Chapter 7

The ARIMA Procedure

Chapter Table of Contents

Chapter 7

The ARIMA Procedure

Overview

The ARIMA procedure analyzes and forecasts equally spaced univariate time series data, transfer function data, and intervention data using the **A**uto**R**egressive **I**ntegrated **M**oving-**A**verage (ARIMA) or autoregressive moving-average (ARMA) model. An ARIMA model predicts a value in a response time series as a linear combination of its own past values, past errors (also called shocks or innovations), and current and past values of other time series.

The ARIMA approach was first popularized by Box and Jenkins, and ARIMA models are often referred to as Box-Jenkins models. The general transfer function model employed by the ARIMA procedure was discussed by Box and Tiao (1975). When an ARIMA model includes other time series as input variables, the model is sometimes referred to as an ARIMAX model. Pankratz (1991) refers to the ARIMAX model as *dynamic regression*.

The ARIMA procedure provides a comprehensive set of tools for univariate time series model identification, parameter estimation, and forecasting, and it offers great flexibility in the kinds of ARIMA or ARIMAX models that can be analyzed. The ARIMA procedure supports seasonal, subset, and factored ARIMA models; intervention or interrupted time series models; multiple regression analysis with ARMA errors; and rational transfer function models of any complexity.

The design of PROC ARIMA closely follows the Box-Jenkins strategy for time series modeling with features for the identification, estimation and diagnostic checking, and forecasting steps of the Box-Jenkins method.

Before using PROC ARIMA, you should be familiar with Box-Jenkins methods, and you should exercise care and judgment when using the ARIMA procedure. The ARIMA class of time series models is complex and powerful, and some degree of expertise is needed to use them correctly.

If you are unfamiliar with the principles of ARIMA modeling, refer to textbooks on time series analysis. Also refer to *SAS/ETS Software: Applications Guide 1, Version 6, First Edition*. You might consider attending the SAS Training Course "Forecasting Techniques Using SAS/ETS Software." This course provides in-depth training on ARIMA modeling using PROC ARIMA, as well as training on the use of other forecasting tools available in SAS/ETS software.

Getting Started

This section outlines the use of the ARIMA procedure and gives a cursory description of the ARIMA modeling process for readers less familiar with these methods.

The Three Stages of ARIMA Modeling

The analysis performed by PROC ARIMA is divided into three stages, corresponding to the stages described by Box and Jenkins (1976). The IDENTIFY, ESTIMATE, and FORECAST statements perform these three stages, which are summarized below.

1. In the *identification* stage, you use the IDENTIFY statement to specify the response series and identify candidate ARIMA models for it. The IDENTIFY statement reads time series that are to be used in later statements, possibly differencing them, and computes autocorrelations, inverse autocorrelations, partial autocorrelations, and cross correlations. Stationarity tests can be performed to determine if differencing is necessary. The analysis of the IDENTIFY statement output usually suggests one or more ARIMA models that could be fit. Options allow you to test for stationarity and tentative ARMA order identification.

2. In the *estimation and diagnostic checking* stage, you use the ESTIMATE statement to specify the ARIMA model to fit to the variable specified in the previous IDENTIFY statement, and to estimate the parameters of that model. The ESTIMATE statement also produces diagnostic statistics to help you judge the adequacy of the model.

   Significance tests for parameter estimates indicate whether some terms in the model may be unnecessary. Goodness-of-fit statistics aid in comparing this model to others. Tests for white noise residuals indicate whether the residual series contains additional information that might be utilized by a more complex model. If the diagnostic tests indicate problems with the model, you try another model, then repeat the estimation and diagnostic checking stage.

3. In the *forecasting* stage you use the FORECAST statement to forecast future values of the time series and to generate confidence intervals for these forecasts from the ARIMA model produced by the preceding ESTIMATE statement.

These three steps are explained further and illustrated through an extended example in the following sections.

Identification Stage

Suppose you have a variable called SALES that you want to forecast. The following example illustrates ARIMA modeling and forecasting using a simulated data set TEST containing a time series SALES generated by an ARIMA(1,1,1) model. The output produced by this example is explained in the following sections. The simulated SALES series is shown in Figure 7.1.

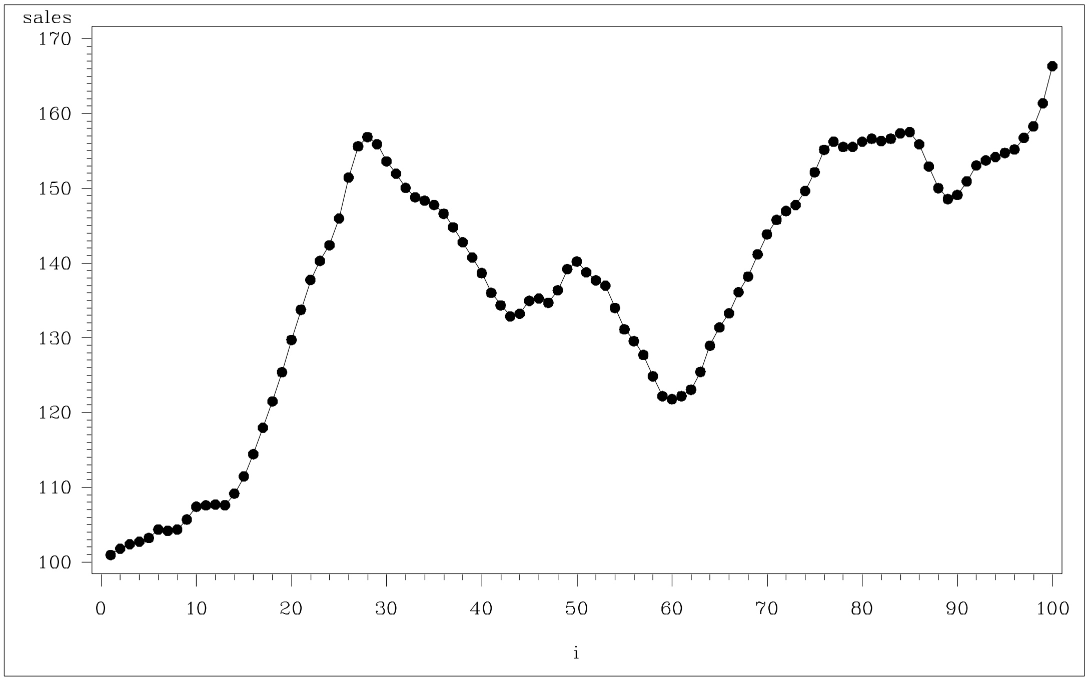


Figure 7.1. Simulated ARIMA(1,1,1) Series SALES

Using the IDENTIFY Statement

You first specify the input data set in the PROC ARIMA statement. Then, you use an IDENTIFY statement to read in the SALES series and plot its autocorrelation function. You do this using the following statements:

```
proc arima data=test;
   identify var=sales nlag=8;
   run;
```

Descriptive Statistics

The IDENTIFY statement first prints descriptive statistics for the SALES series. This part of the IDENTIFY statement output is shown in Figure 7.2.

```
                              The ARIMA Procedure

                            Name of Variable = sales

                       Mean of Working Series    137.3662
                       Standard Deviation        17.36385
                       Number of Observations         100
```

Figure 7.2. IDENTIFY Statement Descriptive Statistics Output

Autocorrelation Function Plots

The IDENTIFY statement next prints three plots of the correlations of the series with its past values at different lags. These are the

- sample autocorrelation function plot

- sample partial autocorrelation function plot
- sample inverse autocorrelation function plot

The sample autocorrelation function plot output of the IDENTIFY statement is shown in Figure 7.3.

```
                              The ARIMA Procedure

                                Autocorrelations

Lag   Covariance   Correlation   -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

  0      301.503       1.00000   |                    |********************|
  1      288.454       0.95672   |                .   |******************* |
  2      273.437       0.90691   |             .      |******************  |
  3      256.787       0.85169   |            .       |*****************   |
  4      238.518       0.79110   |          .         |****************    |
  5      219.033       0.72647   |         .          |***************     |
  6      198.617       0.65876   |         .          |*************       |
  7      177.150       0.58755   |        .           |************        |
  8      154.914       0.51381   |       .            |**********  .       |

                         "." marks two standard errors
```

Figure 7.3. IDENTIFY Statement Autocorrelations Plot

The autocorrelation plot shows how values of the series are correlated with past values of the series. For example, the value 0.95672 in the "Correlation" column for the Lag 1 row of the plot means that the correlation between SALES and the SALES value for the previous period is .95672. The rows of asterisks show the correlation values graphically.

These plots are called autocorrelation functions because they show the degree of correlation with past values of the series as a function of the number of periods in the past (that is, the lag) at which the correlation is computed.

The NLAG= option controls the number of lags for which autocorrelations are shown. By default, the autocorrelation functions are plotted to lag 24; in this example the NLAG=8 option is used, so only the first 8 lags are shown.

Most books on time series analysis explain how to interpret autocorrelation plots and partial autocorrelation plots. See the section "The Inverse Autocorrelation Function" later in this chapter for a discussion of inverse autocorrelation plots.

By examining these plots, you can judge whether the series is *stationary* or *nonstationary*. In this case, a visual inspection of the autocorrelation function plot indicates that the SALES series is nonstationary, since the ACF decays very slowly. For more formal stationarity tests, use the STATIONARITY= option. (See the section "Stationarity" later in this chapter.)

The inverse and partial autocorrelation plots are printed after the autocorrelation plot. These plots have the same form as the autocorrelation plots, but display inverse and partial autocorrelation values instead of autocorrelations and autocovariances. The partial and inverse autocorrelation plots are not shown in this example.

White Noise Test

The last part of the default IDENTIFY statement output is the check for white noise. This is an approximate statistical test of the hypothesis that none of the autocorrelations of the series up to a given lag are significantly different from 0. If this is true for all lags, then there is no information in the series to model, and no ARIMA model is needed for the series.

The autocorrelations are checked in groups of 6, and the number of lags checked depends on the NLAG= option. The check for white noise output is shown in Figure 7.4.

```
                                The ARIMA Procedure

                       Autocorrelation Check for White Noise

    To      Chi-         Pr >
   Lag    Square   DF   ChiSq  ---------------Autocorrelations---------------

     6    426.44    6  <.0001   0.957   0.907   0.852   0.791   0.726   0.659
```

Figure 7.4. IDENTIFY Statement Check for White Noise

In this case, the white noise hypothesis is rejected very strongly, which is expected since the series is nonstationary. The *p* value for the test of the first six autocorrelations is printed as <0.0001, which means the *p* value is less than .0001.

