the FAME CHLI is an experimental feature of the
        FAME Engine. Please make sure your port number is defined correctly
        on your frdb_m master server.
NOTE: Libref TEST1 was successfully assigned as follows:
      Engine:          FAMECHLI
      Physical Name: #5555\@booker \$FAME/util
26          data a;
27          set test1.driecon;
$N -- SERIES (PRECISION by MONTHLY)
$N copied to work data base as $N.
BOPMERCH -- SERIES (PRECISION by QUARTERLY)
BOPMERCH copied to work data base as BOPMERCH.
CUSA0 -- SERIES (PRECISION by MONTHLY)
CUSA0 copied to work data base as CUSA0.
CUSA0NS -- SERIES (PRECISION by MONTHLY)
CUSA0NS copied to work data base as CUSA0NS.
DBTGFNS -- SERIES (PRECISION by MONTHLY)
DBTGFNS copied to work data base as DBTGFNS.
NOTE: DESC -- SERIES (STRING by CASE) rejected for conversion.
DJ30C -- SERIES (PRECISION by BUSINESS)
DJ30C copied to work data base as DJ30C.
DJ65CMC -- SERIES (PRECISION by BUSINESS)
DJ65CMC copied to work data base as DJ65CMC.
FBL3Y -- SERIES (PRECISION by BUSINESS)
FBL3Y copied to work data base as FBL3Y.
FCN30YY -- SERIES (PRECISION by BUSINESS)
FCN30YY copied to work data base as FCN30YY.
FIP1Q -- SERIES (PRECISION by BUSINESS)
FIP1Q copied to work data base as FIP1Q.
FIP30Q -- SERIES (PRECISION by BUSINESS)
FIP30Q copied to work data base as FIP30Q.
FSCD30Y -- SERIES (PRECISION by BUSINESS)
FSCD30Y copied to work data base as FSCD30Y.
GDP -- SERIES (PRECISION by QUARTERLY)
GDP copied to work data base as GDP.
GDP92C -- SERIES (PRECISION by QUARTERLY)
GDP92C copied to work data base as GDP92C.
GICV -- SERIES (PRECISION by MONTHLY)
GICV copied to work data base as GICV.
GNP -- SERIES (PRECISION by QUARTERLY)
GNP copied to work data base as GNP.
GNP92C -- SERIES (PRECISION by QUARTERLY)
GNP92C copied to work data base as GNP92C.
.
.
.
JLAG -- SERIES (PRECISION by MONTHLY)
JLAG copied to work data base as JLAG.
JLEAD -- SERIES (PRECISION by MONTHLY)
JLEAD copied to work data base as JLEAD.
JQIND -- SERIES (PRECISION by MONTHLY)
JQIND copied to work data base as JQIND.
.
.
.
NOTE: NAMES -- SERIES (STRING by CASE) rejected for conversion.
ZA.CP -- SERIES (PRECISION by ANNUAL)
ZB.CP -- SERIES (PRECISION by ANNUAL)
28          run;
NOTE: There were 53 observations read from the data set TEST1.DRIECON.
NOTE: The data set WORK.A has 53 observations and 53 variables.

```


Output 2.2.2. Printout of the FAME CHLI Remote Access to FAME Data

The SAS System

1

The CONTENTS Procedure

Data Set Name:	WORK.A	Observations:	53
Member Type:	DATA	Variables:	53
Engine:	V8	Indexes:	0
Created:	16:49 Friday, November 17, 2000	Observation Length:	424
Last Modified:	16:49 Friday, November 17, 2000	Deleted Observations:	0
Protection:		Compressed:	NO
Data Set Type:		Sorted:	NO
Label:			

-----Engine/Host Dependent Information-----

Data Set Page Size:	40960
Number of Data Set Pages:	1
First Data Page:	1
Max Obs per Page:	96
Obs in First Data Page:	53
Number of Data Set Repairs:	0
File Name:	/tmp/SAS_work30D40000397D/a.sas7bdat
Release Created:	8.0202M0
Host Created:	HP-UX
Inode Number:	3076
Access Permission:	rw-r--r--
Owner Name:	myuser
File Size (bytes):	49152

-----Alphabetic List of Variables and Attributes-----

#	Variable	Type	Len	Pos	Format Informat Label
2	\$N	Num	8	8	POPULATION INCLUDING ARMED FORCES OVERSEAS (P25E)
3	BOPMERCH	Num	8	16	U.S. BALANCE ON MERCHANDISE TRADE (BOP)
4	CUSA0	Num	8	24	CPI (ALL URBAN) - ALL ITEMS
5	CUSA0NS	Num	8	32	CPIU - All items
1	DATE	Num	8	0 YEAR4. 4.	Date of Observation
6	DBTGFNS	Num	8	40	
7	DJ30C	Num	8	48	DOW JONES: 30 INDUSTRIAL AVERAGE - CLOSE
8	DJ65CMC	Num	8	56	DOW JONES: 65 COMPOSITE AVERAGE
9	FBL3Y	Num	8	64	TREASURY BILL: SECONDARY, 3-MONTH BOND-EQUIVALENT YIELD (H15) - US
10	FCN30YY	Num	8	72	GOVT ISSUE: CONSTANT MATURITY, 30-YR (H15) - US
11	FIP1Q	Num	8	80	COMMERCIAL PAPER: NON-FINAN, 1-DAY QUOTED YIELD - US
12	FIP30Q	Num	8	88	COMMERCIAL PAPER: NON-FINAN, 1-MO QUOTED YIELD - US
13	FSCD30Y	Num	8	96	CD: SECONDARY MKT, 1-MO YIELD - US
14	GDP	Num	8	104	GROSS DOMESTIC PRODUCT
15	GDP92C	Num	8	112	GROSS DOMESTIC PRODUCT (CHAINED)
16	GICV	Num	8	120	NEW CONSTRUCTION PUT IN PLACE - PUBLIC TOTAL (C30)
17	GNP	Num	8	128	GROSS NATIONAL PRODUCT
18	GNP92C	Num	8	136	GROSS NATIONAL PRODUCT
19	HUCMPNC	Num	8	144	HOUSING COMPLETIONS, PRIVATE - NORTH CENTRAL (C22)
20	HUCMPNE	Num	8	152	HOUSING COMPLETIONS, PRIVATE - NORTHEAST (C22)
21	HUCMPSO	Num	8	160	HOUSING COMPLETIONS, PRIVATE - SOUTH (C22)
22	HUCMPWT	Num	8	168	HOUSING COMPLETIONS, PRIVATE-WEST(C22)

Output 2.2.3. PROC CONTENTS of the Remote FAME Data

The SAS System						2
The CONTENTS Procedure						
-----Alphabetic List of Variables and Attributes-----						
#	Variable	Type	Len	Pos	Format	Informat Label
23	HUSTS	Num	8	176		HOUSING STARTS, PRIVATE INCLUDING FARM - TOTAL (C20)
24	HUSTS1	Num	8	184		HOUSING STARTS, PRIVATE INCL FARM - ONE UNIT (C20)
25	HUSTS1NS	Num	8	192		HOUSING STARTS, PRIVATE INCL FARM - ONE UNIT (C20)
26	I	Num	8	200		GROSS PRIVATE DOMESTIC INVESTMENT
27	I92C	Num	8	208		GROSS PRIVATE DOMESTIC INVESTMENT (CHAINED)
28	ICV_G	Num	8	216		NEW CONSTRUCTION PUT IN PLACE - TOTAL (C30)
29	JCOIN%LAG	Num	8	224		RATIO, COINCIDENT INDEX TO LAGGING INDEX (BCI)
30	JLAG	Num	8	232		LAGGING INDICATORS COMPOSITE INDEX (BCI)
31	JLEAD	Num	8	240		LEADING INDICATORS COMPOSITE INDEX (BCI)
32	JQIND	Num	8	248		INDUSTRIAL PROD INDEX - TOTAL INDEX (G17)
33	JQIND20	Num	8	256		INDUSTRIAL PROD INDEX - FOODS (G17)
35	JQIND21	Num	8	272		INDUSTRIAL PROD INDEX - TOBACCO PRODUCTS (G17)
36	JQIND26	Num	8	280		INDUSTRIAL PROD INDEX - PAPER & PRODUCTS (G17)
37	JQIND28	Num	8	288		INDUSTRIAL PROD INDEX - CHEMICALS & PRODUCTS (G17)
38	JQIND37	Num	8	296		INDUSTRIAL PROD INDEX - TRANSPORTATION EQUIPMENT (G17)
39	JQIND39	Num	8	304		INDUSTRIAL PROD INDEX - MISC MANUFACTURES (G17)
34	JQIND208	Num	8	264		INDUSTRIAL PROD INDEX - BEVERAGES (G17)
40	JQINDEQPB	Num	8	312		INDUSTRIAL PROD INDEX - BUSINESS EQUIPMENT (G17)
41	MNY1	Num	8	320		MONEY SUPPLY - CURRENCY, DEMAND DEPOSITS, OTHER CHECKABLE DEPOSITS (H6)
42	MNY2	Num	8	328		MONEY SUPPLY - M2 (H6)
43	PIDGNP	Num	8	336		IMPLICIT PRICE DEFLATOR - GROSS NATIONAL PRODUCT
44	RUC	Num	8	344		UNEMPLOYMENT RATE - CIVILIAN (ESIT)
45	RXC132% USNS	Num	8	352		EXCHANGE RATE IN NEW YORK - FRENCH FRANC PER U.S. DOLLAR (G5)
46	RXC134% USNS	Num	8	360		EXCHANGE RATE IN NEW YORK - GERMAN MARK PER U.S. DOLLAR (G5)
47	RXC158% USNS	Num	8	368		EXCHANGE RATE IN NEW YORK - JAPANESE YEN PER U.S. DOLLAR (G5)
48	RXUS% C112NS	Num	8	376		EXCHANGE RATE IN NEW YORK - U.S. CENTS PER BRITISH POUND (G5)
49	STR	Num	8	384		RETAIL SALES -TOTAL (RTR)
50	WPINS	Num	8	392		PRODUCER PRICE INDEX - ALL COMMODITIES (PPI)
51	YP	Num	8	400		PERSONAL INCOME
52	ZA	Num	8	408		CORPORATE PROFITS AFTER TAX EXCLUDING IVA
53	ZB	Num	8	416		CORPORATE PROFITS BEFORE TAX EXCLUDING IVA

Output 2.2.4. PROC MEANS of the Remote Fame Data

The SAS System	4
The MEANS Procedure	
Variable	

DATE	
\$N	
BOPMERCH	
CUSA0	
CUSA0NS	
DETFNS	
DJ30C	
DJ65CMC	
FBL3Y	
FCN30YY	
FIP1Q	
FIP30Q	
FSCD30Y	
GDP	
GDP92C	
GICV	
GNP	
GNP92C	
HUCMPNC	
HUCMPNE	
HUCMPSO	
HUCMPWT	
HUSTS	
HUSTS1	
HUSTS1NS	
I	
I92C	
ICV_G	
JCOIN%LAG	
JLAG	
JLEAD	
JQIND	
JQIND20	
JQIND208	
JQIND21	
JQIND26	
JQIND28	
JQIND37	
JQIND39	
JQINDEQPB	
MNY1	
MNY2	
PIDGNP	
RUC	
RXC132%USNS	
RXC134%USNS	
RXC158%USNS	
RXUS%C112NS	
STR	
WPINS	
YP	
ZA	
ZB	

Output 2.2.5. PROC MEANS of the Remote FAME Data

The SAS System		5
The MEANS Procedure		
Label		N

Date of Observation		53
POPULATION INCLUDING ARMED FORCES OVERSEAS (P25E)		49
U.S. BALANCE ON MERCHANDISE TRADE (BOP)		39
CPI (ALL URBAN) - ALL ITEMS		52
CPIU - All items		52
		41
DOW JONES: 30 INDUSTRIAL AVERAGE - CLOSE		19
DOW JONES: 65 COMPOSITE AVERAGE		19
TREASURY BILL: SECONDARY, 3-MONTH BOND-EQUIVALENT YIELD (H15) - US		20
GOVT ISSUE: CONSTANT MATURITY, 30-YR (H15) - US		22
COMMERCIAL PAPER: NON-FINAN, 1-DAY QUOTED YIELD - US		1
COMMERCIAL PAPER: NON-FINAN, 1-MO QUOTED YIELD - US		20
CD: SECONDARY MKT, 1-MO YIELD - US		23
GROSS DOMESTIC PRODUCT		53
GROSS DOMESTIC PRODUCT (CHAINED)		52
NEW CONSTRUCTION PUT IN PLACE - PUBLIC TOTAL (C30)		35
GROSS NATIONAL PRODUCT		53
GROSS NATIONAL PRODUCT		52
HOUSING COMPLETIONS, PRIVATE - NORTH CENTRAL (C22)		20
HOUSING COMPLETIONS, PRIVATE - NORTHEAST (C22)		20
HOUSING COMPLETIONS, PRIVATE - SOUTH (C22)		20
HOUSING COMPLETIONS, PRIVATE - WEST (C22)		20
HOUSING STARTS, PRIVATE INCLUDING FARM - TOTAL (C20)		52
HOUSING STARTS, PRIVATE INCL FARM - ONE UNIT (C20)		40
HOUSING STARTS, PRIVATE INCL FARM - ONE UNIT (C20)		40
GROSS PRIVATE DOMESTIC INVESTMENT		53
GROSS PRIVATE DOMESTIC INVESTMENT (CHAINED)		52
NEW CONSTRUCTION PUT IN PLACE - TOTAL (C30)		35
RATIO, COINCIDENT INDEX TO LAGGING INDEX (BCI)		40
LAGGING INDICATORS COMPOSITE INDEX (BCI)		40
LEADING INDICATORS COMPOSITE INDEX (BCI)		40
INDUSTRIAL PROD INDEX - TOTAL INDEX (G17)		52
INDUSTRIAL PROD INDEX - FOODS (G17)		52
INDUSTRIAL PROD INDEX - BEVERAGES (G17)		45
INDUSTRIAL PROD INDEX - TOBACCO PRODUCTS (G17)		52
INDUSTRIAL PROD INDEX - PAPER & PRODUCTS (G17)		52
INDUSTRIAL PROD INDEX - CHEMICALS & PRODUCTS (G17)		52
INDUSTRIAL PROD INDEX - TRANSPORTATION EQUIPMENT (G17)		52
INDUSTRIAL PROD INDEX - MISC MANUFACTURES (G17)		52
INDUSTRIAL PROD INDEX - BUSINESS EQUIPMENT (G17)		52
MONEY SUPPLY - CURRENCY, DEMAND DEPOSITS, OTHER CHECKABLE DEPOSITS (H6)		40
MONEY SUPPLY - M2 (H6)		40
IMPLICIT PRICE DEFLATOR - GROSS NATIONAL PRODUCT		52
UNEMPLOYMENT RATE - CIVILIAN (ESIT)		9
EXCHANGE RATE IN NEW YORK - FRENCH FRANC PER U.S. DOLLAR (G5)		32
EXCHANGE RATE IN NEW YORK - GERMAN MARK PER U.S. DOLLAR (G5)		32
EXCHANGE RATE IN NEW YORK - JAPANESE YEN PER U.S. DOLLAR (G5)		32
EXCHANGE RATE IN NEW YORK - U.S. CENTS PER BRITISH POUND (G5)		32
RETAIL SALES -TOTAL (RTR)		32
PRODUCER PRICE INDEX - ALL COMMODITIES (PPI)		52
PERSONAL INCOME		53
CORPORATE PROFITS AFTER TAX EXCLUDING IVA		53
CORPORATE PROFITS BEFORE TAX EXCLUDING IVA		53

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), *Command Reference for Release 7.6, Vols. 1 and 2*, Ann Arbor, MI.
- FAME Information Services (1998), *Guide to FAME Database Servers*, New York, NY, [<http://www.fame.com>].
- FAME Information Services (1995), *Reference Guide to Seamless C HLI*, Ann Arbor, MI.
- FAME Information Services (1995), *User's Guide to FAME*, Ann Arbor, MI.
- 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.
- SAS Institute Inc. (1996), *Communications Access Methods for SAS/CONNECT and SAS/SHARE Software*, Cary, NC: SAS Institute, Inc.

Part 2

Procedure Reference

Table of Contents

Chapter 3. The ARIMA Procedure	53
Chapter 4. The MODEL Procedure	67
Chapter 5. The VARMAX Procedure	91
Chapter 6. The X12 Procedure	97

Chapter 3

The ARIMA Procedure

Chapter Table of Contents

OVERVIEW	55
SYNTAX	55
OUTLIER Statement	55
DETAILS	56
Displayed Output	58
Modeling in the Presence of Outliers	58
EXAMPLES	59
Example 3.1 Nile Data	59
Example 3.2 Airline Data	62
REFERENCES	65

Chapter 3

The ARIMA Procedure

Overview

The OUTLIER statement is a new enhancement to the ARIMA procedure. This statement is used as a diagnostic tool to detect shifts in the mean level of the response series that are not accounted for by the model in the preceding ESTIMATE statement. You can control different aspects of this detection process, such as the type and number of changes to detect, the significance level of the underlying tests, etc., by setting various options. Currently three types of changes in the mean level can be detected. These types are

- **Additive Outlier (AO)**, which represents a spurious change in one of the measurements
- **Level Shift (LS)**, which stands for a permanent shift in the level starting at some point in the observation period
- **Temporary Change (TC([d]))**, which represents a temporary change in the level of duration d

Syntax

OUTLIER Statement

OUTLIER *options*;

The OUTLIER statement applies to the model fitted in the preceding ESTIMATE statement. The following options are used in the OUTLIER statement:

TYPE = ADDITIVE | SHIFT | TEMP(d_1, \dots, d_k)

The TYPE= option specifies the types of level shifts to search for. The default is TYPE= (ADDITIVE SHIFT), which requests searching for additive outliers and permanent level shifts. The option TEMP(d_1, \dots, d_k) requests searching for temporary changes in the level of durations d_1, \dots, d_k . These options can also be abbreviated as AO, LS, and TC.

ALPHA= *significance-level*

The ALPHA= option specifies the significance level for tests in the OUTLIER statement. The default is 0.05.

SIGMA= ROBUST | MSE

The statistical tests performed during the outlier detection require an estimate of error

variance. Using the SIGMA= option you can choose between two types of error variance estimates. SIGMA= MSE corresponds to the usual mean squared error (MSE) estimate, and SIGMA= ROBUST corresponds to a robust estimate of the error variance. The default is SIGMA= ROBUST.

MAXNUM= *number*

This option is used to limit the number of outliers to search. The default is MAXNUM= 5.

MAXPCT= *number*

This option is similar to MAXNUM= option. In the MAXPCT= option you can limit the number of outliers to search to a percentage of the series length. The default is MAXPCT= 2. When both of these options are specified, the smaller of the two search numbers is used.

The following examples illustrate the various possibilities for the OUTLIER statement.

```
outlier;
```

This sets all the options to their default values, that is, it is equivalent to

```
outlier type=(ao ls) alpha=0.05 sigma=robust
      maxnum=5 maxpct=2;
```

The following statement requests a search for permanent level shifts and for temporary level changes of durations 6 and 12. The search is limited to at most three changes, and the significance level of the underlying tests is 0.001. MSE is used as the estimate of error variance.

```
outlier type=(ls tc(6 12)) alpha=0.001 sigma=mse maxnum=3;
```

Details

You can use the OUTLIER statement to detect changes in the level of the response series that are not accounted for by the currently estimated model. The types of changes considered are Additive Outliers (AO), Level Shifts (LS), and Temporary Changes (TC).

Let η_t be a regression variable describing some type of change in the mean response. In time series literature η_t is called a shock signature. An additive outlier at some time point i corresponds to a shock signature η_t such that $\eta_i = 1.0$ and η_t is 0.0 at all other points. Similarly a permanent level shift originating at time i has a shock signature such that η_t is 0.0 for $t < i$ and 1.0 for $t \geq i$. A temporary level shift of duration d originating at time i will have η_t equal to 1.0 between i and $i + d$ and 0.0 otherwise.

Suppose that the preceding ESTIMATE statement has the ARIMA model

$$D(B)Y_t = \mu_t + \frac{\theta(B)}{\phi(B)}a_t$$

where Y_t is the response series, $D(B)$ is the differencing polynomial in the backward shift operator B (possibly identity), μ_t is the transfer function input, $\phi(B)$ and $\theta(B)$ are the AR and MA polynomials, and a_t is the Gaussian white noise series.

The problem of detection of level shifts in the OUTLIER statement is formulated as a problem of sequential selection of shock signatures that improve the model in the ESTIMATE statement. This is similar to the forward selection process in the stepwise regression procedure. The selection process starts with considering shock signatures of the type specified in the TYPE= option, originating at each non-missing measurement. This involves testing $H_0: \beta = 0$ versus $H_a: \beta \neq 0$ in the model

$$D(B)(Y_t - \beta\eta_t) = \mu_t + \frac{\theta(B)}{\phi(B)}a_t$$

for each of these shock signatures. The most significant shock signature, if it also satisfies the significance criterion in ALPHA= option, is included in the model. If no significant shock signature is found then the outlier detection process stops, otherwise this augmented model, which incorporates the selected shock signature in its transfer function input, becomes the null model for the subsequent selection process. This iterative process stops if at any stage no more significant shock signatures are found or if the number of iterations exceed the maximum search number resulting due to the MAXNUM= and MAXPCT= settings. In all these iterations the parameters of the ARIMA model in the ESTIMATE statement are held fixed.

The precise details of the testing procedure for a given shock signature η_t are as follows:

The preceding testing problem is equivalent to testing $H_0: \beta = 0$ versus $H_a: \beta \neq 0$ in the following “regression with ARMA errors” model

$$N_t = \beta\zeta_t + \frac{\theta(B)}{\phi(B)}a_t$$

where $N_t = (D(B)Y_t - \mu_t)$ is the “noise” process and $\zeta_t = D(B)\eta_t$ is the “effective” shock signature.

In this setting, under H_0 , $N = (N_1, N_2, \dots, N_n)^T$ is a mean zero Gaussian vector with variance covariance matrix $\sigma^2\Sigma$. Here σ^2 is the variance of the white noise process a_t and Σ is the variance covariance matrix associated with the ARMA model. Moreover, under H_a , N has $\beta\zeta$ as the mean vector where $\zeta = (\zeta_1, \zeta_2, \dots, \zeta_n)^T$. Additionally, the generalized least squares estimate of β and its variance is given by

$$\begin{aligned}\hat{\beta} &= \delta/\kappa \\ \text{Var}(\hat{\beta}) &= \sigma^2/\kappa\end{aligned}$$

where $\delta = \zeta^T \Sigma^{-1} N$ and $\kappa = \zeta^T \Sigma^{-1} \zeta$. The test statistic $\tau^2 = \delta^2/(\sigma^2\kappa)$ is used to test the significance of β , which has an approximate chi-squared distribution with 1 degree of freedom under H_0 . The type of estimate of σ^2 used in the calculation of τ^2 can be specified by the SIGMA= option. The default setting is SIGMA=ROBUST

that corresponds to a robust estimate suggested in an outlier detection procedure in X-12-ARIMA, the Census Bureau's time series analysis program; refer to Findley et al. (1998) for additional information. The setting SIGMA=MSE corresponds to the usual mean squared error estimate (MSE) computed the same way as in the ESTIMATE statement with the NODF option. The robust estimate of σ^2 is computed by the formula

$$\hat{\sigma}^2 = (1.49 \times \text{Median}(|\hat{a}_t|))^2$$

where \hat{a}_t are the standardized residuals of the null ARIMA model.

The quantities δ and κ are efficiently computed by a method described in de Jong and Penzer (1998); refer also to Kohn and Ansley (1985).

Displayed Output

The output of the OUTLIER statement has two parts. The first part, the Outlier Summary, contains the information about the maximum number of outliers searched, the number of outliers actually detected, and the significance level used in the outlier detection. The second part, the Outlier Table, tabulates the results of the outlier detection process. The outliers are listed in the order in which they are found. This table contains five columns:

- The Obs column contains the observation number of the start of the level shift.
- The Type column lists the type of the level shift.
- The Estimate column contains $\hat{\beta}$, the estimate of the regression coefficient of the shock signature.
- The Chi-Square column lists the value of the test statistic τ^2 .
- The Approx Prob > ChiSq column lists the approximate p -value of the test statistic.

Modeling in the Presence of Outliers

In practice, modeling and forecasting time series data in the presence of outliers is a difficult problem for several reasons. The presence of outliers can adversely affect the model identification and estimation steps. Their presence close to the end of the observation period can have a serious impact on the forecasting performance of the model. In some cases level shifts are associated with changes in the mechanism driving the observation process, and separate models may be appropriate to different sections of the data. In view of all these difficulties, diagnostic tools such as outlier detection and residual analysis are essential in any modeling process.

The following modeling strategy, which incorporates level shift detection in the familiar Box-Jenkins modeling methodology, seems to work in many cases:

1. Proceed with model identification and estimation as usual. Suppose this results in a tentative ARIMA model, say M.

2. Check for suspected level shifts unaccounted for by model M using the OUTLIER statement. In this step, unless there is evidence to justify it, the number of level shifts searched should be kept small.
3. Augment the original dataset with the regression variables corresponding to the detected outliers.
4. Include the first few of these regression variables in M, and call this model M1. Re-estimate all the parameters of M1. It is important not to include too many of these outlier variables in the model in order to avoid the danger of over-fitting.
5. Check the adequacy of M1 by examining the parameter estimates, residual analysis, and outlier detection. Refine it more if necessary.

Examples

The following examples illustrate the syntax and usage of the OUTLIER statement in some simple situations.

Example 3.1. Nile Data

This example is discussed in de Jong and Penzer (1998). The data consist of readings of the annual flow volume of the Nile River at Aswan from 1871 to 1970. These data have also been studied by Cobb (1978). These studies indicate that levels in the years 1877 and 1913, that is, the 7th and 43rd measurements, are strong candidates for additive outliers, and that there was a shift in the flow levels starting from the year 1899. The year 1899 corresponds to the 29th observation. This shift in 1899 is attributed partly to the weather changes and partly to the start of construction work for a new dam at Aswan.