Identification of the Differenced Series

Since the series is nonstationary, the next step is to transform it to a stationary series by differencing. That is, instead of modeling the SALES series itself, you model the change in SALES from one period to the next. To difference the SALES series, use another IDENTIFY statement and specify that the first difference of SALES be analyzed, as shown in the following statements:

```
identify var=sales(1) nlag=8;
run;
```

The second IDENTIFY statement produces the same information as the first but for the change in SALES from one period to the next rather than for the total sales in each period. The summary statistics output from this IDENTIFY statement is shown in Figure 7.5. Note that the period of differencing is given as 1, and one observation was lost through the differencing operation.

```
                                The ARIMA Procedure

                             Name of Variable = sales

             Period(s) of Differencing                        1
             Mean of Working Series                    0.660589
             Standard Deviation                        2.011543
             Number of Observations                          99
             Observation(s) eliminated by differencing        1
```

Figure 7.5. IDENTIFY Statement Output for Differenced Series

The autocorrelation plot for the differenced series is shown in Figure 7.6.

```
                              The ARIMA Procedure

                                Autocorrelations

Lag   Covariance   Correlation   -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

  0     4.046306       1.00000   |                    |********************|
  1     3.351258       0.82823   |                .   |*****************   |
  2     2.390895       0.59088   |              .     |************        |
  3     1.838925       0.45447   |             .      |*********           |
  4     1.494253       0.36929   |            .       |*******.            |
  5     1.135753       0.28069   |            .       |****** .            |
  6     0.801319       0.19804   |            .       |****   .            |
  7     0.610543       0.15089   |            .       |***    .            |
  8     0.326495       0.08069   |            .       |**     .            |

                         "." marks two standard errors
```

Figure 7.6. Autocorrelations Plot for Change in SALES

The autocorrelations decrease rapidly in this plot, indicating that the change in SALES is a stationary time series.

The next step in the Box-Jenkins methodology is to examine the patterns in the autocorrelation plot to choose candidate ARMA models to the series. The partial and inverse autocorrelation function plots are also useful aids in identifying appropriate ARMA models for the series. The partial and inverse autocorrelation function plots are shown in Figure 7.7 and Figure 7.8.

```
                              The ARIMA Procedure

                            Inverse Autocorrelations

       Lag    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

         1       -0.73867    |     ***************|   .                |
         2        0.36801    |                .   |*******             |
         3       -0.17538    |                ****|   .                |
         4        0.11431    |                .   |** .                |
         5       -0.15561    |                .***|   .                |
         6        0.18899    |                .   |****                |
         7       -0.15342    |                .***|   .                |
         8        0.05952    |                .   |*  .                |
```

Figure 7.7. Inverse Autocorrelation Function Plot for Change in SALES

```
                              The ARIMA Procedure

                            Partial Autocorrelations

Lag    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

  1        0.82823    |               .   |*****************  |
  2       -0.30275    |             ******|   .               |
  3        0.23722    |               .   |*****              |
  4       -0.07450    |               .  *|   .               |
  5       -0.02654    |               .  *|   .               |
  6       -0.01012    |               .   |   .               |
  7        0.04189    |               .   |*  .               |
  8       -0.17668    |               ****|   .               |
```

Figure 7.8. Partial Autocorrelation Plot for Change in SALES

In the usual Box and Jenkins approach to ARIMA modeling, the sample autocorrelation function, inverse autocorrelation function, and partial autocorrelation function are compared with the theoretical correlation functions expected from different kinds of ARMA models. This matching of theoretical autocorrelation functions of different ARMA models to the sample autocorrelation functions computed from the response series is the heart of the identification stage of Box-Jenkins modeling. Most textbooks on time series analysis discuss the theoretical autocorrelation functions for different kinds of ARMA models.

Since the input data is only a limited sample of the series, the sample autocorrelation functions computed from the input series will only approximate the true autocorrelation functions of the process generating the series. This means that the sample autocorrelation functions will not exactly match the theoretical autocorrelation functions for any ARMA model and may have a pattern similar to that of several different ARMA models.

If the series is white noise (a purely random process), then there is no need to fit a model. The check for white noise, shown in Figure 7.9, indicates that the change in sales is highly autocorrelated. Thus, an autocorrelation model, for example an AR(1) model, might be a good candidate model to fit to this process.

```
                              The ARIMA Procedure

                      Autocorrelation Check for White Noise

 To        Chi-          Pr >
Lag      Square    DF    ChiSq    ---------------Autocorrelations---------------

  6      154.44     6   <.0001     0.828    0.591    0.454    0.369    0.281    0.198
```

Figure 7.9. IDENTIFY Statement Check for White Noise

Estimation and Diagnostic Checking Stage

The autocorrelation plots for this series, as shown in the previous section, suggest an AR(1) model for the change in SALES. You should check the diagnostic statistics to see if the AR(1) model is adequate. Other candidate models include an MA(1) model, and low-order mixed ARMA models. In this example, the AR(1) model is tried first.

Estimating an AR(1) Model

The following statements fit an AR(1) model (an autoregressive model of order 1), which predicts the change in sales as an average change, plus some fraction of the previous change, plus a random error. To estimate an AR model, you specify the order of the autoregressive model with the P= option on an ESTIMATE statement, as shown in the following statements:

```
   estimate p=1;
   run;
```

The ESTIMATE statement fits the model to the data and prints parameter estimates and various diagnostic statistics that indicate how well the model fits the data. The first part of the ESTIMATE statement output, the table of parameter estimates, is shown in Figure 7.10.

```
                              The ARIMA Procedure

                     Conditional Least Squares Estimation

                                Approx Std
   Parameter      Estimate           Error       t Value    Pr > |t|     Lag

   MU              0.90280         0.65984          1.37      0.1744       0
   AR1,1           0.86847         0.05485         15.83      <.0001       1
```

Figure 7.10. Parameter Estimates for AR(1) Model

The table of parameter estimates is titled "Condidtional Least Squares Estimation," which indicates the estimation method used. You can request different estimation methods with the METHOD= option.

The table of parameter estimates lists the parameters in the model; for each parameter, the table shows the estimated value and the standard error and *t* value for the estimate. The table also indicates the lag at which the parameter appears in the model.

In this case, there are two parameters in the model. The mean term is labeled MU; its estimated value is .90280. The autoregressive parameter is labeled AR1,1; this is the coefficient of the lagged value of the change in SALES, and its estimate is .86847.

The *t* values provide significance tests for the parameter estimates and indicate whether some terms in the model may be unnecessary. In this case, the *t* value for the autoregressive parameter is 15.83, so this term is highly significant. The *t* value for MU indicates that the mean term adds little to the model.

The standard error estimates are based on large sample theory. Thus, the standard errors are labeled as approximate, and the standard errors and *t* values may not be reliable in small samples.

The next part of the ESTIMATE statement output is a table of goodness-of-fit statistics, which aid in comparing this model to other models. This output is shown in Figure 7.11.

```
                              The ARIMA Procedure

                        Constant Estimate      0.118749
                        Variance Estimate       1.15794
                        Std Error Estimate     1.076076
                        AIC                    297.4469
                        SBC                    302.6372
                        Number of Residuals          99
                 * AIC and SBC do not include log determinant.
```

Figure 7.11. Goodness-of-Fit Statistics for AR(1) Model

The "Constant Estimate" is a function of the mean term MU and the autoregressive parameters. This estimate is computed only for AR or ARMA models, but not for strictly MA models. See the section "General Notation for ARIMA Models" later in this chapter for an explanation of the constant estimate.

The "Variance Estimate" is the variance of the residual series, which estimates the innovation variance. The item labeled "Std Error Estimate" is the square root of the variance estimate. In general, when comparing candidate models, smaller AIC and SBC statistics indicate the better fitting model. The section "Estimation Details" later in this chapter explains the AIC and SBC statistics.

The ESTIMATE statement next prints a table of correlations of the parameter estimates, as shown in Figure 7.12. This table can help you assess the extent to which collinearity may have influenced the results. If two parameter estimates are very highly correlated, you might consider dropping one of them from the model.

```
                              The ARIMA Procedure

                           Correlations of Parameter
                                   Estimates

                         Parameter        MU     AR1,1

                         MU            1.000     0.114
                         AR1,1         0.114     1.000
```

Figure 7.12. Correlations of the Estimates for AR(1) Model

The next part of the ESTIMATE statement output is a check of the autocorrelations of the residuals. This output has the same form as the autocorrelation check for white noise that the IDENTIFY statement prints for the response series. The autocorrelation check of residuals is shown in Figure 7.13.

```
                              The ARIMA Procedure

                      Autocorrelation Check of Residuals

  To      Chi-         Pr >
 Lag    Square   DF   ChiSq  ---------------Autocorrelations---------------

   6     19.09    5  0.0019   0.327  -0.220  -0.128   0.068  -0.002  -0.096
  12     22.90   11  0.0183   0.072   0.116  -0.042  -0.066   0.031  -0.091
  18     31.63   17  0.0167  -0.233  -0.129  -0.024   0.056  -0.014  -0.008
  24     32.83   23  0.0841   0.009  -0.057  -0.057  -0.001   0.049  -0.015
```

Figure 7.13. Check for White Noise Residuals for AR(1) Model

The χ^2 test statistics for the residuals series indicate whether the residuals are uncorrelated (white noise) or contain additional information that might be utilized by a more complex model. In this case, the test statistics reject the no-autocorrelation hypothesis at a high level of significance. (*p*=0.0019 for the first six lags.) This means that the residuals are not white noise, and so the AR(1) model is not a fully adequate model for this series.

The final part of the ESTIMATE statement output is a listing of the estimated model using the back shift notation. This output is shown in Figure 7.14.

```
                              The ARIMA Procedure

                            Model for variable sales

                     Estimated Mean               0.902799
                     Period(s) of Differencing           1


                             Autoregressive Factors

                          Factor 1:  1 - 0.86847 B**(1)
```

Figure 7.14. Estimated ARIMA(1,1,0) Model for SALES

This listing combines the differencing specification given in the IDENTIFY statement with the parameter estimates of the model for the change in sales. Since the AR(1) model is for the change in sales, the final model for sales is an ARIMA(1,1,0) model. Using B, the back shift operator, the mathematical form of the estimated model shown in this output is as follows:

$$(1-B)sales_t = 0.902799 + \frac{1}{(1-0.86847B)}a_t$$

See the section "General Notation for ARIMA Model" later in this chapter for further explanation of this notation.

Estimating an ARMA(1,1) Model

The IDENTIFY statement plots suggest a mixed autoregressive and moving average model, and the previous ESTIMATE statement check of residuals indicates that an AR(1) model is not sufficient. You now try estimating an ARMA(1,1) model for the change in SALES.

An ARMA(1,1) model predicts the change in SALES as an average change, plus some fraction of the previous change, plus a random error, plus some fraction of the random error in the preceding period. An ARMA(1,1) model for the change in sales is the same as an ARIMA(1,1,1) model for the level of sales.

To estimate a mixed autoregressive moving average model, you specify the order of the moving average part of the model with the Q= option on an ESTIMATE statement in addition to specifying the order of the autoregressive part with the P= option. The following statements fit an ARMA(1,1) model to the differenced SALES series:

```
estimate p=1 q=1;
run;
```

The parameter estimates table and goodness-of-fit statistics for this model are shown in Figure 7.15.

```
                              The ARIMA Procedure

                     Conditional Least Squares Estimation

                               Approx Std
Parameter      Estimate           Error       t Value    Pr > |t|     Lag

MU              0.89288         0.49391          1.81      0.0738       0
MA1,1          -0.58935         0.08988         -6.56      <.0001       1
AR1,1           0.74755         0.07785          9.60      <.0001       1


                    Constant Estimate      0.225409
                    Variance Estimate      0.904034
                    Std Error Estimate     0.950807
                    AIC                    273.9155
                    SBC                    281.7009
                    Number of Residuals          99
                * AIC and SBC do not include log determinant.
```

Figure 7.15. Estimated ARMA(1,1) Model for Change in SALES

The moving average parameter estimate, labeled "MA1,1", is -0.58935. Both the moving average and the autoregressive parameters have significant *t* values. Note that the variance estimate, AIC, and SBC are all smaller than they were for the AR(1) model, indicating that the ARMA(1,1) model fits the data better without overparameterizing.

The check for white noise residuals is shown in Figure 7.16. The χ^2 tests show that we cannot reject the hypothesis that the residuals are uncorrelated. Thus, you conclude that the ARMA(1,1) model is adequate for the change in sales series, and there is no point in trying more complex models.

```
                              The ARIMA Procedure

                       Autocorrelation Check of Residuals

   To      Chi-         Pr >
  Lag    Square   DF   ChiSq  ---------------Autocorrelations---------------

    6      3.95    4  0.4127   0.016  -0.044  -0.068   0.145   0.024  -0.094
   12      7.03   10  0.7227   0.088   0.087  -0.037  -0.075   0.051  -0.053
   18     15.41   16  0.4951  -0.221  -0.033  -0.092   0.086  -0.074  -0.005
   24     16.96   22  0.7657   0.011  -0.066  -0.022  -0.032   0.062  -0.047
```

Figure 7.16. Check for White Noise Residuals for ARMA(1,1) Model

The output showing the form of the estimated ARIMA(1,1,1) model for SALES is shown in Figure 7.17.

```
                              The ARIMA Procedure

                           Model for variable sales

                     Estimated Mean               0.892875
                     Period(s) of Differencing           1


                            Autoregressive Factors

                         Factor 1:  1 - 0.74755 B**(1)


                            Moving Average Factors

                         Factor 1:  1 + 0.58935 B**(1)
```

Figure 7.17. Estimated ARIMA(1,1,1) Model for SALES

The estimated model shown in this output is

$$(1-B)sales_t = 0.892875 + \frac{(1+0.58935B)}{(1-0.74755B)}a_t$$

Since the model diagnostic tests show that all the parameter estimates are significant and the residual series is white noise, the estimation and diagnostic checking stage is complete. You can now proceed to forecasting the SALES series with this ARIMA(1,1,1) model.

Forecasting Stage

To produce the forecast, use a FORECAST statement after the ESTIMATE statement for the model you decide is best. If the last model fit were not the best, then repeat the ESTIMATE statement for the best model before using the FORECAST statement.

Suppose that the SALES series is monthly, that you wish to forecast one year ahead from the most recently available sales figure, and that the dates for the observations are given by a variable DATE in the input data set TEST. You use the following FORECAST statement:

```
forecast lead=12 interval=month id=date out=results;
run;
```

The LEAD= option specifies how many periods ahead to forecast (12 months, in this case). The ID= option specifies the ID variable used to date the observations of the SALES time series. The INTERVAL= option indicates that data are monthly and enables PROC ARIMA to extrapolate DATE values for forecast periods. The OUT= option writes the forecasts to an output data set RESULTS. See the section "OUT= Data Set" later in this chapter for information on the contents of the output data set.

By default, the FORECAST statement also prints the forecast values, as shown in Figure 7.18. This output shows for each forecast period the observation number, forecast value, standard error estimate for the forecast value, and lower and upper limits for a 95% confidence interval for the forecast.

```
                              The ARIMA Procedure

                          Forecasts for variable sales

        Obs       Forecast    Std Error       95% Confidence Limits

        101       171.0320       0.9508       169.1684       172.8955
        102       174.7534       2.4168       170.0165       179.4903
        103       177.7608       3.9879       169.9445       185.5770
        104       180.2343       5.5658       169.3256       191.1430
        105       182.3088       7.1033       168.3866       196.2310
        106       184.0850       8.5789       167.2707       200.8993
        107       185.6382       9.9841       166.0698       205.2066
        108       187.0247      11.3173       164.8433       209.2061
        109       188.2866      12.5807       163.6289       212.9443
        110       189.4553      13.7784       162.4501       216.4605
        111       190.5544      14.9153       161.3209       219.7879
        112       191.6014      15.9964       160.2491       222.9538
```

Figure 7.18. Estimated ARIMA(1,1,1) Model for SALES

Normally, you want the forecast values stored in an output data set, and you are not interested in seeing this printed list of the forecast. You can use the NOPRINT option on the FORECAST statement to suppress this output.

Using ARIMA Procedure Statements

The IDENTIFY, ESTIMATE, and FORECAST statements are related in a hierarchy. An IDENTIFY statement brings in a time series to be modeled; several ESTIMATE statements can follow to estimate different ARIMA models for the series; for each model estimated, several FORECAST statements can be used. Thus, a FORECAST statement must be preceded at some point by an ESTIMATE statement, and an ESTIMATE statement must be preceded at some point by an IDENTIFY statement. Additional IDENTIFY statements can be used to switch to modeling a different response series or to change the degree of differencing used.

The ARIMA procedure can be used interactively in the sense that all ARIMA procedure statements can be executed any number of times without reinvoking PROC ARIMA. You can execute ARIMA procedure statements singly or in groups by following the single statement or group of statements with a RUN statement. The output for each statement or group of statements is produced when the RUN statement is entered.

A RUN statement does not terminate the PROC ARIMA step but tells the procedure to execute the statements given so far. You can end PROC ARIMA by submitting a QUIT statement, a DATA step, another PROC step, or an ENDSAS statement.

The example in the preceding section illustrates the interactive use of ARIMA procedure statements. The complete PROC ARIMA program for that example is as follows:

```
proc arima data=test;
   identify var=sales nlag=8;
   run;
   identify var=sales(1) nlag=8;
   run;
   estimate p=1;
   run;
   estimate p=1 q=1;
   run;
   forecast lead=12 interval=month id=date out=results;
   run;
quit;
```

General Notation for ARIMA Models

ARIMA is an acronym for AutoRegressive Integrated Moving-Average. The order of an ARIMA model is usually denoted by the notation ARIMA(p,d,q), where

p	is the order of the autoregressive part
d	is the order of the differencing
q	is the order of the moving-average process

If no differencing is done (d = 0), the models are usually referred to as ARMA(p,q) models. The final model in the preceding example is an ARIMA(1,1,1) model since the IDENTIFY statement specified d = 1, and the final ESTIMATE statement specified p = 1 and q = 1.

Notation for Pure ARIMA Models

Mathematically the pure ARIMA model is written as

$$W_t = \mu + \frac{\theta(B)}{\phi(B)} a_t$$

where

t	indexes time
W_t	is the response series Y_t or a difference of the response series
μ	is the mean term
B	is the backshift operator; that is, $BX_t = X_{t-1}$
$\phi(B)$	is the autoregressive operator, represented as a polynomial in the back shift operator: $\phi(B) = 1 - \phi_1 B - \ldots - \phi_p B^p$
$\theta(B)$	is the moving-average operator, represented as a polynomial in the back shift operator: $\theta(B) = 1 - \theta_1 B - \ldots - \theta_q B^q$
a_t	is the independent disturbance, also called the random error.

The series W_t is computed by the IDENTIFY statement and is the series processed by the ESTIMATE statement. Thus, W_t is either the response series Y_t or a difference of Y_t specified by the differencing operators in the IDENTIFY statement.

For simple (nonseasonal) differencing, $W_t = (1 - B)^d Y_t$. For seasonal differencing $W_t = (1 - B)^d (1 - B^s)^D Y_t$, where *d* is the degree of nonseasonal differencing, *D* is the degree of seasonal differencing, and *s* is the length of the seasonal cycle.

For example, the mathematical form of the ARIMA(1,1,1) model estimated in the preceding example is

$$(1 - B)Y_t = \mu + \frac{(1 - \theta_1 B)}{(1 - \phi_1 B)} a_t$$

Model Constant Term

The ARIMA model can also be written as

$$\phi(B)(W_t - \mu) = \theta(B) a_t$$

or

$$\phi(B) W_t = const + \theta(B) a_t$$

where

$$const = \phi(B)\mu = \mu - \phi_1 \mu - \phi_2 \mu - \ldots - \phi_p \mu$$

Thus, when an autoregressive operator and a mean term are both included in the model, the constant term for the model can be represented as $\phi(B)\mu$. This value is printed with the label "Constant Estimate" in the ESTIMATE statement output.

Notation for Transfer Function Models

The general ARIMA model with input series, also called the ARIMAX model, is written as

$$W_t = \mu + \sum_i \frac{\omega_i(B)}{\delta_i(B)} B^{k_i} X_{i,t} + \frac{\theta(B)}{\phi(B)} a_t$$

where

$X_{i,t}$	is the *i*th input time series or a difference of the *i*th input series at time *t*
k_i	is the pure time delay for the effect of the *i*th input series
$\omega_i(B)$	is the numerator polynomial of the transfer function for the *i*th input series
$\delta_i(B)$	is the denominator polynomial of the transfer function for the *i*th input series.

The model can also be written more compactly as

$$W_t = \mu + \sum_i \Psi_i(B) X_{i,t} + n_t$$

where

$\Psi_i(B)$	is the transfer function weights for the *i*th input series modeled as a ratio of the ω and δ polynomials: $\Psi_i(B) = (\omega_i(B)/\delta_i(B))B^{k_i}$
n_t	is the noise series: $n_t = (\theta(B)/\phi(B))a_t$

This model expresses the response series as a combination of past values of the random shocks and past values of other input series. The response series is also called the *dependent series* or *output series*. An input time series is also referred to as an *independent series* or a *predictor series*. Response variable, dependent variable, independent variable, or predictor variable are other terms often used.