```
data nile;
  input level @@;
  datalines;
1120 1160 963 1210 1160 1160 813 1230 1370 1140
995 935 1110 994 1020 960 1180 799 958 1140
1100 1210 1150 1250 1260 1220 1030 1100 774 840
874 694 940 833 701 916 692 1020 1050 969
831 726 456 824 702 1120 1100 832 764 821
768 845 864 862 698 845 744 796 1040 759
781 865 845 944 984 897 822 1010 771 676
649 846 812 742 801 1040 860 874 848 890
744 749 838 1050 918 986 797 923 975 815
1020 906 901 1170 912 746 919 718 714 740
;
```

You can start the modeling process with the ARIMA(0, 1, 1) model, an ARIMA model close to the Structural model suggested in de Jong and Penzer (1998), and examine the parameter estimates, the residual autocorrelations, and the outliers.

```
proc arima data=nile;
  identify var=level(1) noprint;
  estimate q=1 noint method=ml plot;
  outlier maxnum=5;
run;
```

A portion of the estimation and the outlier detection output is shown in Output 3.1.1.

Output 3.1.1. Output from Fitting ARIMA(0, 1, 1) Model

The ARIMA Procedure					
Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag
MA1,1	0.73271	0.07132	10.27	<.0001	1
Variance Estimate			20810.22		
Std Error Estimate			144.2575		
AIC			1267.091		
SBC			1269.686		
Number of Residuals			99		
Model for variable level					
Period(s) of Differencing			1		
No mean term in this model.					
Outlier Summary					
Maximum number searched			5		
Number found			5		
Significance used			0.05		
Outlier Table					
Obs	Type	Estimate	Chi-Square	Approx Prob> ChiSq	
29	Shift	-315.75346	13.13	0.0003	
43	Additive	-403.97105	11.83	0.0006	
7	Additive	-335.49351	7.69	0.0055	
94	Additive	305.03568	6.16	0.0131	
18	Additive	-287.81484	6.00	0.0143	

Note that the first three outliers detected are indeed the ones discussed earlier. You can include the shock signatures corresponding to these three outliers in the Nile data set.

```
data nile;
  set nile;
  if _n_ = 7 then AO7 = 1.0;
  else AO7 = 0.0;
  if _n_ = 43 then AO43 = 1.0;
  else AO43 = 0.0;
  if _n_ >= 29 then LS29 = 1.0;
  else LS29 = 0.0;
run;
```


Now you can refine the earlier model by including these outliers. After examining the parameter estimates and residuals (not shown) of the ARIMA(0, 1, 1) model with these regressors, the following stationary MA1 model (with regressors) appears to fit the data well.

```
proc arima data=nile;
  identify var=level crosscorr=( AO7 AO43 LS29 ) noprint;
  estimate q=1 noint
    input=(AO7 AO43 LS29) method=ml plot;
  outlier maxnum=5 alpha=0.01;
run;
```

The relevant estimation output is shown in Output 3.1.2. No outliers, at significance level 0.01, were detected.

Output 3.1.2. MA1 Model with Outliers

The ARIMA Procedure									
Maximum Likelihood Estimation									
Parameter	Estimate	Standard Error	t Value	Pr > t	Lag	Variable	Shift		
MU	1109.9	26.65625	41.64	<.0001	0	level		0	
MA1,1	-0.19584	0.10148	-1.93	0.0536	1	level		0	
NUM1	-319.63689	116.50628	-2.74	0.0061	0	AO7		0	
NUM2	-376.48708	116.11662	-3.24	0.0012	0	AO43		0	
NUM3	-255.20007	31.32609	-8.15	<.0001	0	LS29		0	
Constant Estimate				1109.861					
Variance Estimate				13761.25					
Std Error Estimate				117.3083					
AIC				1241.659					
SBC				1254.685					
Number of Residuals				100					
Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	-----Autocorrelations-----					
6	3.43	5	0.6334	-0.000	0.007	0.074	-0.100	-0.068	-0.109
12	9.73	11	0.5552	-0.070	0.100	-0.107	-0.123	-0.083	-0.087
18	14.36	17	0.6417	0.079	-0.071	-0.025	0.132	-0.007	0.094
24	17.22	23	0.7983	0.008	-0.058	0.055	-0.049	-0.073	-0.087
Outlier Summary									
Maximum number searched					5				
Number found					0				
Significance used					0.01				

Example 3.2. Airline Data

This is a test example where an additive outlier at observation number 50 and a level shift at observation number 100 are artificially introduced in the well-known airline passenger data, the Series G in Box and Jenkins (1976).

```
data airline;
  set sashelp.air;
  logair = log(air);
  if _n_ = 50 then logair = logair - 0.25;
  if _n_ >= 100 then logair = logair + 0.5;
run;
```

The “Airline model,” $ARIMA(0, 1, 1) \times (0, 1, 1)_{12}$, is known to be a good fit to the unmodified log-transformed airline passenger series. The preliminary identification steps (not shown) again suggest the Airline model for the modified data because they exhibit a strong trend and seasonal behavior, and the Airline model has wide applicability in such cases.

```
proc arima data=airline;
  identify var=logair( 1, 12 ) noprint;
  estimate q=(1)(12) noint method=ml;
  outlier maxnum=3 alpha=0.01;
run;
```

A portion of the estimation and outlier detection output is shown in Output 3.2.1.

Output 3.2.1. Output of Airline Model

The ARIMA Procedure					
Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag
MA1,1	0.20390	0.08309	2.45	0.0141	1
MA2,1	0.76150	0.07875	9.67	<.0001	12
Variance Estimate			0.004689		
Std Error Estimate			0.068479		
AIC			-318.305		
SBC			-312.555		
Number of Residuals			131		
Model for variable logair					
Period(s) of Differencing			1,12		
No mean term in this model.					
Outlier Summary					
Maximum number searched			3		
Number found			3		
Significance used			0.01		
Outlier Table					
Obs	Type	Estimate	Chi-Square	Approx Prob> ChiSq	
100	Shift	0.49325	199.36	<.0001	
50	Additive	-0.27508	104.78	<.0001	
135	Additive	-0.10488	13.08	0.0003	

Clearly the level shift at observation number 100 and the additive outlier at observation number 50 are the dominant outliers. Moreover, the corresponding regression coefficients seem to correctly estimate the size and sign of the change. You can augment the airline data with these two regressors.

```
data airline;
  set airline;
  if _n_ = 50 then AO = 1;
  else AO = 0.0;
  if _n_ >= 100 then LS = 1;
  else LS = 0.0;
run;
```

You can now refine the previous model by including these regressors. Note that the differencing order of the dependent series is matched to the differencing orders of the outlier regressors to get the correct “effective” outlier signatures.