Notation for Factored Models

ARIMA models are sometimes expressed in a factored form. This means that the ϕ, θ, ω, or δ polynomials are expressed as products of simpler polynomials. For example, we could express the pure ARIMA model as

$$W_t = \mu + \frac{\theta_1(B)\theta_2(B)}{\phi_1(B)\phi_2(B)} a_t$$

where $\phi_1(B)\phi_2(B) = \phi(B)$ and $\theta_1(B)\theta_2(B) = \theta(B)$.

When an ARIMA model is expressed in factored form, the order of the model is usually expressed using a factored notation also. The order of an ARIMA model expressed as the product of two factors is denoted as ARIMA(p,d,q)$\times$(P,D,Q).

Notation for Seasonal Models

ARIMA models for time series with regular seasonal fluctuations often use differencing operators and autoregressive and moving average parameters at lags that are multiples of the length of the seasonal cycle. When all the terms in an ARIMA model factor refer to lags that are a multiple of a constant s, the constant is factored out and suffixed to the ARIMA(p,d,q) notation.

Thus, the general notation for the order of a seasonal ARIMA model with both seasonal and nonseasonal factors is ARIMA(p,d,q)$\times$(P,D,Q)$_s$. The term (p,d,q) gives the order of the nonseasonal part of the ARIMA model; the term (P,D,Q)$_s$ gives the order of the seasonal part. The value of s is the number of observations in a seasonal cycle: 12 for monthly series, 4 for quarterly series, 7 for daily series with day-of-week effects, and so forth.

For example, the notation ARIMA(0,1,2)$\times$(0,1,1)$_{12}$ describes a seasonal ARIMA model for monthly data with the following mathematical form:

$$(1-B)(1-B^{12})Y_t = \mu + (1-\theta_{1,1}B-\theta_{1,2}B^2)(1-\theta_{2,1}B^{12})a_t$$

Stationarity

The noise (or residual) series for an ARMA model must be *stationary*, which means that both the expected values of the series and its autocovariance function are independent of time.

The standard way to check for nonstationarity is to plot the series and its autocorrelation function. You can visually examine a graph of the series over time to see if it has a visible trend or if its variability changes noticeably over time. If the series is nonstationary, its autocorrelation function will usually decay slowly.

Another way of checking for stationarity is to use the stationarity tests described in the section “Stationarity Tests” on page 241.

Most time series are nonstationary and must be transformed to a stationary series before the ARIMA modeling process can proceed. If the series has a nonstationary variance, taking the log of the series may help. You can compute the log values in a DATA step and then analyze the log values with PROC ARIMA.

If the series has a trend over time, seasonality, or some other nonstationary pattern, the usual solution is to take the difference of the series from one period to the next and then analyze this differenced series. Sometimes a series may need to be differenced more than once or differenced at lags greater than one period. (If the trend or seasonal effects are very regular, the introduction of explanatory variables may be an appropriate alternative to differencing.)

Differencing

Differencing of the response series is specified with the VAR= option of the IDENTIFY statement by placing a list of differencing periods in parentheses after the variable name. For example, to take a simple first difference of the series SALES, use the statement

```
identify var=sales(1);
```

In this example, the change in SALES from one period to the next will be analyzed.

A deterministic seasonal pattern will also cause the series to be nonstationary, since the expected value of the series will not be the same for all time periods but will be higher or lower depending on the season. When the series has a seasonal pattern, you may want to difference the series at a lag corresponding to the length of the cycle of seasons. For example, if SALES is a monthly series, the statement

```
identify var=sales(12);
```

takes a seasonal difference of SALES, so that the series analyzed is the change in SALES from its value in the same month one year ago.

To take a second difference, add another differencing period to the list. For example, the following statement takes the second difference of SALES:

```
identify var=sales(1,1);
```

That is, SALES is differenced once at lag 1 and then differenced again, also at lag 1. The statement

```
identify var=sales(2);
```

creates a 2-span difference, that is current period sales minus sales from two periods ago. The statement

```
identify var=sales(1,12);
```

takes a second-order difference of SALES, so that the series analyzed is the difference between the current period-to-period change in SALES and the change 12 periods ago. You might want to do this if the series had both a trend over time and a seasonal pattern.

There is no limit to the order of differencing and the degree of lagging for each difference.

Differencing not only affects the series used for the IDENTIFY statement output but also applies to any following ESTIMATE and FORECAST statements. ESTIMATE statements fit ARMA models to the differenced series. FORECAST statements forecast the differences and automatically sum these differences back to undo the differencing operation specified by the IDENTIFY statement, thus producing the final forecast result.

Differencing of input series is specified by the CROSSCORR= option and works just like differencing of the response series. For example, the statement

```
   identify var=y(1) crosscorr=(x1(1) x2(1));
```

takes the first difference of Y, the first difference of X1, and the first difference of X2. Whenever X1 and X2 are used in INPUT= options in following ESTIMATE statements, these names refer to the differenced series.

Subset, Seasonal, and Factored ARMA Models

The simplest way to specify an ARMA model is to give the order of the AR and MA parts with the P= and Q= options. When you do this, the model has parameters for the AR and MA parts for all lags through the order specified. However, you can control the form of the ARIMA model exactly as shown in the following section.

Subset Models

You can control which lags have parameters by specifying the P= or Q= option as a list of lags in parentheses. A model like this that includes parameters for only some lags is sometimes called a *subset* or *additive model*. For example, consider the following two ESTIMATE statements:

```
   identify var=sales;
   estimate p=4;
   estimate p=(1 4);
```

Both specify AR(4) models, but the first has parameters for lags 1, 2, 3, and 4, while the second has parameters for lags 1 and 4, with the coefficients for lags 2 and 3 constrained to 0. The mathematical form of the autoregressive models produced by these two specifications is shown in Table 7.1.

Table 7.1. Saturated versus Subset Models

Option	**Autoregressive Operator**
P=4	$(1 - \phi_1 B - \phi_2 B^2 - \phi_3 B^3 - \phi_4 B^4)$
P=(1 4)	$(1 - \phi_1 B - \phi_4 B^4)$

Seasonal Models

One particularly useful kind of subset model is a *seasonal model*. When the response series has a seasonal pattern, the values of the series at the same time of year in previous years may be important for modeling the series. For example, if the series SALES is observed monthly, the statements

```
   identify var=sales;
   estimate p=(12);
```

model SALES as an average value plus some fraction of its deviation from this average value a year ago, plus a random error. Although this is an AR(12) model, it has only one autoregressive parameter.

Factored Models

A factored model (also referred to as a multiplicative model) represents the ARIMA model as a product of simpler ARIMA models. For example, you might model SALES as a combination of an AR(1) process reflecting short term dependencies and an AR(12) model reflecting the seasonal pattern.

It might seem that the way to do this is with the option P=(1 12), but the AR(1) process also operates in past years; you really need autoregressive parameters at lags 1, 12, and 13. You can specify a subset model with separate parameters at these lags, or you can specify a factored model that represents the model as the product of an AR(1) model and an AR(12) model. Consider the following two ESTIMATE statements:

```
   identify var=sales;
   estimate p=(1 12 13);
   estimate p=(1)(12);
```

The mathematical form of the autoregressive models produced by these two specifications are shown in Table 7.2.

Table 7.2. Subset versus Factored Models

Option	Autoregressive Operator
P=(1 12 13)	$(1 - \phi_1 B - \phi_{12} B^{12} - \phi_{13} B^{13})$
P=(1)(12)	$(1 - \phi_1 B)(1 - \phi_{12} B^{12})$

Both models fit by these two ESTIMATE statements predict SALES from its values 1, 12, and 13 periods ago, but they use different parameterizations. The first model has three parameters, whose meanings may be hard to interpret.

The factored specification P=(1)(12) represents the model as the product of two different AR models. It has only two parameters: one that corresponds to recent effects and one that represents seasonal effects. Thus the factored model is more parsimonious, and its parameter estimates are more clearly interpretable.

Input Variables and Regression with ARMA Errors

In addition to past values of the response series and past errors, you can also model the response series using the current and past values of other series, called *input series*.

Several different names are used to describe ARIMA models with input series. *Transfer function model*, *intervention model*, *interrupted time series model*, *regression model with ARMA errors*, *Box-Tiao model*, and *ARIMAX model* are all different names for ARIMA models with input series. Pankratz (1991) refers to these models as *dynamic regression*.

Using Input Series

To use input series, list the input series in a CROSSCORR= option on the IDENTIFY statement and specify how they enter the model with an INPUT= option on the ESTIMATE statement. For example, you might use a series called PRICE to help model SALES, as shown in the following statements:

```
proc arima data=a;
   identify var=sales crosscorr=price;
   estimate input=price;
   run;
```

This example performs a simple linear regression of SALES on PRICE, producing the same results as PROC REG or another SAS regression procedure. The mathematical form of the model estimated by these statements is

$$Y_t = \mu + \omega_0 X_t + a_t$$

The parameter estimates table for this example (using simulated data) is shown in Figure 7.19. The intercept parameter is labeled MU. The regression coefficient for PRICE is labeled NUM1. (See the section "Naming of Model Parameters" later in this chapter for information on how parameters for input series are named.)

```
                              The ARIMA Procedure

                      Conditional Least Squares Estimation

                          Approx Std
Parameter    Estimate        Error     t Value  Pr > |t|   Lag  Variable  Shift

MU          199.83602      2.99463       66.73    <.0001     0  sales         0
NUM1         -9.99299      0.02885     -346.38    <.0001     0  price         0
```

Figure 7.19. Parameter Estimates Table for Regression Model

Any number of input variables can be used in a model. For example, the following statements fit a multiple regression of SALES on PRICE and INCOME:

```
proc arima data=a;
   identify var=sales crosscorr=(price income);
   estimate input=(price income);
   run;
```

The mathematical form of the regression model estimated by these statements is

$$Y_t = \mu + \omega_1 X_{1,t} + \omega_2 X_{2,t} + a_t$$

Lagging and Differencing Input Series

You can also difference and lag the input series. For example, the following statements regress the change in SALES on the change in PRICE lagged by one period. The difference of PRICE is specified with the CROSSCORR= option and the lag of the change in PRICE is specified by the 1 $ in the INPUT= option.

```
proc arima data=a;
   identify var=sales(1) crosscorr=price(1);
   estimate input=( 1 $ price );
   run;
```

These statements estimate the model

$$(1-B)Y_t = \mu + \omega_0(1-B)X_{t-1} + a_t$$

Regression with ARMA Errors

You can combine input series with ARMA models for the errors. For example, the following statements regress SALES on INCOME and PRICE but with the error term of the regression model (called the *noise series* in ARIMA modeling terminology) assumed to be an ARMA(1,1) process.

```
proc arima data=a;
   identify var=sales crosscorr=(price income);
   estimate p=1 q=1 input=(price income);
   run;
```

These statements estimate the model

$$Y_t = \mu + \omega_1 X_{1,t} + \omega_2 X_{2,t} + \frac{(1-\theta_1 B)}{(1-\phi_1 B)} a_t$$

Stationarity and Input Series

Note that the requirement of stationarity applies to the noise series. If there are no input variables, the response series (after differencing and minus the mean term) and the noise series are the same. However, if there are inputs, the noise series is the residual after the effect of the inputs is removed.