```
proc arima data=airline;
  identify var=logair(1, 12)
    crosscorr=( AO(1, 12) LS(1, 12) ) noprint;
  estimate q=(1)(12) noint
    input=(AO LS) method=ml plot;
  outlier maxnum=3 alpha=0.01;
run;
```

The estimation and outlier detection results are shown in Output 3.2.2.

Output 3.2.2. Airline Model with Outliers

The ARIMA Procedure							
Maximum Likelihood Estimation							
Parameter	Estimate	Standard Error	t Value	Approx Pr > t	Lag	Variable	Shift
MA1,1	0.39529	0.08117	4.87	<.0001	1	logair	0
MA2,1	0.55870	0.08678	6.44	<.0001	12	logair	0
NUM1	-0.27421	0.02761	-9.93	<.0001	0	AO	0
NUM2	0.49913	0.02985	16.72	<.0001	0	LS	0
Variance Estimate				0.001382			
Std Error Estimate				0.037173			
AIC				-482.191			
SBC				-470.69			
Number of Residuals				131			
No mean term in this model.							
Outlier Summary							
Maximum number searched					3		
Number found					3		
Significance used					0.01		
Outlier Table							
Obs	Type	Estimate	Chi-Square	Approx Prob> ChiSq			
135	Additive	-0.10310	12.63	0.0004			
62	Additive	-0.08872	12.33	0.0004			
29	Additive	0.08686	11.66	0.0006			

The output shows that a few outliers still remain to be accounted for and that the model could be refined further.

References

- Box, G.E.P. and Jenkins, G.M. (1976), *Time Series Analysis: Forecasting and Control*, San Francisco: Holden-Day.
- Cobb, G.W. (1978), "The Problem of the Nile: Conditional Solution to a Change Point Problem," *Biometrika*, 65, 243-251.
- de Jong, P. and Penzer, J. (1998), "Diagnosing Shocks in Time Series," *Journal of the American Statistical Association*, Vol. 93, No. 442.
- Fidley, 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, 127-177.
- Kohn, R. and Ansley, C. (1985), "Efficient Estimation and Prediction in Time Series Regression Models," *Biometrika*, 72, 3, 694-697.

Chapter 4

The MODEL Procedure

Chapter Table of Contents

OVERVIEW	69
SYNTAX	70
ERRORMODEL Statement	70
New FIT and SOLVE Statement Syntax	71
DETAILS	72
Multivariate t-Distribution Estimation	72
Multivariate t-Distribution Simulation	73
General Likelihood Estimation and Simulation	75
Alternate Distribution Simulation	77
Mixtures of Distributions - Copulas	79
EXAMPLES	80
Example 4.1 Cauchy Distribution Estimation	80
Example 4.2 Switching Regression Example	82
Example 4.3 Simulating from a Mixture of Distributions	86
REFERENCES	89

Chapter 4

The MODEL Procedure

Overview

The MODEL procedure estimates and simulates systems of variables with correlated residuals. In prior releases, these residuals were restricted to have normal distributions with mean zero and non-constant variance. The following new distributions are now supported for multivariate estimation:

- The centered t -distribution with estimated degrees of freedom and non-constant variance
- A user-specified likelihood function

The following distributions are now supported for multivariate Monte Carlo simulation:

- Cauchy
- Chi-squared
- Empirical
- F
- Poisson
- t
- Uniform

A transformation technique is used to create a covariance matrix for generating the correct innovations in a Monte Carlo simulation.

Syntax

ERRORMODEL Statement

ERRORMODEL *equation-name* ~ *distribution* [CDF=(*CDF options*)];

The ERRORMODEL statement is the mechanism for specifying the distribution of the residuals. You must specify the dependent/endogenous variables or general form model name, a tilde (~), and then a distribution with its parameters. The following options are used in the ERRORMODEL statement:

Options to Specify the Distribution

CAUCHY(<location, scale>)

specifies the Cauchy distribution. This option is only supported for simulation. The arguments correspond to the arguments of the SAS CDF function (ignoring the random variable argument).

CHISQUARED (*df* < , *nc*>)

specifies the χ^2 distribution. This option is only supported for simulation. The arguments correspond to the arguments of the SAS CDF function (ignoring the random variable argument).

GENERAL(*Likelihood* <, *parm1*, *parm2*, ... *parmn*>)

specifies a general log-likelihood function that you construct using SAS programming statements. *parm1*, *parm2*, ... *parmn* are optional parameters for this distribution and are used for documentation purposes only.

F(*ndf*, *ddf* < , *nc*>)

specifies the F distribution. This option is only supported for simulation. The arguments correspond to the arguments of the SAS CDF function (ignoring the random variable argument).

NORMAL($v_1 v_2 \cdots v_n$)

specifies a multivariate normal (Gaussian) distribution with mean 0 and variances v_1 through v_n .

POISSON(*mean*)

specifies the Poisson distribution. This option is only supported for simulation. The arguments correspond to the arguments of the SAS CDF function (ignoring the random variable argument).

T($v_1 v_2 \cdots v_n$, *df*)

specifies a multivariate t -distribution with noncentrality 0, variance v_1 through v_n , and common degrees of freedom *df*.

UNIFORM(<left, right>)

specifies the uniform distribution. This option is only supported for simulation. The arguments correspond to the arguments of the SAS CDF function (ignoring the random variable argument).

Options to Specify the CDF for Simulation**CDF=(CDF(options))**

specifies the univariate distribution that is used for simulation so that the estimation can be done for one set of distributional assumptions and the simulation for another. The CDF can be any of the distributions from the previous section with the exception of the General Likelihood. In addition, you can specify the empirical distribution of the residuals.

EMPIRICAL= (<TAILS=(options)>)

uses the sorted residual data to create an empirical CDF.

TAILS=(tail options)

specifies how to handle the tails in computing the inverse CDF from an empirical distribution, where *Tail options* are:

NORMAL	specifies the normal distribution to extrapolate the tails.
T(<i>df</i>)	specifies the <i>t</i> -distribution to extrapolate the tails.
PERCENT= <i>p</i>	specifies the percent of the observations to use in constructing each tail. The default for the PERCENT= option is 10. A normal distribution or a <i>t</i> -distribution is used to extrapolate the tails to infinity. The variance for the tail distributions is obtained from the data so that the empirical CDF is continuous.

New FIT and SOLVE Statement Syntax

The following option is now available on the FIT statement and is used to output a residual covariance matrix for non-normal distributions. This covariance matrix should be used in a Monte Carlo simulation and is created by converting the residuals to a standard normal distribution and using these converted residuals to compute a covariance matrix.

OUTSN= sas data-set

specifies a data set that the normalized residual covariance matrix will be written to.

The following options are now available on the SOLVE statement and apply only when Monte Carlo simulation is selected.

RESIDDATA= sas data-set

specifies a data set that contains the residuals that are to be used in the empirical distribution. This data set can be created using the OUT= option on the FIT statement.

WISHART=df

specifies that a Wishart distribution with degrees of freedom *df* be used in place of the normal error covariance matrix. This option is used to model the variance of the error covariance matrix when Monte Carlo simulation is selected.

Details

Multivariate t-Distribution Estimation

The multivariate t -distribution is specified using the `ERRORMODEL` statement with the `T` option. Other method specifications (`FIML` and `OLS`, for example) are ignored when the `ERRORMODEL` statement is used for a distribution other than normal.

The probability density function for the multivariate t -distribution is

$$P_q = \frac{\Gamma(\frac{df+m}{2})}{(\pi * df)^{\frac{m}{2}} * \Gamma(\frac{df}{2}) |\Sigma(\sigma)|^{\frac{1}{2}}} * \left(1 + \frac{\mathbf{q}'(\mathbf{y}_t, \mathbf{x}_t, \theta) \Sigma(\sigma)^{-1} \mathbf{q}(\mathbf{y}_t, \mathbf{x}_t, \theta)}{df} \right)^{-\frac{df+m}{2}}$$

where m is the number of equations and df is the degrees of freedom.

The maximum likelihood estimators of θ and σ are the $\hat{\theta}$ and $\hat{\sigma}$ that minimize the negative log-likelihood function:

$$\begin{aligned} l_n(\theta, \sigma) = & - \sum_{t=1}^n \ln \left(\frac{\Gamma(\frac{df+m}{2})}{(\pi * df)^{\frac{m}{2}} * \Gamma(\frac{df}{2})} * \left(1 + \frac{q'_t \Sigma^{-1} q_t}{df} \right)^{-\frac{df+m}{2}} \right) \\ & + \frac{n}{2} * \ln(|\Sigma|) - \sum_{t=1}^n \ln \left(\left| \frac{\partial q_t}{\partial y'_t} \right| \right) \end{aligned}$$

The `ERRORMODEL` statement is used to request the t -distribution maximum likelihood estimation. An `OLS` estimation is done to obtain initial parameter estimates and `MSE.var` estimates. Use `NOOLS` to turn off this initial estimation. If the errors are distributed normally, t -distribution estimation will produce results similar to `FIML`.

The multivariate model has a single shared degrees of freedom parameter, which is estimated. The degrees of freedom parameter can also be set to a fixed value. The negative log-likelihood value and the l_2 norm of the gradient of the negative log-likelihood function are shown in the estimation summary.

t-Distribution Details

Since a variance term is explicitly specified using the `ERRORMODEL` statement, $\Sigma(\theta)$ is estimated as a correlation matrix and $\mathbf{q}(\mathbf{y}_t, \mathbf{x}_t, \theta)$ is normalized by the variance.

The gradient of the negative log-likelihood function with respect to the degrees of freedom is

$$\begin{aligned} \frac{\partial l_n}{\partial df} = & \frac{nm}{2 df} - \frac{n}{2} \Gamma'(\frac{df+m}{2}) + \frac{n}{2} \Gamma'(\frac{df}{2}) + \\ & 0.5 \log(1 + \frac{\mathbf{q}' \Sigma^{-1} \mathbf{q}}{df}) - \frac{0.5(df+m)}{(1 + \frac{\mathbf{q}' \Sigma^{-1} \mathbf{q}}{df})} \frac{\mathbf{q}' \Sigma^{-1} \mathbf{q}}{df^2} \end{aligned}$$

The gradient of the negative log-likelihood function with respect to the parameters is

$$\frac{\partial l_n}{\partial \theta_i} = \frac{0.5(df + m)}{(1 + \mathbf{q}'\Sigma^{-1}\mathbf{q}/df)} \left[\frac{(2\mathbf{q}'\Sigma^{-1}\frac{\partial \mathbf{q}}{\partial \theta_i})}{df} + \mathbf{q}'\Sigma^{-1}\frac{\partial \Sigma}{\partial \theta_i}\Sigma^{-1}\mathbf{q} \right] - \frac{n}{2}\text{trace}(\Sigma^{-1}\frac{\partial \Sigma}{\partial \theta_i})$$

where

$$\frac{\partial \Sigma(\theta)}{\partial \theta_i} = \frac{2}{n} \sum_{t=1}^n \mathbf{q}(\mathbf{y}_t, \mathbf{x}_t, \theta) \frac{\partial \mathbf{q}(\mathbf{y}_t, \mathbf{x}_t, \theta)'}{\partial \theta_i}$$

and

$$\mathbf{q}(\mathbf{y}_t, \mathbf{x}_t, \theta) = \frac{\epsilon(\theta)}{\sqrt{h(\theta)}} \in R^{m \times n}$$

The estimator of the variance-covariance of $\hat{\theta}$ (COVB) for the t -distribution is the inverse of the likelihood Hessian. The gradient is computed analytically and the Hessian is computed numerically.

Multivariate t -Distribution Simulation

To perform a Monte Carlo analysis of models that have residuals distributed as a multivariate t , use the `ERRORMODEL` statement with either the `~ t(variance, df)` option or with the `CDF=t(variance, df)` option. The `CDF=` option specifies the distribution that is used for simulation so that the estimation can be done for one set of distributional assumptions and the simulation for another.

The following is an example of estimating and simulating a system of equations with t -distributed errors using the `ERRORMODEL` statement:

```

/* generate simulation data set */
data five;
set xfrate end=last;
if last then do;
  todate = date +5;
  do date = date to todate;
    output;
  end;
end;
end;
```

The preceding `DATA` step generates the data set to request a five-days-ahead forecast. The following statements estimate and forecast the three forward-rate models of the form

$$\begin{aligned} rate_t &= rate_{t-1} + \mu * rate_{t-1} + \nu \\ \nu &= \sigma * rate_{t-1} * \epsilon \\ \epsilon &\sim N(0, 1) \end{aligned}$$

```

Title "Daily Multivariate Geometric Brownian Motion Model "
      "of D-Mark/USDollar Forward Rates";

proc model data=xfrate;

    parms df 15;          /* Give initial value to df */

    demusd1m = lag(demusd1m) + mulm * lag(demusd1m);
    var_demusd1m = sigmal1m ** 2 * lag(demusd1m **2);
    demusd3m = lag(demusd3m) + mu3m * lag(demusd3m);
    var_demusd3m = sigma3m ** 2 * lag(demusd3m ** 2);
    demusd6m = lag(demusd6m) + mu6m * lag(demusd6m);
    var_demusd6m = sigma6m ** 2 * lag(demusd6m ** 2);

    /* Specify the error distribution */
    errormodel demusd1m demusd3m demusd6m
        ~ t( var_demusd1m var_demusd3m var_demusd6m, df );

    /* output normalized S matrix */
    fit demusd1m demusd3m demusd6m / outsn=s;
run;

    /* forecast five days in advance */
    solve demusd1m demusd3m demusd6m /
        data=five sdata=s random=1500 out=monte;
    id date;
run;

    /* select out the last date -----*/
    data monte; set monte;
    if date = '10dec95'd then output;
run;

title "Distribution of demusd1m Five Days Ahead";
proc univariate data=monte noprint;
    var demusd1m ;
    histogram demusd1m / normal(noprint color=red)
        kernel(noprint color=blue) cfill=ligr;
run;

```

The Monte Carlo simulation specified in the preceding example draws from a multivariate t -distribution with constant degrees of freedom and forecasted variance and computes future states of DEMUSD1M, DEMUSD3M, and DEMUSD6M. The OUTSN= option on the FIT statement is used to specify the data set for the normalized Σ matrix. That is the Σ matrix is created by crossing the normally distributed residuals. The normally distributed residuals are created from the t -distributed residuals using the normal inverse CDF and the t CDF.

The distribution of DEMUSD1M on the fifth day is shown in Figure 4.1. The two curves overlayed on the graph are a kernel density estimation and a normal distribution fit to the results.

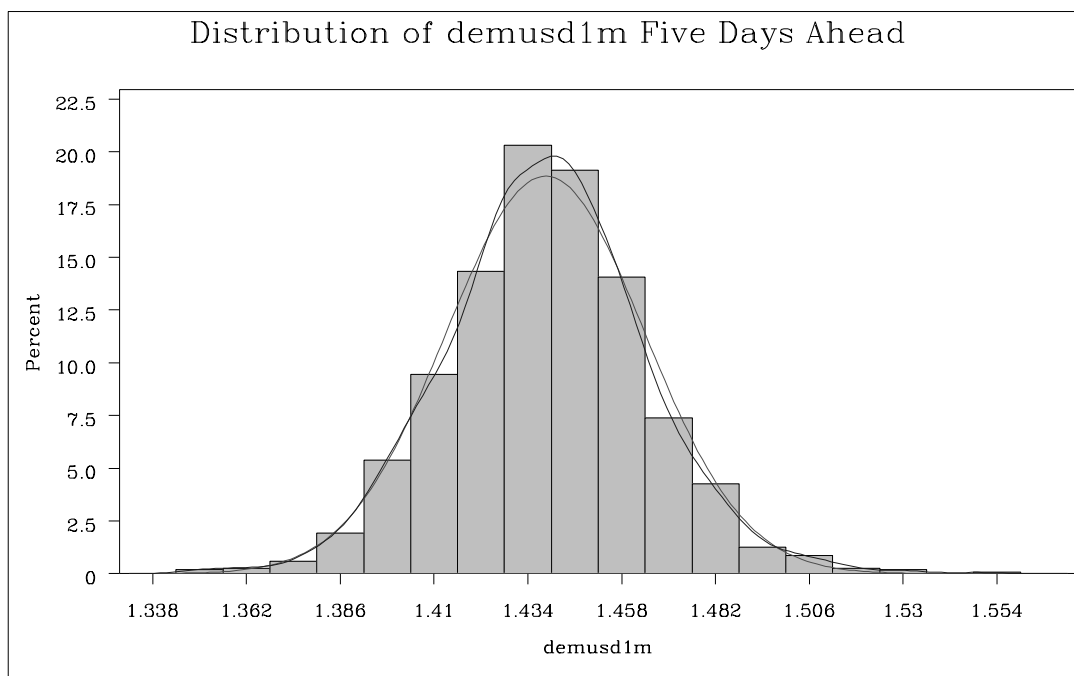


Figure 4.1. Distribution of DEMUSD1M

General Likelihood Estimation and Simulation

The `GENERAL` option on the `ERRORMODEL` statement can be used to estimate distributions other than normal and t . The `GENERAL` option on the `ERRORMODEL` statement expects as its first argument a log-likelihood function.

When a general log-likelihood function is specified, a quasi-Newton method is used to find the minimum of the function. The standard errors are computed using the Hessian of the log-likelihood function at the solution. It is important to include all scale terms in the likelihood if correct standard errors are desired. The `OUTS=` data set is created from the cross of the residuals normalized according to the `VARDEF=` option. This `S` matrix is not normalized by any implied variance terms in the specified likelihood. This is important to consider if the `S` matrix is used in a Monte Carlo simulation.

For simulation and forecasting, the likelihood function is ignored. If Monte Carlo is requested, a CDF function can be specified using the `CDF=` option. If no CDF is specified, an empirical distribution with normal tails is used.

Empirical Distribution Estimation and Simulation

For simulation, if the CDF for the model is not built in to the procedure, you can use the `CDF=EMPIRICAL()` option. This uses the sorted residual data to create an empirical CDF. For computing the inverse CDF the program needs to know how to handle the tails. For continuous data, the tail distribution is generally poorly determined. To counter this, the `PERCENT=` option specifies the percent of the observations to use in constructing each tail. The default for the `PERCENT=` option is 10. A normal

distribution or a t -distribution is used to extrapolate the tails to infinity. The standard errors for this extrapolation are obtained from the data so that the empirical CDF is continuous.

The following SAS statements fit a model using least squares as the likelihood function, but represent the distribution of the residuals with an empirical CDF. The plot of the empirical probability distribution is shown in Figure 4.2.

```
data t; /* Sum of two normals */
  format date monyy.;
  do t=0 to 3 by 0.1;
    date = intnx( 'month', '1jun90'd,(t*10)-1);
    y = (0.1 * rannor(123)-10) +
        ( .5 * rannor(456)+10);
    output;
  end;
run;

proc model data=t time=t itprint;
  dependent y;
  parm a 5 ;
  y = a;
  obj = resid.y * resid.y;
  errormodel y ~ general( obj )
    cdf=(empirical=( tails=( t(15) percent= 5)));

  fit y / outns=s out=r;
  id date;
  solve y / data=t(where=(date='1jun95'd ))
    residdata=r sdata=s random=200 seed=6789 out=monte;
run;

/* Generate the pdf -----*/
proc kde data =monte out=density;
  var y;
run;

symbol1 value=none interpol=join;
proc gplot data=density;
  plot density*y ;
run;
```

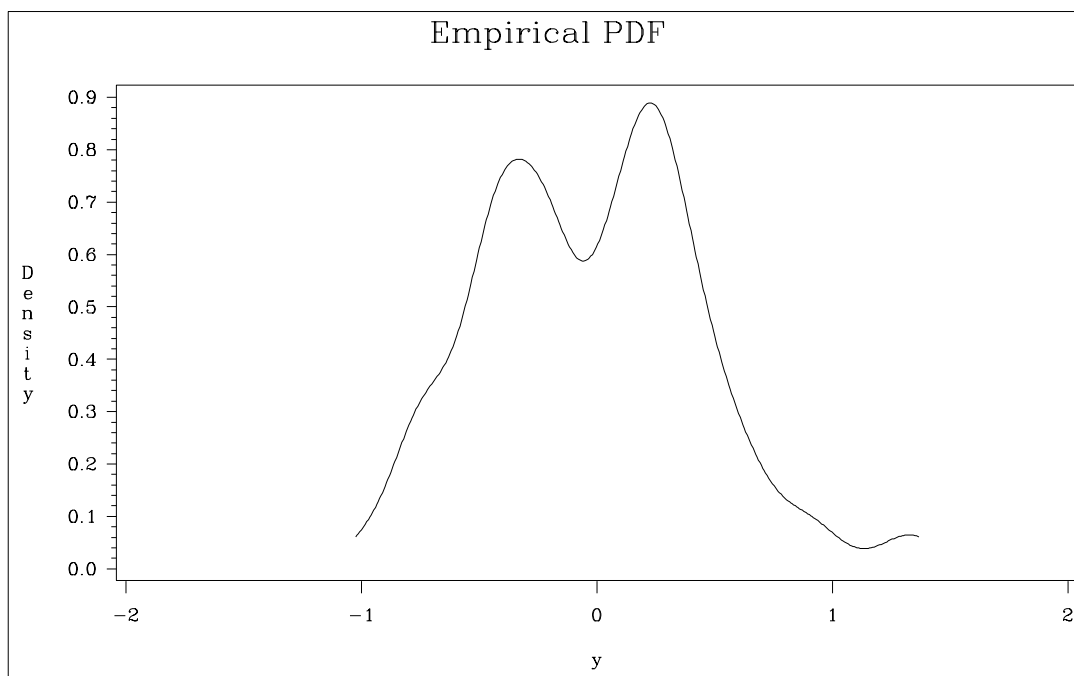



Figure 4.2. Empirical PDF Plot

Alternate Distribution Simulation

As an alternate to the normal distribution, the `ERRORMODEL` statement can be used in a simulation to specify other distributions. The distributions available for simulation are Cauchy, Chi-squared, F , Poisson, t , and Uniform. An empirical distribution can also be used if the residuals are specified using the `RESIDDATA=` option on the `SOLVE` statement.

Except for the t , all of these alternate distributions are univariate but can be used together in a multivariate simulation. The `ERRORMODEL` statement applies to solved for equations only. That is, the normal form or general form equation referred to by the `ERRORMODEL` statement must be one of the equations you have selected in the `SOLVE` statement.

In the following example, two Poisson distributed variables are used to simulate the calls arriving and leaving a call center.

```
data s;          /* Covariance between arriving and leaving */
  arriving = 1; leaving = 0.7; _name_ = "arriving";
  output;
  arriving = 0.7; leaving = 1.0; _name_ = "leaving";
  output;
run;

data calls;
  date = '20mar2001'd;
  output;
run;
```

Part 2. Procedure Reference

The first DATA step generates a data set containing a covariance matrix for the AR-RIVING and LEAVING variables. The covariance is

$$\begin{vmatrix} 1 & .7 \\ .7 & 1 \end{vmatrix}$$

```
proc model data=calls;
  arriving = 10;
  errormodel arriving ~ poisson( 10 );
  /* Have four people answering the phone */
  leaving = 4;
  errormodel leaving ~ poisson( 11 );

  waiting = arriving - leaving;
  solve arriving leaving / random=500 sdata=s out=sim;
run;

title "Distribution of Clients Waiting";
proc univariate data=sim noprint;
  var waiting ;
  histogram waiting / cfill=ligr;
run;
```

The distribution of number of waiting clients is shown in Figure 4.3.

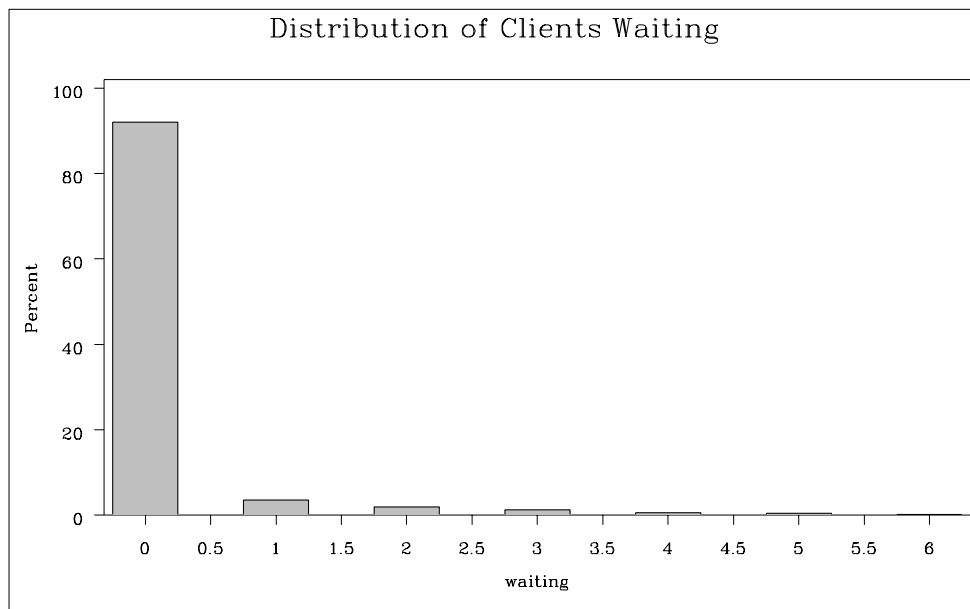


Figure 4.3. Distribution of Number of Clients Waiting

Mixtures of Distributions - Copulas

The theory of copulas is what enables the MODEL procedure to combine and simulate multivariate distributions with different marginals. This section provides a brief overview of copulas.

Modeling a system of variables accurately is a difficult task. The underlying, ideal, distributional assumptions for each variable are usually different from each other. An individual variable may be best modeled as a t -distribution or as a Poisson process. The correlation of the various variables are very important to estimate as well. A joint estimation of a set of variables would make it possible to estimate a correlation structure but would restrict the modeling to single, simple multivariate distribution (for example, the normal). Even with a simple multivariate distribution, the joint estimation would be computationally difficult and would have to deal with issues of missing data.

Using the MODEL procedure ERRORMODEL statement you can combine and simulate from models of different distributions. The covariance matrix for the combined model is constructed using the copula induced by the multivariate normal distribution. A copula is a function that couples joint distributions to their marginal distributions.

The copula used by the model procedure is based on the multivariate normal. This particular multivariate normal has zero mean and covariance matrix R . The user provides R , which can be created using the following steps

1. Each model is estimated separately and their residuals saved.
2. The residuals for each model are converted to a normal distribution using their CDFs, $F_i(\cdot)$, using the relationship $\Phi^{-1}(F(\epsilon_{it}))$.
3. Cross these normal residuals, to create a covariance matrix R .

If the model of interest can be estimated jointly, such as multivariate T, then the OUTSN= option can be used to generate the correct covariance matrix.

A draw from this mixture of distributions is created using the following steps that are performed automatically by the MODEL procedure.

1. Independent $N(0, 1)$ variables are generated.
2. These variables are transformed to a correlated set using the covariance matrix R .
3. These correlated normals are transformed to a uniform using $\Phi(\cdot)$.
4. $F^{-1}(\cdot)$ is used to compute the final sample value.

Examples

Example 4.1. Cauchy Distribution Estimation

In this example a nonlinear model is estimated using the Cauchy distribution. Then a simulation is done for one observation in the data.

The following DATA step creates the data for the model.

```

                /* Generate a Cauchy distributed Y */
data c;
  format date monyy.;
  call streaminit(156789);
  do t=0 to 20 by 0.1;
    date=intnx('month','01jun90'd,(t*10)-1);
    x=rand('normal');
    e=rand('cauchy') + 10 ;
    y=exp(4*x)+e;
    output;
  end;
run;

```

The model to be estimated is

$$y = e^{-a x} + \epsilon$$

$$\epsilon \sim \text{Cauchy}(nc)$$

That is, the residuals of the model are distributed as a Cauchy distribution with non-centrality parameter nc .

The log likelihood for the Cauchy distribution is

$$like = -\log(1 + (x - nc)^2 * \pi)$$

The following SAS statements specify the model and the log-likelihood function.

```

title2 'Cauchy Distribution';

proc model data=c ;
  dependent y;

  parm a -2 nc 4;

  y=exp(-a*x);

                /* Likelihood function for the residuals */
  obj = log(1+(-resid.y-nc)**2 * 3.1415926);

```

```

errormodel y ~ general(obj) cdf=cauchy(nc);