There is no requirement that the input series be stationary. If the inputs are nonstationary, the response series will be nonstationary, even though the noise process may be stationary.

When nonstationary input series are used, you can fit the input variables first with no ARMA model for the errors and then consider the stationarity of the residuals before identifying an ARMA model for the noise part.

Identifying Regression Models with ARMA Errors

Previous sections described the ARIMA modeling identification process using the autocorrelation function plots produced by the IDENTIFY statement. This identification process does not apply when the response series depends on input variables. This is because it is the noise process for which we need to identify an ARIMA model, and when input series are involved the response series adjusted for the mean is no longer an estimate of the noise series.

However, if the input series are independent of the noise series, you can use the residuals from the regression model as an estimate of the noise series, then apply the ARIMA modeling identification process to this residual series. This assumes that the noise process is stationary.

The PLOT option on the ESTIMATE statement produces for the model residuals the same plots as the IDENTIFY statement produces for the response series. The PLOT option prints an autocorrelation function plot, an inverse autocorrelation function plot, and a partial autocorrelation function plot for the residual series.

The following statements show how the PLOT option is used to identify the ARMA(1,1) model for the noise process used in the preceding example of regression with ARMA errors:

```
proc arima data=a;
   identify var=sales crosscorr=(price income) noprint;
   estimate input=(price income) plot;
   run;
   estimate p=1 q=1 input=(price income) plot;
   run;
```

In this example, the IDENTIFY statement includes the NOPRINT option since the autocorrelation plots for the response series are not useful when you know that the response series depends on input series.

The first ESTIMATE statement fits the regression model with no model for the noise process. The PLOT option produces plots of the autocorrelation function, inverse autocorrelation function, and partial autocorrelation function for the residual series of the regression on PRICE and INCOME.

By examining the PLOT option output for the residual series, you verify that the residual series is stationary and identify an ARMA(1,1) model for the noise process. The second ESTIMATE statement fits the final model.

Although this discussion addresses regression models, the same remarks apply to identifying an ARIMA model for the noise process in models that include input series with complex transfer functions.

Intervention Models and Interrupted Time Series

One special kind of ARIMA model with input series is called an *intervention model* or *interrupted time series* model. In an intervention model, the input series is an indicator variable containing discrete values that flag the occurrence of an event affecting the response series. This event is an intervention in or an interruption of the normal evolution of the response time series, which, in the absence of the intervention, is usually assumed to be a pure ARIMA process.

Intervention models can be used both to model and forecast the response series and to analyze the impact of the intervention. When the focus is on estimating the effect of the intervention, the process is often called *intervention analysis* or *interrupted time series analysis*.

Impulse Interventions

The intervention can be a one-time event. For example, you might want to study the effect of a short-term advertising campaign on the sales of a product. In this case, the input variable has the value of 1 for the period during which the advertising campaign took place and the value 0 for all other periods. Intervention variables of this kind are sometimes called *impulse functions* or *pulse functions*.

Suppose that SALES is a monthly series, and a special advertising effort was made during the month of March 1992. The following statements estimate the effect of this intervention assuming an ARMA(1,1) model for SALES. The model is specified just like the regression model, but the intervention variable AD is constructed in the DATA step as a zero-one indicator for the month of the advertising effort.

```
data a;
   set a;
   ad = date = '1mar1992'd;
run;

proc arima data=a;
   identify var=sales crosscorr=ad;
   estimate p=1 q=1 input=ad;
run;
```

Continuing Interventions

Other interventions can be continuing, in which case the input variable flags periods before and after the intervention. For example, you might want to study the effect of a change in tax rates on some economic measure. Another example is a study of the effect of a change in speed limits on the rate of traffic fatalities. In this case, the input variable has the value 1 after the new speed limit went into effect and the value 0 before. Intervention variables of this kind are called *step functions*.

Another example is the effect of news on product demand. Suppose it was reported in July 1996 that consumption of the product prevents heart disease (or causes cancer), and SALES is consistently higher (or lower) thereafter. The following statements model the effect of this news intervention:

```
data a;
   set a;
   news = date >= '1jul1996'd;
run;

proc arima data=a;
   identify var=sales crosscorr=news;
   estimate p=1 q=1 input=news;
run;
```

Interaction Effects

You can include any number of intervention variables in the model. Intervention variables can have any pattern–impulse and continuing interventions are just two possible cases. You can mix discrete valued intervention variables and continuous regressor variables in the same model.

You can also form interaction effects by multiplying input variables and including the product variable as another input. Indeed, as long as the dependent measure forms a regular time series, you can use PROC ARIMA to fit any general linear model in conjunction with an ARMA model for the error process by using input variables that correspond to the columns of the design matrix of the linear model.

Rational Transfer Functions and Distributed Lag Models

How an input series enters the model is called its *transfer function*. Thus, ARIMA models with input series are sometimes referred to as transfer function models.

In the preceding regression and intervention model examples, the transfer function is a single scale parameter. However, you can also specify complex transfer functions composed of numerator and denominator polynomials in the backshift operator. These transfer functions operate on the input series in the same way that the ARMA specification operates on the error term.

Numerator Factors

For example, suppose you want to model the effect of PRICE on SALES as taking place gradually with the impact distributed over several past lags of PRICE. This is illustrated by the following statements:

```
proc arima data=a;
  identify var=sales crosscorr=price;
  estimate input=( (1 2 3) price );
  run;
```

These statements estimate the model

$$Y_t = \mu + (\omega_0 - \omega_1 B - \omega_2 B^2 - \omega_3 B^3) X_t + a_t$$

This example models the effect of PRICE on SALES as a linear function of the current and three most recent values of PRICE. It is equivalent to a multiple linear regression of SALES on PRICE, LAG(PRICE), LAG2(PRICE), and LAG3(PRICE).

This is an example of a transfer function with one *numerator factor*. The numerator factors for a transfer function for an input series are like the MA part of the ARMA model for the noise series.

Denominator Factors

You can also use transfer functions with *denominator factors*. The denominator factors for a transfer function for an input series are like the AR part of the ARMA model for the noise series. Denominator factors introduce exponentially weighted, infinite distributed lags into the transfer function.

To specify transfer functions with denominator factors, place the denominator factors after a slash (/) in the INPUT= option. For example, the following statements estimate the PRICE effect as an infinite distributed lag model with exponentially declining weights:

```
proc arima data=a;
  identify var=sales crosscorr=price;
  estimate input=( / (1) price );
  run;
```

The transfer function specified by these statements is as follows:

$$\frac{\omega_0}{(1-\delta_1 B)} X_t$$

This transfer function also can be written in the following equivalent form:

$$\omega_0 \left(1 + \sum_{i=1}^{\infty} \delta_1^i B^i\right) X_t$$

This transfer function can be used with intervention inputs. When it is used with a pulse function input, the result is an intervention effect that dies out gradually over time. When it is used with a step function input, the result is an intervention effect that increases gradually to a limiting value.

Rational Transfer Functions

By combining various numerator and denominator factors in the INPUT= option, you can specify *rational transfer functions* of any complexity. To specify an input with a general rational transfer function of the form

$$\frac{\omega(B)}{\delta(B)} B^k X_t$$

use an INPUT= option in the ESTIMATE statement of the form

input=(k $ (ω-*lags*) / (δ-*lags*) x)

See the section "Specifying Inputs and Transfer Functions" later in this chapter for more information.

Identifying Transfer Function Models

The CROSSCORR= option of the IDENTIFY statement prints sample cross-correlation functions showing the correlations between the response series and the input series at different lags. The sample cross-correlation function can be used to help identify the form of the transfer function appropriate for an input series. See textbooks on time series analysis for information on using cross-correlation functions to identify transfer function models.

For the cross-correlation function to be meaningful, the input and response series must be filtered with a prewhitening model for the input series. See the section "Prewhitening" later in this chapter for more information on this issue.

Forecasting with Input Variables

To forecast a response series using an ARIMA model with inputs, you need values of the input series for the forecast periods. You can supply values for the input variables for the forecast periods in the DATA= data set, or you can have PROC ARIMA forecast the input variables.

If you do not have future values of the input variables in the input data set used by the FORECAST statement, the input series must be forecast before the ARIMA procedure can forecast the response series. If you fit an ARIMA model to each of the input series for which you need forecasts before fitting the model for the response series, the FORECAST statement automatically uses the ARIMA models for the input series to generate the needed forecasts of the inputs.

For example, suppose you want to forecast SALES for the next 12 months. In this example, we predict the change in SALES as a function of the lagged change in PRICE, plus an ARMA(1,1) noise process. To forecast SALES using PRICE as an input, you also need to fit an ARIMA model for PRICE.

The following statements fit an AR(2) model to the change in PRICE before fitting and forecasting the model for SALES. The FORECAST statement automatically forecasts PRICE using this AR(2) model to get the future inputs needed to produce the forecast of SALES.

```
proc arima data=a;
   identify var=price(1);
   estimate p=2;
   identify var=sales(1) crosscorr=price(1);
   estimate p=1 q=1 input=price;
   forecast lead=12 interval=month id=date out=results;
run;
```

Fitting a model to the input series is also important for identifying transfer functions. (See the section "Prewhitening" later in this chapter for more information.)

Input values from the DATA= data set and input values forecast by PROC ARIMA can be combined. For example, a model for SALES might have three input series: PRICE, INCOME, and TAXRATE. For the forecast, you assume that the tax rate will be unchanged. You have a forecast for INCOME from another source but only for the first few periods of the SALES forecast you want to make. You have no future values for PRICE, which needs to be forecast as in the preceding example.

In this situation, you include observations in the input data set for all forecast periods, with SALES and PRICE set to a missing value, with TAXRATE set to its last actual value, and with INCOME set to forecast values for the periods you have forecasts for and set to missing values for later periods. In the PROC ARIMA step, you estimate ARIMA models for PRICE and INCOME before estimating the model for SALES, as shown in the following statements:

```
proc arima data=a;
  identify var=price(1);
```

```
estimate p=2;
identify var=income(1);
estimate p=2;
identify var=sales(1) crosscorr=( price(1) income(1) taxrate );
estimate p=1 q=1 input=( price income taxrate );
forecast lead=12 interval=month id=date out=results;
run;
```

In forecasting SALES, the ARIMA procedure uses as inputs the value of PRICE forecast by its ARIMA model, the value of TAXRATE found in the DATA= data set, and the value of INCOME found in the DATA= data set, or, when the INCOME variable is missing, the value of INCOME forecast by its ARIMA model. (Because SALES is missing for future time periods, the estimation of model parameters is not affected by the forecast values for PRICE, INCOME, or TAXRATE.)