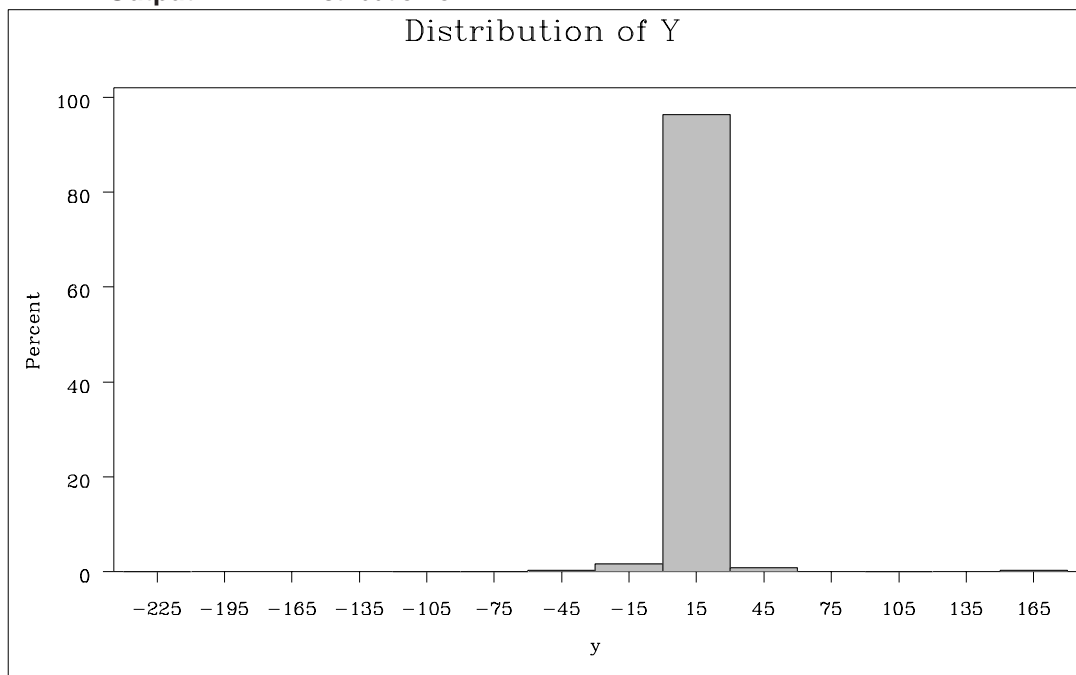
fit y / outsn=s1 method=marquardt;
solve y / sdata=s1 data=c(obs=1) random=1000
      seed=256789 out=out1;
run;

```

The FIT statement uses the OUTSN= option to put out the Σ matrix for residuals from the normal distribution. The Σ matrix is 1×1 and has value 4.65824. The OUTS= matrix is the scalar 2989.0. Because the distribution is univariate (no covariances), the OUTS= would produce the same simulation results. The simulation is performed using the SOLVE statement.

The distribution of y is shown in Output 4.1.1.

Output 4.1.1. Distribution of Y



Example 4.2. Switching Regression Example

Take the usual linear regression problem

$$y = X\beta + u$$

where Y denotes the n column vector of the dependent variable, X denotes the $(n \times k)$ matrix of independent variables, β denotes the k column vector of coefficients to be estimated, n denotes the number of observations ($i=1,2,\dots,n$), and k denotes the number of independent variables.

You can take this basic equation and split it into two regimes, where the i th observation on y is generated by one regime or the other.

$$y_i = \sum_{j=1}^k \beta_{1j} X_{ji} + u_{1i} = x_i' \beta_1 + u_{1i}$$

$$y_i = \sum_{j=1}^k \beta_{2j} X_{ji} + u_{2i} = x_i' \beta_2 + u_{2i}$$

where x_{hi} and x_{hj} are the i th and j th observations, respectively, on x_h . The errors, u_{1i} and u_{2i} , are assumed to be distributed normally and independently, with mean zero and constant variance. The variance for the first regime is σ_1^2 , and the variance for the second regime is σ_2^2 . If $\sigma_1^2 \neq \sigma_2^2$ and $\beta_1 \neq \beta_2$, the regression system given previously is thought to be switching between the two regimes.

The problem is to estimate β_1 , β_2 , σ_1 , and σ_2 without knowing *a priori* which of the n values of the dependent variable, y , was generated by which regime. If it is known *a priori* which observations belong to which regime, a simple Chow test can be used to test $\sigma_1^2 = \sigma_2^2$ and $\beta_1 = \beta_2$.

Using Goldfeld and Quandt's D-method for switching regression, you can solve this problem. Assume that there exists observations on some exogenous variables $z_{1i}, z_{2i}, \dots, z_{pi}$, where z determines whether the i th observation is generated from one equation or the other.

$$y_i = x_i' \beta_1 + u_{1i} \quad \text{if } \sum_{j=1}^p \pi_j z_{ji} \leq 0$$

$$y_i = x_i' \beta_2 + u_{2i} \quad \text{if } \sum_{j=1}^p \pi_j z_{ji} > 0$$

where π_j are unknown coefficients to be estimated. Define $d(z_i)$ as a continuous approximation to a step function. Replacing the unit step function with a continuous

approximation using the cumulative normal integral enables a more practical method that produces consistent estimates.

$$d(z_i) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\sum \pi_j z_{ji}} \exp \left[-\frac{1}{2} \frac{\xi^2}{\sigma^2} \right] d\xi$$

D is the n dimensional diagonal matrix consisting of $d(z_i)$.

$$D = \begin{bmatrix} d(z_1) & 0 & 0 & 0 \\ 0 & d(z_2) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & d(z_n) \end{bmatrix}$$

The parameters to estimate are now the k β_1 's, the k β_2 's, σ_1^2 , σ_2^2 , p π 's, and the σ introduced in the $d(z_i)$ equation. The σ can be considered as given *a priori*, or it can be estimated, in which the estimated magnitude provides an estimate of the success in discriminating between the two regimes (Goldfeld and Quandt 1976).

$$Y = (I - D)X\beta_1 + DX\beta_2 + W$$

where $W = (I - D)U_1 + DU_2$, and W is a vector of unobservable and heteroscedastic error terms. The covariance matrix of W is denoted by Ω , where $\Omega = (I - D)^2\sigma_1^2 + D^2\sigma_2^2$. The maximum likelihood parameter estimates maximize the following log-likelihood function.

$$\begin{aligned} \log L = & -\frac{n}{2} \log 2\pi - \frac{1}{2} \log |\Omega| - \\ & \frac{1}{2} * [[Y - (I - D)X\beta_1 - DX\beta_2]' \Omega^{-1} [Y - (I - D)X\beta_1 - DX\beta_2]] \end{aligned}$$

As an example, you now can use this switching regression likelihood to develop a model of housing starts as a function of changes in mortgage interest rates. The data for this example is from the U.S. Census Bureau and covers the period from January 1973 to March 1999. The hypothesis is that there will be different coefficients on your model based on whether the interest rates are going up or down.

So the model for z_i will be the following

$$z_i = p * (rate_i - rate_{i-1})$$

where $rate_i$ is the mortgage interest rate at time i and p is a scale parameter to be estimated.

The regression model will be the following

$$\begin{aligned} starts_i &= intercept_1 + ar1 * starts_{i-1} + djf1 * decjanfeb & z_i < 0 \\ starts_i &= intercept_2 + ar2 * starts_{i-1} + djf2 * decjanfeb & z_i \geq 0 \end{aligned}$$

where $starts_i$ is the number of housing starts at month i and $decjanfeb$ is a dummy variable indicating that the current month is one of December, January, or February.

This model is written using the following SAS statements.

```
proc model data=switch;
  parms sig1=10 sig2=10 int1 b11 b13 int2 b21 b23 p;
  bounds 0.0001 < sig1 sig2;

  a = p*dif(rate);          /* Upper bound of integral */
  d = probnorm(a);          /* Normal CDF as an approx of switch */

                                /* Regime 1 */
  y1 = int1 + zlag(starts)*b11 + decjanfeb *b13 ;
                                /* Regime 2 */
  y2 = int2 + zlag(starts)*b21 + decjanfeb *b23 ;
                                /* Composite regression equation */
  starts = (1 - d)*y1 + d*y2;

                                /* Resulting log-likelihood function */
  logL = (1/2)*( (318*log(2*3.1415)) +
    log( (sig1**2)*((1-d)**2)+(sig2**2)*(d**2) )
    + (resid.starts*( 1/( (sig1**2)*((1-d)**2)+
      (sig2**2)*(d**2) ) )*resid.starts) ) ;

  errormodel starts ~ general(logL);

fit starts / method=marquardt converge=1.0e-5;

  /* Test for significant differences in the parms */
test int1 = int2 ,/ lm;
test b11 = b21 ,/ lm;
test b13 = b23 ,/ lm;
test sig1 = sig2 ,/ lm;

run;
```

Four TEST statements were added to test the hypothesis that the parameters were the same in both regimes. The parameter estimates and ANOVA table from this run are shown in Output 4.2.1.

Output 4.2.1. Parameter Estimates from the Switching Regression

Nonlinear Likelihood Summary of Residual Errors							
Equation	DF Model	DF Error	SSE	MSE	Root MSE	R-Square	Adj R-Sq
starts	9	304	85877.9	282.5	16.8075	0.7806	0.7748
Nonlinear Likelihood Parameter Estimates							
Parameter	Estimate	Approx Std Err	t Value	Approx Pr > t			
sig1	15.47451	0.9475	16.33	<.0001			
sig2	19.77797	1.2710	15.56	<.0001			
int1	32.82232	5.9070	5.56	<.0001			
b11	0.739529	0.0444	16.65	<.0001			
b13	-15.456	3.1909	-4.84	<.0001			
int2	42.73243	6.8153	6.27	<.0001			
b21	0.734112	0.0477	15.37	<.0001			
b23	-22.5178	4.2979	-5.24	<.0001			
p	25.94332	8.5181	3.05	0.0025			

The test results shown in Output 4.2.2 suggest that the variance of the housing starts, SIG1 and SIG2, are significantly different in the two regimes. The tests also show a significant difference in the AR term on the housing starts.

Output 4.2.2. Parameter Estimates from the Switching Regression

Test Results				
Test	Type	Statistic	Pr > ChiSq	Label
Test0	L.M.	0.02	0.8810	int1 = int2
Test1	L.M.	240001	<.0001	b11 = b21
Test2	L.M.	0.02	0.8933	b13 = b23
Test3	L.M.	319354	<.0001	sig1 = sig2

Example 4.3. Simulating from a Mixture of Distributions

This example illustrates how to perform a multivariate simulation using models that have different error distributions. Three models are used. The first model has t -distributed errors. The second model is a GARCH(1,1) model with normally distributed errors. The third model has a non-central Cauchy distribution.

The following SAS statements generate the data for this example. The T and the CAUCHY data sets use a common seed so that those two series will be correlated.

```
%let df = 7.5;
%let sig1 = .5;
%let var2 = 2.5;

data t;
  format date monyy.;
  do date='1jun2001'd to '1nov2002'd;
    /* t-distribution with df,sig1 */
    t = .05 * date + 5000 + &sig1*tinv(ranuni(1234),&df);
    output;
  end;
run;

data normal;
  format date monyy.;
  le = &var2;
  lv = &var2;
  do date='1jun2001'd to '1nov2002'd;
    /* Normal with GARCH error structure */
    v = 0.0001 + 0.2 * le**2 + .75 * lv;
    e = sqrt( v ) * rannor(12345) ;
    normal = 25 + e;
    le = e;
    lv = v;
    output;
  end;
run;

data cauchy;
  format date monyy.;
  PI = 3.1415926;
  do date='1jun2001'd to '1nov2002'd;
    cauchy = -4 + tan((ranuni(1234) - 0.5) * PI);
    output;
  end;
run;
```

Since the multivariate joint likelihood is unknown, the models must be estimated separately. The residuals for each model are saved using the OUT= option. Also, each model is saved using the OUTMODEL= option. The ID statement is used to provide a variable in the residual data set to merge by. The XLAG function is used

to model the GARCH(1,1) process. The XLAG function returns the lag of the first argument if it is nonmissing, otherwise it returns the second argument.

```
proc model data=t outmod=t;
parms df 10 vt 4;
    t = a * date + c;
    errormodel t ~ t( vt, df );
fit t / out=tresid;
id date;
run;

proc model data=normal outmod=normal;
    normal = b0 ;
    h.normal = arch0 + arch1 * xlag(resid.normal **2 , mse.normal) +
        GARCH1 * xlag(h.normal, mse.normal);

fit normal /fiml out=nresid;
id date;
run;

proc model data= cauchy outmod=cauchy;
parms nc = 1;
    /* nc is noncentrality parm to Cauchy dist */
    cauchy = nc;
    obj = log(1+resid.cauchy**2 * 3.1415926);
    errormodel cauchy ~ general(obj) cdf=cauchy(nc);

fit cauchy / out=cresid;
id date;
run;
```

The simulation requires a covariance matrix created from normal residuals. The following Data Step code uses the inverse CDFs of the t and Cauchy distributions to convert the residuals to the normal distribution. The CORR procedure is used to create a correlation matrix using the converted residuals.

```
/* Merge and normalize the 3 residual data sets */
data c; merge tresid nresid cresid; by date;
t = probit(cdf("T", t/sqrt(0.2789), 16.58 ));
cauchy = probit(cdf("CAUCHY", cauchy, -4.0623));
run;

proc corr data=c out=s;
var t normal cauchy;
run;
```

Now the models can be simulated together using the MODEL procedure SOLVE statement. The data set created by the CORR procedure is used as the correlation matrix. Note that the errormodel statement is not saved with the model and must be restated for this simulation.

```

/* Create one observation driver data set */
data sim; merge t normal cauchy; by date;
data sim; set sim(firstobs = 519 );

proc model data=sim model=( t normal cauchy);
  errormodel t ~ t( vt, df );
  errormodel cauchy ~ cauchy(nc);
solve t cauchy normal / random=2000 seed=1962 out=monte
  sdata=s(where=(_type_="CORR"));
run;

```

An estimation of the joint density of the t and Cauchy distribution is created using the KDE procedure. Bounds were placed on the Cauchy dimension because of its fat tail behavior. The joint PDF is shown in Output 4.3.1.

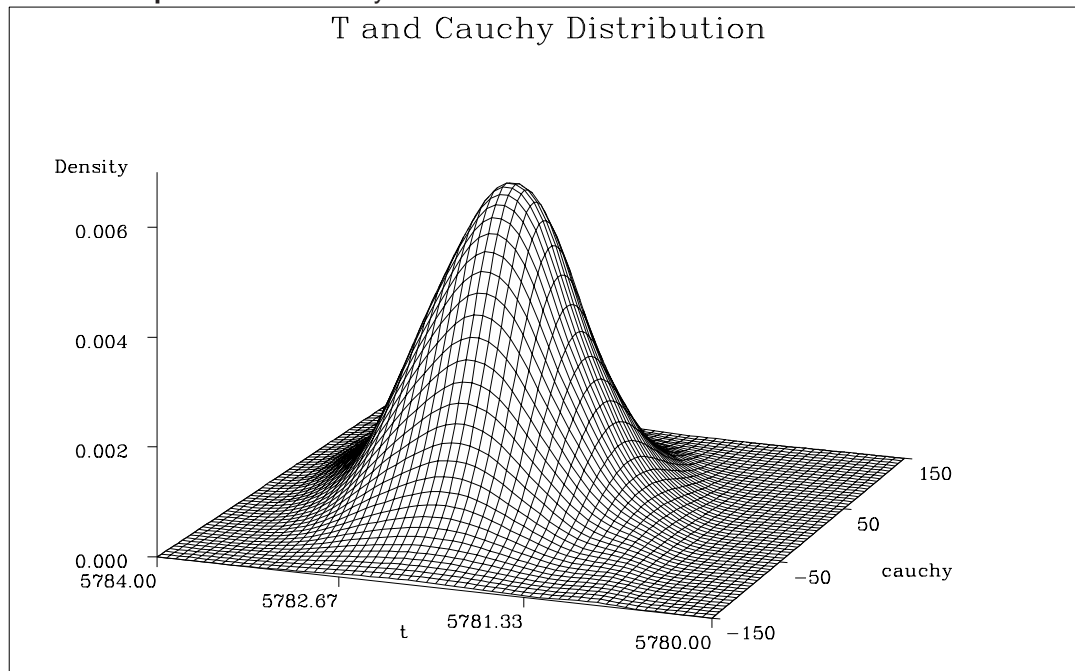
```

proc kde gridl=5780,-150 gridu=5784,150 data=monte out=density;
var t cauchy;
run;

title "T and Cauchy Distribution";
proc g3d data=density;
  plot t*cauchy=density;
run;

```

Output 4.3.1. Density of T and CAUCHY Truncated in the CAUCHY dimension



References

- Goldfeld, S.M. and Quandt, R.E. (1972), *Nonlinear Methods in Econometrics*, Amsterdam: North-Holland Publishing Company.
- Goldfeld, S.M. and Quandt, R.E. (1973), "A Markov Model for Switching Regressions," *Journal of Econometrics*, 3-16.
- Goldfeld, S.M. and Quandt, R.E. (1973), "The Estimation of Structural Shifts by Switching Regressions," *Annals of Economic and Social Measurement*, 2/4.
- Goldfeld, S.M. and Quandt, R.E. (1976), *Studies in Nonlinear Estimation*, Cambridge, MA: Ballinger Publishing Company.
- Nelsen, Roger B. (1999), *Introduction to Copulas*, New York, New York: Springer-Verlag.

Chapter 5

The VARMAX Procedure

Chapter Table of Contents

OVERVIEW	93
SYNTAX	93
MODEL Statement	93
RESTRICT Statement	94
OUTEST= Data Set	94

Chapter 5

The VARMAX Procedure

Overview

The VARMAX procedure now includes maximum likelihood estimation for VARMA models and multivariate GARCH models:

Vector time series can now be estimated by a finite order VAR process with a finite order MA error term. The VARMA processes have infinite order pure VAR and MA representations. Impulse response analysis and forecasting VARMA processes are included.

Multivariate GARCH models now enable you to specify a functional form for the conditional covariance and also enable you to model the first and second moments jointly.

Syntax

MODEL Statement

Lag Specification Options

Q= *number*

Q= (*number-list*)

specifies the order of the moving-average error process. Subset models of moving-average orders can be specified as, for example, Q=(1,5). The default is Q=0.

GARCH Model Estimation Options

GARCH=(Q=number P=number FORM= value MEAN)

specifies a GARCH-type multivariate conditional heteroscedasticity model. The GARCH= option in the MODEL statement specifies the family of GARCH models to be estimated. See the “??” section on page ?? for details.

The following options can be used in the GARCH= option.

FORM= *value*

specifies the representation for a GARCH model. Valid values are as follows:

BEKK specifies a *BEKK* representation. This is the default.

BEW specifies a vectorized representation.

DIAG specifies a diagonal representation.

MEAN

specifies the GARCH-M model.

P=*number*

P=(number-list)

specifies the order of the process or the subset of GARCH terms to be fitted.
By default, P=0.

Q=number

Q=(number-list)

specifies the order of the process or the subset of ARCH terms to be fitted. This option is required in the GARCH= option.

Printing Options

PRINT=(ROOTS)

In addition, when the GARCH= option is specified, this option prints the roots of the GARCH characteristic polynomials to check covariance stationarity for the GARCH process.

RESTRICT Statement

The following keyword is for the fitted moving-average model:

$MA(l, i, j)$ is the moving-average parameter of the previous lag l value of the j th error process, $\epsilon_{j,t-l}$

The following keywords are for the fitted GARCH model. The indexes i and j refer to the position of the element in the coefficient matrix.

- $GCHM(i, j)$ is the GARCH-M parameter of $\text{vech}(H_t)$ for $i = 1, \dots, k$ and $j = 1, \dots, k(k+1)/2$
- $GCHC(i, j)$ is the constant parameter of the covariance matrix, H_t , for $1 \leq i \leq j \leq k$
- $ACH(l, i, j)$ is the ARCH parameter of the previous lag l value of $\epsilon_t \epsilon_t'$
- $GCH(l, i, j)$ is the GARCH parameter of the previous lag l value of covariance matrix, H_t

The indexes i and j for $ACH(l, i, j)$ and $GCH(l, i, j)$ have different ranges according to the GARCH model representation. For example, $i, j = 1, \dots, k$ for a *BEKK* representation, $i, j = 1, \dots, k(k+1)/2$ for a *BEW* representation, and $i = j = 1, \dots, k(k+1)/2$ for a *diagonal* representation.

OUTEST= Data Set

The following keywords are for the fitted GARCH model:

- $GCHM_i$ are numeric variables containing the estimates of the GARCH-M parameters of the covariance matrix and their standard errors, where $i = 1, \dots, k(k+1)/2$, k is the number of endogenous variables.

- $ACHl_i$ are numeric variables containing the estimates of the ARCH parameters of the covariance matrix and their standard errors, where l is the lag l th coefficient matrix and $i = 1, \dots, k$ for a *BEKK* representation, or $i = 1, \dots, k(k+1)/2$ for other representations. k is the number of endogenous variables.
- $GCHl_i$ are numeric variables containing the estimates of the GARCH parameters of the covariance matrix and their standard errors, where l is the lag l th coefficient matrix and $i = 1, \dots, k$ for a *BEKK* representation, or $i = 1, \dots, k(k+1)/2$ for other representations. k is the number of endogenous variables.
- $GCHC_i$ are numeric variables containing the estimates of the constant parameters of the covariance matrix and their standard errors, where $i = 1, \dots, k$ for a *BEKK* representation, k is the number of endogenous variables. $i = 1$ for other representations.

Chapter 6

The X12 Procedure

Chapter Table of Contents

OVERVIEW	99
SYNTAX	99
BY Statement	99
ID Statement	99
FORECAST Statement	100
TRANSFORM Statement	102
X11 Statement	104
REFERENCES	107

Chapter 6

The X12 Procedure

Overview

The X12 procedure has several new and enhanced statements, as well as some new predefined regression variables and tables:

BY and ID statements have been added. The user can control forecasting with the FORECAST statement that has the LEAD option. In the TRANSFORM statement, some standard Box-Cox transformations can be requested using the FUNCTION= options, which include the values NONE, LOG, SQRT, INVERSE, LOGISTIC, and AUTO. The X11 statement has been greatly expanded with the addition of the OUT-FORECAST, SEASONALMA=, and TRENDMA= options.

The new predefined regression variables available through the REGRESSION PREDEFINED= option are CONSTANT, LOMSTOCK, SEASONAL, TD, TD-NOLPYEAR, TD1COEF, and TD1NOPYEAR.

The following new tables are available: prior-adjustment factors (A2), chi-squared tests for groups of regressors (RegressorGroupChisq), average absolute percentage error in win-sample forecasts (AvgFcstErr), seasonal MA roots (Roots), and day of the week trading day component factors (F4).

Syntax

BY Statement

BY *variables*;

A BY statement can be used with PROC X12 to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input DATA= data set to be sorted in order of the BY variables.

ID Statement

ID *variables*;

If you are creating an output data set, use the ID statement to put values of the ID variables, in addition to the table values, into the output data set. The ID statement has no effect when an output data set is not created. If the DATE= variable is specified in the MONTHLY or QUARTERLY statement, this variable is included automatically

in the OUTPUT data set. If no DATE= variable is specified, the variable _DATE_ is added.

The date variable (or _DATE_) values outside the range of the actual data (from forecasting) are extrapolated, while all other ID variables are missing.

FORECAST Statement

FORECAST *options;*

The FORECAST statement is used to forecast the time series using the estimated model. The output contains point forecast and forecast statistics for the transformed and original series.

The following option can appear in the FORECAST statement.

LEAD= *value*

Specifies the number of periods ahead to forecast. The default is the number of periods in a year (4 or 12) and the maximum is 60.

Forecasts and Standard Errors Tables are displayed in association with the FORECAST statement. Confidence limits are also included. If the data are transformed, then two tables will be displayed, one table for the original data, and one table for the transformed data.

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. Table A6 provides information related to trading-day effects. Table A8 provides information related to outlier factors. 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= CONSTANT
PREDEFINED= LOM
PREDEFINED= LOMSTOCK
PREDEFINED= LOQ
PREDEFINED= LPYEAR
PREDEFINED= SEASONAL
PREDEFINED= TD
PREDEFINED= TDNOLPYEAR
PREDEFINED= TD1COEF
PREDEFINED= TD1NOLPYEAR

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 can be used. The syntax for using both a length-of-month and a seasonal regression could be in one of the following forms:

```
regression predefined=lom seasonal;

regression predefined=(lom seasonal);