Data Requirements

PROC ARIMA can handle time series of moderate size; there should be at least 30 observations. With 30 or fewer observations, the parameter estimates may be poor. With thousands of observations, the method requires considerable computer time and memory.

Syntax

The ARIMA procedure uses the following statements:

> **PROC ARIMA** *options*;
> **BY** *variables*;
> **IDENTIFY** *VAR=variable options*;
> **ESTIMATE** *options*;
> **FORECAST** *options*;

Functional Summary

The statements and options controlling the ARIMA procedure are summarized in the following table.

Description	Statement	Option
Data Set Options		
specify the input data set	PROC ARIMA	DATA=
	IDENTIFY	DATA=
specify the output data set	PROC ARIMA	OUT=
	FORECAST	OUT=
include only forecasts in the output data set	FORECAST	NOOUTALL
write autocovariances to output data set	IDENTIFY	OUTCOV=
write parameter estimates to an output data set	ESTIMATE	OUTEST=
write correlation of parameter estimates	ESTIMATE	OUTCORR
write covariance of parameter estimates	ESTIMATE	OUTCOV
write estimated model to an output data set	ESTIMATE	OUTMODEL=
write statistics of fit to an output data set	ESTIMATE	OUTSTAT=
Options for Identifying the Series		
difference time series and plot autocorrelations	IDENTIFY	
specify response series and differencing	IDENTIFY	VAR=
specify and cross correlate input series	IDENTIFY	CROSSCORR=
center data by subtracting the mean	IDENTIFY	CENTER
exclude missing values	IDENTIFY	NOMISS
delete previous models and start fresh	IDENTIFY	CLEAR
specify the significance level for tests	IDENTIFY	ALPHA=
perform tentative ARMA order identification using the ESACF Method	IDENTIFY	ESACF
perform tentative ARMA order identification using the MINIC Method	IDENTIFY	MINIC
perform tentative ARMA order identification using the SCAN Method	IDENTIFY	SCAN

Description	Statement	Option
specify the range of autoregressive model orders for estimating the error series for the MINIC Method	IDENTIFY	PERROR=
determines the AR dimension of the SCAN, ESACF, and MINIC tables	IDENTIFY	P=
determines the MA dimension of the SCAN, ESACF, and MINIC tables	IDENTIFY	Q=
perform stationarity tests	IDENTIFY	STATIONARITY=
Options for Defining and Estimating the Model		
specify and estimate ARIMA models	ESTIMATE	
specify autoregressive part of model	ESTIMATE	P=
specify moving average part of model	ESTIMATE	Q=
specify input variables and transfer functions	ESTIMATE	INPUT=
drop mean term from the model	ESTIMATE	NOINT
specify the estimation method	ESTIMATE	METHOD=
use alternative form for transfer functions	ESTIMATE	ALTPARM
suppress degrees-of-freedom correction in variance estimates	ESTIMATE	NODF
Printing Control Options		
limit number of lags shown in correlation plots	IDENTIFY	NLAG=
suppress printed output for identification	IDENTIFY	NOPRINT
plot autocorrelation functions of the residuals	ESTIMATE	PLOT
print log likelihood around the estimates	ESTIMATE	GRID
control spacing for GRID option	ESTIMATE	GRIDVAL=
print details of the iterative estimation process	ESTIMATE	PRINTALL
suppress printed output for estimation	ESTIMATE	NOPRINT
suppress printing of the forecast values	FORECAST	NOPRINT
print the one-step forecasts and residuals	FORECAST	PRINTALL
Options to Specify Parameter Values		
specify autoregressive starting values	ESTIMATE	AR=
specify moving average starting values	ESTIMATE	MA=
specify a starting value for the mean parameter	ESTIMATE	MU=
specify starting values for transfer functions	ESTIMATE	INITVAL=
Options to Control the Iterative Estimation Process		
specify convergence criterion	ESTIMATE	CONVERGE=
specify the maximum number of iterations	ESTIMATE	MAXITER=

Description	**Statement**	**Option**
specify criterion for checking for singularity	ESTIMATE	SINGULAR=
suppress the iterative estimation process	ESTIMATE	NOEST
omit initial observations from objective	ESTIMATE	BACKLIM=
specify perturbation for numerical derivatives	ESTIMATE	DELTA=
omit stationarity and invertibility checks	ESTIMATE	NOSTABLE
use preliminary estimates as starting values for ML and ULS	ESTIMATE	NOLS
Options for Forecasting		
forecast the response series	FORECAST	
specify how many periods to forecast	FORECAST	LEAD=
specify the ID variable	FORECAST	ID=
specify the periodicity of the series	FORECAST	INTERVAL=
specify size of forecast confidence limits	FORECAST	ALPHA=
start forecasting before end of the input data	FORECAST	BACK=
specify the variance term used to compute forecast standard errors and confidence limits	FORECAST	SIGSQ=
control the alignment of SAS Date values	FORECAST	ALIGN=
BY Groups		
specify BY group processing	BY	

PROC ARIMA Statement

PROC ARIMA *options;*

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

DATA= *SAS-data-set*
specifies the name of the SAS data set containing the time series. If different DATA= specifications appear in the PROC ARIMA and IDENTIFY statements, the one in the IDENTIFY statement is used. If the DATA= option is not specified in either the PROC ARIMA or IDENTIFY statement, the most recently created SAS data set is used.

OUT= *SAS-data-set*
specifies a SAS data set to which the forecasts are output. If different OUT= specifications appear in the PROC ARIMA and FORECAST statement, the one in the FORECAST statement is used.

BY Statement

BY *variables;*

A BY statement can be used in the ARIMA procedure to process a data set in groups of observations defined by the BY variables. Note that all IDENTIFY, ESTIMATE, and FORECAST statements specified are applied to all BY groups.

Because of the need to make data-based model selections, BY-group processing is not usually done with PROC ARIMA. You usually want different models for the different series contained in different BY-groups, and the PROC ARIMA BY statement does not let you do this.

Using a BY statement imposes certain restrictions. The BY statement must appear before the first RUN statement. If a BY statement is used, the input data must come from the data set specified in the PROC statement; that is, no input data sets can be specified in IDENTIFY statements.

When a BY statement is used with PROC ARIMA, interactive processing only applies to the first BY group. Once the end of the PROC ARIMA step is reached, all ARIMA statements specified are executed again for each of the remaining BY groups in the input data set.

IDENTIFY Statement

IDENTIFY *VAR=variable options;*

The IDENTIFY statement specifies the time series to be modeled, differences the series if desired, and computes statistics to help identify models to fit. Use an IDENTIFY statement for each time series that you want to model.

If other time series are to be used as inputs in a subsequent ESTIMATE statement, they must be listed in a CROSSCORR= list in the IDENTIFY statement.

The following options are used in the IDENTIFY statement. The VAR= option is required.

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

CENTER
centers each time series by subtracting its sample mean. The analysis is done on the centered data. Later, when forecasts are generated, the mean is added back. Note that centering is done after differencing. The CENTER option is normally used in conjunction with the NOCONSTANT option of the ESTIMATE statement.

CLEAR
deletes all old models. This option is useful when you want to delete old models so

that the input variables are not prewhitened. (See the section "Prewhitening" later in this chapter for more information.)

CROSSCORR= *variable* **(***d11, d12, ..., d1k***)**
CROSSCORR= (*variable* **(***d11, d12, ..., d1k***) ...** *variable* **(***d21, d22, ..., d2k***))**

names the variables cross correlated with the response variable given by the VAR= specification.

Each variable name can be followed by a list of differencing lags in parentheses, the same as for the VAR= specification. If differencing is specified for a variable in the CROSSCORR= list, the differenced series is cross correlated with the VAR= option series, and the differenced series is used when the ESTIMATE statement INPUT= option refers to the variable.

DATA= *SAS-data-set*

specifies the input SAS data set containing the time series. If the DATA= option is omitted, the DATA= data set specified in the PROC ARIMA statement is used; if the DATA= option is omitted from the PROC ARIMA statement as well, the most recently created data set is used.

ESACF

computes the extended sample autocorrelation function and uses these estimates to tentatively identify the autoregressive and moving average orders of mixed models.

The ESACF option generates two tables. The first table displays extended sample autocorrelation estimates, and the second table displays probability values that can be used to test the significance of these estimates. The P=$(p_{min} : p_{max})$ and Q=$(q_{min} : q_{max})$ options determine the size of the table.

The autoregressive and moving average orders are tentatively identified by finding a triangular pattern in which all values are insignificant. The ARIMA procedure finds these patterns based on the IDENTIFY statement ALPHA= option and displays possible recommendations for the orders.

The following code generates an ESACF table with dimensions of p=(0:7) and q=(0:8).

```
proc arima data=test;
   identify var=x esacf p=(0:7) q=(0:8);
run;
```

See the “The ESACF Method” section on page 236 for more information.

MINIC

uses information criteria or penalty functions to provide tentative ARMA order identification. The MINIC option generates a table containing the computed information criterion associated with various ARMA model orders. The PERROR=$(p_{\epsilon,min} : p_{\epsilon,max})$ option determines the range of the autoregressive model orders used to estimate the error series. The P=$(p_{min} : p_{max})$ and Q=$(q_{min} : q_{max})$ options determine the size of the table. The ARMA orders are tentatively identified by those orders that minimize the information criterion.

The following code generates a MINIC table with default dimensions of p=(0:5) and q=(0:5) and with the error series estimated by an autoregressive model with an order, p_ϵ, that minimizes the AIC in the range from 8 to 11.

```
proc arima data=test;
   identify var=x minic perror=(8:11);
run;
```

See the “The MINIC Method” section on page 238 for more information.

NLAG= *number*

indicates the number of lags to consider in computing the autocorrelations and cross correlations. To obtain preliminary estimates of an ARIMA(*p*,*d*,*q*) model, the NLAG= value must be at least *p*+*q*+*d*. The number of observations must be greater than or equal to the NLAG= value. The default value for NLAG= is 24 or one-fourth the number of observations, whichever is less. Even though the NLAG= value is specified, the NLAG= value can be changed according to the data set.

NOMISS

uses only the first continuous sequence of data with no missing values. By default, all observations are used.

NOPRINT

suppresses the normal printout (including the correlation plots) generated by the IDENTIFY statement.

OUTCOV= *SAS-data-set*

writes the autocovariances, autocorrelations, inverse autocorrelations, partial autocorrelations, and cross covariances to an output SAS data set. If the OUTCOV= option is not specified, no covariance output data set is created. See the section "OUTCOV= Data Set" later in this chapter for more information.

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

see the ESCAF, MINIC, and SCAN options for details.

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

see the ESCAF, MINIC, and SCAN options for details.

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

see the ESACF, MINIC, and SCAN options for details.