regression predefined=lom predefined=seasonal;
```

Certain restrictions apply when using more than one predefined regression variable. Only one of TD, TDNOLPYEAR, TD1COEF, or TD1NOLPYEAR may be specified. LPYEAR cannot be used with TD, TD1COEF, LOM, LOMSTOCK, or LOQ. LOM or LOQ cannot be used with TD or TD1COEF.

Table 6.1. Predefined Regression Variables in X-12-ARIMA

Regression Effect	Variable Definitions
Trend Constant CONSTANT	$(1 - B)^{-d}(1 - B^s)^{-D}I(t \geq 1)$, where $I(t \geq 1) = \begin{cases} 1 & \text{for } t \geq 1 \\ 0 & \text{for } t < 1 \end{cases}$
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)
Stock Length-of-Month LOMSTOCK	$SLOM_t = \begin{cases} m_t - \bar{m} - \mu(l) & \text{for } t = 1 \\ SLOM_{t-1} + m_t - \bar{m} & \text{otherwise} \end{cases}$ where \bar{m} and m_t are defined in LOM and $\mu(l) = \begin{cases} 0.375 & \text{when 1st February in series is a leap year} \\ 0.125 & \text{when 2nd February in series is a leap year} \\ -0.125 & \text{when 3rd February in series is a leap year} \\ -0.375 & \text{when 4th February in series is a leap year} \end{cases}$
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	$LY_t = \begin{cases} 0.75 & \text{in leap year February (first quarter)} \\ -0.25 & \text{in other Februaries (first quarter)} \\ 0 & \text{otherwise} \end{cases}$
Fixed Seasonal SEASONAL	$M_{1,t} = \begin{cases} 1 & \text{in January} \\ -1 & \text{in December, ..., } M_{11,t} = \begin{cases} 1 & \text{in November} \\ -1 & \text{in December} \\ 0 & \text{otherwise} \end{cases} \end{cases}$
Trading Day TD, TDNOLPYEAR	$T_{1,t} = (\text{no. of Mondays}) - (\text{no. of Sundays}), \dots$ $T_{6,t} = (\text{no. of Saturdays}) - (\text{no. of Sundays})$
One Coefficient Trading Day TD1COEF, TD1NOLPYEAR	$(\text{no. of weekdays}) - \frac{5}{2}(\text{no. of Saturdays and Sundays})$

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 Prior Adjustment Factors table is associated with the TRANSFORM statement.

Only one of the following options can appear in the TRANSFORM statement.

POWER= value

Transform the input series Y_t using a Box-Cox power transformation,

$$Y_t \rightarrow y_t = \begin{cases} \log(Y_t) & \lambda = 0 \\ \lambda^2 + (Y_t^\lambda - 1)/\lambda & \lambda \neq 0 \end{cases}$$

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 `FUNCTION=` option.

Table 6.2. Power Values Related to the Census Bureau Function Argument

function=	transformation	range for Y_t	equivalent power argument
none	Y_t	all values	<i>power = 1</i>
log	$\log(Y_t)$	$Y_t > 0$ for all t	<i>power = 0</i>
sqrt	$2(\sqrt{Y_t} - 0.875)$	$Y_t \geq 0$ for all t	<i>power = 0.5</i>
inverse	$2 - \frac{1}{Y_t}$	$Y_t \neq 0$ for all t	<i>power = -1</i>
logistic	$\log(\frac{Y_t}{1-Y_t})$	$0 < Y_t < 1$ for all t	<i>none equivalent</i>

FUNCTION=NONE

FUNCTION=LOG

FUNCTION=SQRT

FUNCTION=INVERSE

FUNCTION=LOGISTIC

FUNCTION=AUTO

The transformation used by `FUNCTION=NONE`, `LOG`, `SQRT`, `INVERSE`, and `LOGISTIC` is related to the `POWER=` option as shown in Table 6.2. `FUNCTION=AUTO` uses an AIC-based selection to decide between a log transformation and no transformation. The default is `FUNCTION=NONE`.

However, the `FUNCTION=` and `POWER=` options are not completely equivalent. In some cases, using the `FUNCTION=` option causes the program to automatically select other options. For instance, `FUNCTION=NONE` causes the default mode to be `MODE=ADD` in the `X11` statement. Also, the choice of transformation invoked by the `FUNCTION=AUTO` option may impact the default mode of the `X11` statement.

Note that there are restrictions on the value used in the `POWER` and `FUNCTION` options 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. The preceding restrictions also apply to `FUNCTION=NONE` and `FUNCTION=LOG`.

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 in Table 6.3.

Table 6.3. 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	<i>F</i> 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
F4	day of the week trading day component factors
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 6.4 shows the values of the MODE= option and the corresponding models for the original (O) and the seasonally adjusted (SA) series.

Table 6.4. 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	$Log(O) = C + S + I$	$SA = exp(C + I)$

OUTFCST

OUTFORECAST

determines if forecasts will be included in certain tables sent to the output data set. If OUTFORECAST is specified, then forecasted values will be included in the output data set for tables A6, A8, A16, B1, D10, and D16. The default is not to include forecasts.

SEASONALMA=S3X1

SEASONALMA=S3X3

SEASONALMA=S3X5

SEASONALMA=S3X9

SEASONALMA=S3X15

SEASONALMA=STABLE

SEASONALMA=X11DEFAULT

SEASONALMA=MSR

specifies which seasonal moving average (also called seasonal “filter”) will be used to estimate the seasonal factors. These seasonal moving averages are $n \times m$ moving averages, meaning that an n -term simple average is taken of a sequence of consecutive m -term simple averages. X11DEFAULT is the method used by the U.S. Census Bureau’s X-11-ARIMA program. The default for PROC X12 is SEASONALMA=MSR, which is the methodology of Statistic Canada’s X-11-ARIMA/88 program.

The following seasonal filters can be selected for the entire series:

Table 6.5. X-12-ARIMA Seasonal Filter Options and Descriptions

Filter Name	Description of Filter
S3X1	A 3×1 moving average.
S3X3	A 3×3 moving average.
S3X5	A 3×5 moving average.
S3X9	A 3×9 moving average.
S3X15	A 3×15 moving average.
STABLE	Stable seasonal filter. As single seasonal factor for each calendar month or quarter is generated by calculating the simple average of all the values for each month or quarter (taken after detrending and outlier adjustment).
X11DEFAULT	A 3×3 moving average is used to calculate the initial seasonal factors in each iteration, and a 3×5 moving average to calculate the final seasonal factors.
MSR	Choose filter automatically using moving seasonality ratio of X-11-ARIMA/88 (Dagum 1988).

TRENDMA= *value*

specifies which Henderson moving average will be used to estimate the final trend-cycle. Any odd number greater than one and less than or equal to 101 can be specified. Example: trendma=23. If no selection is made the program will select a trend moving average based on statistical characteristics of the data. For monthly series, either a 9-, 13-, or 23-term Henderson moving average will be selected. For quarterly series, the program will choose either a 5- or a 7-term Henderson moving average.

References

- Ladiray, D. and Quenneville B. (2001), "Seasonal Adjustment with the X-11 Method," New York: Springer-Verlag.
- 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.

Subject Index

@

- @CRSPDB Date Informats
 - SASECRSP engine, 15
- @CRSPDR Date Informats
 - SASECRSP engine, 15
- @CRSPDT Date Informats
 - SASECRSP engine, 15

B

- BY groups
 - X12 procedure, 99

C

- Cauchy distribution estimation
 - example, 80
- CONTENTS procedure
 - SASECRSP engine, 6
- creating a FAME view
 - See SASEFAME engine
- CRSP and SAS Dates
 - SASECRSP engine, 14
- CRSP Date Formats
 - SASECRSP engine, 14
- CRSP Date Functions
 - SASECRSP engine, 14
- CRSP Date Informats
 - SASECRSP engine, 15
- CRSP Integer Date Format
 - SASECRSP engine, 14
- CRSPDB_SASCAL environment variable
 - SASECRSP engine, 5
- CRSPDCI Date Functions
 - SASECRSP engine, 16
- CRSPDCS Date Functions
 - SASECRSP engine, 16
- CRSPDI2S Date Function
 - SASECRSP engine, 16
- CRSPDIC Date Functions
 - SASECRSP engine, 16
- CRSPDS2I Date Function
 - SASECRSP engine, 16
- CRSPDSC Date Functions
 - SASECRSP engine, 16
- CRSPDT Date Formats
 - SASECRSP engine, 14

D

- DRI data files in FAME.db
 - See SASEFAME engine

- DRI/McGraw-Hill data files in FAME.db
 - See SASEFAME engine

E

- Environment variable, CRSPDB_SASCAL
 - SASECRSP engine, 5
- example
 - Cauchy distribution estimation, 80
 - Multivariate Mixture of Distributions, 86
 - Simulating from a Mixture of Distributions, 86
 - Switching Regression, 82
 - The D-method, 82

F

- FAME data files
 - See SASEFAME engine
- FAME Information Services Databases
 - See SASEFAME engine

G

- Gaussian distribution
 - MODEL procedure, 70

I

- ID variables
 - X12 procedure, 99

L

- LIBNAME interface engine for FAME database
 - See SASEFAME engine
- libname statement
 - SASECRSP engine, 5
- LIBNAME statement
 - SASEFAME engine, 31

M

- main economic indicators (OECD) data files in FAME.db
 - See SASEFAME engine
- MODEL procedure
 - Gaussian distribution, 70
 - multivariate t -distribution, 72
 - normal distribution, 70
- multivariate t -distribution
 - MODEL procedure, 72
- Multivariate Mixture of Distributions
 - example, 86

N

national accounts data files (OECD) in FAME.db
 See SASEFAME engine
 negative log-likelihood function, 72
 Nonlinear time series analysis
 VARMAX procedure, 93
 normal distribution
 MODEL procedure, 70

O

OECD data files in FAME.db
 See SASEFAME engine
 Organization for Economic Cooperation and Development data files in FAME.db
 See SASEFAME engine

R

reading from CRSP data files
 SASECRSP engine, 5
 Remote FAME Access, using FAME CHLI
 SASEFAME engine, 31
 Remote FAME Access, using SAS CONNECT
 SASEFAME engine, 31

S

SAS and CRSP Dates
 SASECRSP engine, 14
 SAS DATA step
 SASECRSP engine, 6
 SAS Date Format
 SASECRSP engine, 14
 SAS output data set
 SASECRSP engine, 9
 SASECRSP engine
 @CRSPDB Date Informats, 15
 @CRSPDR Date Informats, 15
 @CRSPDT Date Informats, 15
 CONTENTS procedure, 6
 CRSP and SAS Dates, 14
 CRSP Date Formats, 14
 CRSP Date Functions, 14
 CRSP Date Informats, 15
 CRSP Integer Date Format, 14
 CRSPDB_SASCAL environment variable, 5
 CRSPDCI Date Functions, 16
 CRSPDCS Date Functions, 16
 CRSPDI2S Date Function, 16
 CRSPDIC Date Functions, 16
 CRSPDS2I Date Function, 16
 CRSPDSC Date Functions, 16
 CRSPDT Date Formats, 14
 Environment variable, CRSPDB_SASCAL, 5
 libname statement, 5
 reading from CRSP data files, 5
 SAS and CRSP Dates, 14
 SAS DATA step, 6
 SAS Date Format, 14
 SAS output data set, 9
 setid option, 5

SQL procedure, creating a view, 6
 SQL procedure, using clause, 6

SASEFAME engine

creating a FAME view, 31
 DRI data files in FAME.db, 31
 DRI/McGraw-Hill data files in FAME.db, 31
 FAME data files, 31
 FAME Information Services Databases, 31
 LIBNAME interface engine for FAME databases, 31
 LIBNAME statement, 31
 main economic indicators (OECD) data files in FAME.db, 31
 national accounts data files (OECD) in FAME.db, 31
 OECD data files in FAME.db, 31
 Organization for Economic Cooperation and Development data files in FAME.db, 31
 Remote FAME Access, using FAME CHLI, 31
 Remote FAME Access, using SAS CONNECT, 31
 viewing a FAME database, 31

setid option

SASECRSP engine, 5

Simulating from a Mixture of Distributions
 example, 86

SQL procedure, creating a view

SASECRSP engine, 6

SQL procedure, using clause

SASECRSP engine, 6

Switching Regression

example, 82

T

The D-method
 example, 82

V

VARMAX procedure

Nonlinear time series analysis, 93

viewing a FAME database

See SASEFAME engine

X

X12 procedure

BY groups, 99

ID variables, 99

Syntax Index

A

ALPHA= option
OUTLIER statement (ARIMA), 55

B

BY statement
X12 procedure, 99

C

CAUCHY option
ERRORMODEL statement (MODEL), 70
CDF= option
ERRORMODEL statement (MODEL), 71
CHISQUARED option
ERRORMODEL statement (MODEL), 70

E

EMPIRICAL= option
ERRORMODEL statement (MODEL), 71

F

F option
ERRORMODEL statement (MODEL), 70
FORECAST statement
X12 procedure, 100
FUNCTION= option
TRANSFORM statement (X12), 103

G

GARCH option
MODEL statement, 93
GENERAL= option
ERRORMODEL statement (MODEL), 70

I

ID statement
X12 procedure, 99
IDENTIFY statement
ARIMA procedure, 55
INSET= option
LIBNAME statement (SASECRSP), 8

L

LEAD= option
FORECAST statement (X12), 100
LIBNAME *libref* SASEFAME statement, 32
LIBNAME *libref* SASECRSP statement, 7

M

MAXNUM= option
OUTLIER statement (ARIMA), 56
MAXPCT= option
OUTLIER statement (ARIMA), 56
MODE= option
X11 statement (X12), 105

N

NORMAL option
ERRORMODEL statement (MODEL), 70

O

OUTFORECAST option
X11 statement (X12), 105

P

PERMNO= option
LIBNAME statement (SASECRSP), 7
POISSON option
ERRORMODEL statement (MODEL), 70
POWER= option
TRANSFORM statement (X12), 102
PREDEFINED= option
PREDEFINED statement (X12), 101

Q

Q= option
MODEL statement (VARMAX), 93

R

RANGE= option
LIBNAME statement (SASECRSP), 7
REGRESSION statement
X12 procedure, 100
Remote FAME data access
physical name using #port number, 32

S

SEASONALMA= option
X11 statement (X12), 105
SETID= option
LIBNAME statement (SASECRSP), 7
SIGMA= option
OUTLIER statement (ARIMA), 55

T

T option

 ERRORMODEL statement (MODEL), 70

t option

 ERRORMODEL statement (MODEL), 72

TRANSFORM statement

 X12 procedure, 102

TRENDMA= option

 X11 statement (X12), 106

TYPE= option

 OUTLIER statement (ARIMA), 55

U

U option

 ERRORMODEL statement (MODEL), 70

X

X11 statement

 X12 procedure, 104

Your Turn

If you have comments or suggestions about *SAS/ETS® Software: Changes and Enhancements, Release 8.2*, 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 Inc.
SAS Publishing
SAS Campus Drive
Cary, NC 27513
E-mail: yourturn@sas.com

For suggestions about the software, please return the photocopy to

SAS Institute Inc.
Technical Support Division
SAS Campus Drive
Cary, NC 27513
E-mail: 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/selecttext, or contact our SelecText coordinators by sending e-mail to selecttext@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.