SCAN

computes estimates of the squared canonical correlations and uses these estimates to tentatively identify the autoregressive and moving average orders of mixed models.

The SCAN option generates two tables. The first table displays squared canonical correlation estimates, and the second table displays probability values that can be used to test the significance of these estimates. The P=$(p_{min} : p_{max})$ and Q=$(q_{min} : q_{max})$ options determine the size of each table.

The autoregressive and moving average orders are tentatively identified by finding a rectangular pattern in which all values are insignificant. The ARIMA procedure finds these patterns based on the IDENTIFY statement ALPHA= option and displays possible recommendations for the orders.

The following code generates a SCAN table with default dimensions of p=(0:5) and q=(0:5). The recommended orders are based on a significance level of 0.1.

```
proc arima data=test;
   identify var=x scan alpha=0.1;
run;
```

See the “The SCAN Method” section on page 239 for more information.

STATIONARITY=

performs stationarity tests. Stationarity tests can be used to determine whether differencing terms should be included in the model specification. In each stationarity test, the autoregressive orders can be specified by a range, *test*=ar_{max}, or as a list of values, *test*=$(ar_1, .., ar_n)$, where *test* is ADF, PP, or RW. The default is (0,1,2).

See the “Stationarity Tests” section on page 241 for more information.

STATIONARITY=(ADF= *AR orders* **DLAG=** *s***)**
STATIONARITY=(DICKEY= *AR orders* **DLAG=** *s***)**

performs augmented Dickey-Fuller tests. If the DLAG=s option specified with s is greater than one, seasonal Dickey-Fuller tests are performed. The maximum allowable value of s is 12. The default value of s is one. The following code performs augmented Dickey-Fuller tests with autoregressive orders 2 and 5.

```
proc arima data=test;
   identify var=x stationarity=(adf=(2,5));
run;
```

STATIONARITY=(PP= *AR orders***)**
STATIONARITY=(PHILLIPS= *AR orders***)**

performs Phillips-Perron tests. The following code performs Augmented Phillips-Perron tests with autoregressive orders ranging from 0 to 6.

```
proc arima data=test;
   identify var=x stationarity=(pp=6);
run;
```

STATIONARITY=(RW= *AR orders***)**
STATIONARITY=(RANDOMWALK= *AR orders***)**

performs random-walk with drift tests. The following code performs random-walk with drift tests with autoregressive orders ranging from 0 to 2.

```
proc arima data=test;
   identify var=x stationarity=(rw);
run;
```

VAR= *variable*
VAR= *variable* **(** *d1, d2, ..., dk* **)**

names the variable containing the time series to analyze. The VAR= option is required.

A list of differencing lags can be placed in parentheses after the variable name to request that the series be differenced at these lags. For example, VAR=X(1) takes the first differences of X. VAR=X(1,1) requests that X be differenced twice, both times with lag 1, producing a second difference series, which is $(X_t - X_{t-1}) - (X_{t-1} - X_{t-2}) = X_t - 2X_{t-1} + X_{t-2}$.

VAR=X(2) differences X once at lag two $(X_t - X_{t-2})$.

If differencing is specified, it is the differenced series that is processed by any subsequent ESTIMATE statement.

ESTIMATE Statement

ESTIMATE *options;*

The ESTIMATE statement specifies an ARMA model or transfer function model for the response variable specified in the previous IDENTIFY statement, and produces estimates of its parameters. The ESTIMATE statement also prints diagnostic information by which to check the model. Include an ESTIMATE statement for each model that you want to estimate.

Options used in the ESTIMATE statement are described in the following sections.

Options for Defining the Model and Controlling Diagnostic Statistics

The following options are used to define the model to be estimated and to control the output that is printed.

ALTPARM

specifies the alternative parameterization of the overall scale of transfer functions in the model. See the section "Alternative Model Parameterization" later in this chapter for details.

INPUT= *variable*
INPUT= (*transfer-function variable* **...)**

specifies input variables and their transfer functions.

The variables used on the INPUT= option must be included in the CROSSCORR= list in the previous IDENTIFY statement. If any differencing is specified in the CROSSCORR= list, then the differenced series is used as the input to the transfer function.

The transfer function specification for an input variable is optional. If no transfer function is specified, the input variable enters the model as a simple regressor. If specified, the transfer function specification has the following syntax:

$S\$(L_{1,1}, L_{1,2}, \ldots)(L_{2,1}, \ldots) \ldots /(L_{j,1}, \ldots) \ldots$

Here, *S* is a shift or lag of the input variable, the terms before the slash (/) are numerator factors, and the terms after the slash (/) are denominator factors of the transfer function. All three parts are optional. See the section "Specifying Inputs and Transfer Functions" later in this chapter for details.

METHOD=ML
METHOD=ULS
METHOD=CLS

specifies the estimation method to use. METHOD=ML specifies the maximum likelihood method. METHOD=ULS specifies the unconditional least-squares method. METHOD=CLS specifies the conditional least-squares method. METHOD=CLS is the default. See the section "Estimation Details" later in this chapter for more information.

NOCONSTANT
NOINT

suppresses the fitting of a constant (or intercept) parameter in the model. (That is, the parameter μ is omitted.)

NODF

estimates the variance by dividing the error sum of squares (SSE) by the number of residuals. The default is to divide the SSE by the number of residuals minus the number of free parameters in the model.

NOPRINT

suppresses the normal printout generated by the ESTIMATE statement. If the NOPRINT option is specified for the ESTIMATE statement, then any error and warning messages are printed to the SAS log.

P= *order*
P= (*lag*, ..., *lag***) ... (***lag*, ..., *lag***)**

specifies the autoregressive part of the model. By default, no autoregressive parameters are fit.

P=(l_1, l_2, ..., l_k) defines a model with autoregressive parameters at the specified lags. P= *order* is equivalent to P=(1, 2, ..., *order*).

A concatenation of parenthesized lists specifies a factored model. For example, P=(1,2,5)(6,12) specifies the autoregressive model

$$(1 - \phi_{1,1}B - \phi_{1,2}B^2 - \phi_{1,3}B^5)(1 - \phi_{2,1}B^6 - \phi_{2,2}B^{12})$$

PLOT

plots the residual autocorrelation functions. The sample autocorrelation, the sample inverse autocorrelation, and the sample partial autocorrelation functions of the model residuals are plotted.

Q= *order*
Q= (*lag, ..., lag***) ... (***lag, ..., lag***)**
specifies the moving-average part of the model. By default, no moving-average part is included in the model.

Q=(l_1, l_2, ..., l_k) defines a model with moving-average parameters at the specified lags. Q= *order* is equivalent to Q=(1, 2, ..., *order*). A concatenation of parenthesized lists specifies a factored model. The interpretation of factors and lags is the same as for the P= option.

Options for Output Data Sets

The following options are used to store results in SAS data sets:

OUTEST= *SAS-data-set*
writes the parameter estimates to an output data set. If the OUTCORR or OUTCOV option is used, the correlations or covariances of the estimates are also written to the OUTEST= data set. See the section "OUTEST= Data Set" later in this chapter for a description of the OUTEST= output data set.

OUTCORR
writes the correlations of the parameter estimates to the OUTEST= data set.

OUTCOV
writes the covariances of the parameter estimates to the OUTEST= data set.

OUTMODEL= *SAS-data-set*
writes the model and parameter estimates to an output data set. If OUTMODEL= is not specified, no model output data set is created. See the section "OUTMODEL= Data Set" for a description of the OUTMODEL= output data set.

OUTSTAT= *SAS-data-set*
writes the model diagnostic statistics to an output data set. If OUTSTAT= is not specified, no statistics output data set is created. See the section "OUTSTAT= Data Set" later in this chapter for a description of the OUTSTAT= output data set.

Options to Specify Parameter Values

The following options enable you to specify values for the model parameters. These options can provide starting values for the estimation process, or you can specify fixed parameters for use in the FORECAST stage and suppress the estimation process with the NOEST option. By default, the ARIMA procedure finds initial parameter estimates and uses these estimates as starting values in the iterative estimation process.

If values for any parameters are specified, values for all parameters should be given. The number of values given must agree with the model specifications.

AR= *value* **...**
lists starting values for the autoregressive parameters. See "Initial Values" later in this chapter for more information.

INITVAL= (*initializer-spec variable* **...)**
specifies starting values for the parameters in the transfer function parts of the model. See "Initial Values" later in this chapter for more information.

MA= *value* **...**
lists starting values for the moving-average parameters. See "Initial Values" later in this chapter for more information.

MU= *value*
specifies the MU parameter.

NOEST
uses the values specified with the AR=, MA=, INITVAL=, and MU= options as final parameter values. The estimation process is suppressed except for estimation of the residual variance. The specified parameter values are used directly by the next FORECAST statement. When NOEST is specified, standard errors, *t* values, and the correlations between estimates are displayed as 0 or missing. (The NOEST option is useful, for example, when you wish to generate forecasts corresponding to a published model.)

Options to Control the Iterative Estimation Process

The following options can be used to control the iterative process of minimizing the error sum of squares or maximizing the log likelihood function. These tuning options are not usually needed but may be useful if convergence problems arise.

BACKLIM= $-n$
omits the specified number of initial residuals from the sum of squares or likelihood function. Omitting values can be useful for suppressing transients in transfer function models that are sensitive to start-up values.

CONVERGE= *value*
specifies the convergence criterion. Convergence is assumed when the largest change in the estimate for any parameter is less that the CONVERGE= option value. If the absolute value of the parameter estimate is greater than 0.01, the relative change is used; otherwise, the absolute change in the estimate is used. The default is CONVERGE=.001.

DELTA= *value*
specifies the perturbation value for computing numerical derivatives. The default is DELTA=.001.

GRID
prints the error sum of squares (SSE) or concentrated log likelihood surface in a small grid of the parameter space around the final estimates. For each pair of parameters, the SSE is printed for the nine parameter-value combinations formed by the grid, with a center at the final estimates and with spacing given by the GRIDVAL= specification. The GRID option may help you judge whether the estimates are truly at the optimum, since the estimation process does not always converge. For models with a large number of parameters, the GRID option produces voluminous output.

GRIDVAL= *number*
controls the spacing in the grid printed by the GRID option. The default is GRIDVAL=0.005.

MAXITER= n
MAXIT= n
specifies the maximum number of iterations allowed. The default is MAXITER=50. (The default was 15 in previous releases of SAS/ETS software.)

NOLS
begins the maximum likelihood or unconditional least-squares iterations from the preliminary estimates rather than from the conditional least-squares estimates that are produced after four iterations. See the section "Estimation Details" later in this chapter for details.

NOSTABLE
specifies that the autoregressive and moving-average parameter estimates for the noise part of the model not be restricted to the stationary and invertible regions, respectively. See the section "Stationarity and Invertibility" later in this chapter for more information.

PRINTALL
prints preliminary estimation results and the iterations in the final estimation process.

SINGULAR= *value*
specifies the criterion for checking singularity. If a pivot of a sweep operation is less than the SINGULAR= value, the matrix is deemed singular. Sweep operations are performed on the Jacobian matrix during final estimation and on the covariance matrix when preliminary estimates are obtained. The default is SINGULAR=1E-7.

FORECAST Statement

FORECAST *options;*

The FORECAST statement generates forecast values for a time series using the parameter estimates produced by the previous ESTIMATE statement. See the section "Forecasting Details" later in this chapter for more information on calculating forecasts.

The following options can be used in the FORECAST statement:

ALIGN= *option*
controls the alignment of SAS dates used to identify output observations. The ALIGN= option allows the following values: BEGINNING|BEG|B, MIDDLE|MID|M, and ENDING|END|E. BEGINNING is the default.

ALPHA= *n*
sets the size of the forecast confidence limits. The ALPHA= value must be between 0 and 1. When you specify ALPHA=α, the upper and lower confidence limits will have a $1 - \alpha$ confidence level. The default is ALPHA=.05, which produces 95% confidence intervals. ALPHA values are rounded to the nearest hundredth.

BACK= *n*

specifies the number of observations before the end of the data that the multistep forecasts are to begin. The BACK= option value must be less than or equal to the number of observations minus the number of parameters.

The default is BACK=0, which means that the forecast starts at the end of the available data. The end of the data is the last observation for which a noise value can be calculated. If there are no input series, the end of the data is the last nonmissing value of the response time series. If there are input series, this observation can precede the last nonmissing value of the response variable, since there may be missing values for some of the input series.

ID= *variable*

names a variable in the input data set that identifies the time periods associated with the observations. The ID= variable is used in conjunction with the INTERVAL= option to extrapolate ID values from the end of the input data to identify forecast periods in the OUT= data set.

If the INTERVAL= option specifies an interval type, the ID variable must be a SAS date or datetime variable with the spacing between observations indicated by the INTERVAL= value. If the INTERVAL= option is not used, the last input value of the ID= variable is incremented by one for each forecast period to extrapolate the ID values for forecast observations.

INTERVAL= *interval*
INTERVAL= *n*

specifies the time interval between observations. See Chapter 3, “Date Intervals, Formats, and Functions,” for information on valid INTERVAL= values.

The value of the INTERVAL= option is used by PROC ARIMA to extrapolate the ID values for forecast observations and to check that the input data are in order with no missing periods. See the section "Specifying Series Periodicity" later in this chapter for more details.

LEAD= *n*

specifies the number of multistep forecast values to compute. For example, if LEAD=10, PROC ARIMA forecasts for ten periods beginning with the end of the input series (or earlier if BACK= is specified). It is possible to obtain fewer than the requested number of forecasts if a transfer function model is specified and insufficient data are available to compute the forecast. The default is LEAD=24.

NOOUTALL

includes only the final forecast observations in the output data set, not the one-step forecasts for the data before the forecast period.

NOPRINT

suppresses the normal printout of the forecast and associated values.

OUT= *SAS-data-set*
writes the forecast (and other values) to an output data set. If OUT= is not specified, the OUT= data set specified in the PROC ARIMA statement is used. If OUT= is also not specified in the PROC ARIMA statement, no output data set is created. See the section "OUT= Data Set" later in this chapter for more information.

PRINTALL
prints the FORECAST computation throughout the whole data set. The forecast values for the data before the forecast period (specified by the BACK= option) are one-step forecasts.

SIGSQ=
specifies the variance term used in the formula for computing forecast standard errors and confidence limits. The default value is the Variance Estimate computed by the preceding ESTIMATE statement. This option is useful when you wish to generate forecast standard errors and confidence limits based on a published model. It would often be used in conjunction with the NOEST option in the preceding ESTIMATE statement.

Details

The Inverse Autocorrelation Function

The sample inverse autocorrelation function (SIACF) plays much the same role in ARIMA modeling as the sample partial autocorrelation function (SPACF) but generally indicates subset and seasonal autoregressive models better than the SPACF.

Additionally, the SIACF may be useful for detecting over-differencing. If the data come from a nonstationary or nearly nonstationary model, the SIACF has the characteristics of a noninvertible moving average. Likewise, if the data come from a model with a noninvertible moving average, then the SIACF has nonstationary characteristics and therefore decays slowly. In particular, if the data have been over-differenced, the SIACF looks like a SACF from a nonstationary process.

The inverse autocorrelation function is not often discussed in textbooks, so a brief description is given here. More complete discussions can be found in Cleveland (1972), Chatfield (1980), and Priestly (1981).

Let W_t be generated by the ARMA(*p*,*q*) process

$$\phi(B)W_t = \theta(B)a_t$$

where a_t is a white noise sequence. If $\theta(B)$ is invertible (that is, if θ considered as a polynomial in B has no roots less than or equal to 1 in magnitude), then the model

$$\theta(B)Z_t = \phi(B)a_t$$

is also a valid ARMA(*q*,*p*) model. This model is sometimes referred to as the dual model. The autocorrelation function (ACF) of this dual model is called the inverse autocorrelation function (IACF) of the original model.

Notice that if the original model is a pure autoregressive model, then the IACF is an ACF corresponding to a pure moving-average model. Thus, it cuts off sharply when the lag is greater than *p*; this behavior is similar to the behavior of the partial autocorrelation function (PACF).

The sample inverse autocorrelation function (SIACF) is estimated in the ARIMA procedure by the following steps. A high-order autoregressive model is fit to the data by means of the Yule-Walker equations. The order of the autoregressive model used to calculate the SIACF is the minimum of the NLAG= value and one-half the number of observations after differencing. The SIACF is then calculated as the autocorrelation function that corresponds to this autoregressive operator when treated as a moving-average operator. That is, the autoregressive coefficients are convolved with themselves and treated as autocovariances.

Under certain conditions, the sampling distribution of the SIACF can be approximated by the sampling distribution of the SACF of the dual model (Bhansali 1980). In the plots generated by ARIMA, the confidence limit marks (.) are located at $\pm 2/\sqrt{n}$. These limits bound an approximate 95% confidence interval for the hypothesis that the data are from a white noise process.

The Partial Autocorrelation Function

The approximation for a standard error for the estimated partial autocorrelation function at lag *k* is based on a null hypothesis that a pure autoregressive Gaussian process of order *k*-1 generated the time series. This standard error is $1/\sqrt{n}$ and is used to produce the approximate 95% confidence intervals depicted by the dots in the plot.

The Cross-Correlation Function

The autocorrelation, partial and inverse autocorrelation functions described in the preceding sections help when you want to model a series as a function of its past values and past random errors. When you want to include the effects of past and current values of other series in the model, the correlations of the response series and the other series must be considered.

The CROSSCORR= option on the IDENTIFY statement computes cross correlations of the VAR= series with other series and makes these series available for use as inputs in models specified by later ESTIMATE statements.

When the CROSSCORR= option is used, PROC ARIMA prints a plot of the cross-correlation function for each variable in the CROSSCORR= list. This plot is similar in format to the other correlation plots, but shows the correlation between the two series at both lags and leads. For example

```
   identify var=y crosscorr=x ...;
```

plots the cross-correlation function of Y and X, $\mathrm{Cor}(\mathrm{y_t}, \mathrm{x_{t-s}})$, for $s = -L$ to L, where L is the value of the NLAG= option. Study of the cross-correlation functions can indicate the transfer functions through which the input series should enter the model for the response series.

The cross-correlation function is computed after any specified differencing has been done. If differencing is specified for the VAR= variable or for a variable in the CROSSCORR= list, it is the differenced series that is cross correlated (and the differenced series is processed by any following ESTIMATE statement).

For example,

```
identify var=y(1) crosscorr=x(1);
```

computes the cross correlations of the changes in Y with the changes in X. Any following ESTIMATE statement models changes in the variables rather than the variables themselves.

The ESACF Method

The **E**xtended **S**ample **A**uto**c**orrelation **F**unction (ESACF) method can tentatively identify the orders of a *stationary or nonstationary* ARMA process based on iterated least squares estimates of the autoregressive parameters. Tsay and Tiao (1984) proposed the technique, and Choi (1990) provides useful descriptions of the algorithm.

Given a stationary or nonstationary time series $\{z_t : 1 \leq t \leq n\}$ with mean corrected form $\tilde{z}_t = z_t - \mu_z$, with a true autoregressive order of $p+d$, and with a true moving-average order of q, you can use the ESACF method to estimate the unknown orders $p+d$ and q by analyzing the autocorrelation functions associated with filtered series of the form

$$w_t^{(m,j)} = \hat{\Phi}_{(m,j)}(B)\tilde{z}_t = \tilde{z}_t - \sum_{i=1}^{m} \hat{\phi}_i^{(m,j)} \tilde{z}_{t-i}$$

where B represents the backshift operator, where $m = p_{min}, \ldots, p_{max}$ are the autoregressive *test* orders, where $j = q_{min}+1, \ldots, q_{max}+1$ are the moving average *test* orders, and where $\hat{\phi}_i^{(m,j)}$ are the autoregressive parameter estimates under the assumption that the series is an ARMA(m, j) process.

For purely autoregressive models ($j = 0$), ordinary least squares (OLS) is used to consistently estimate $\hat{\phi}_i^{(m,0)}$. For ARMA models, consistent estimates are obtained by the iterated least squares recursion formula, which is initiated by the pure autoregressive estimates:

$$\hat{\phi}_i^{(m,j)} = \hat{\phi}_i^{(m+1,j-1)} - \hat{\phi}_{i-1}^{(m,j-1)} \frac{\hat{\phi}_{m+1}^{(m+1,j-1)}}{\hat{\phi}_m^{(m,j-1)}}$$

The jth lag of the sample autocorrelation function of the filtered series, $w_t^{(m,j)}$, is the *extended sample autocorrelation function*, and it is denoted as $r_{j(m)} = r_j(w^{(m,j)})$.

The standard errors of $r_{j(m)}$ are computed in the usual way using Bartlett's approximation of the variance of the sample autocorrelation function, $var(r_{j(m)}) \approx (1 + \sum_{t=1}^{j-1} r_j^2(w^{(m,j)}))$.

If the true model is an ARMA $(p+d,q)$ process, the filtered series, $w_t^{(m,j)}$, follows an MA(q) model for $j{\geq}q$ so that

$$r_{j(p+d)} \approx 0 \quad j > q$$

$$r_{j(p+d)} \neq 0 \quad j = q$$

Additionally, Tsay and Tiao (1984) show that the extended sample autocorrelation satisfies

$$r_{j(m)} \approx 0 \qquad\qquad j-q > m-p-d \leq 0$$

$$r_{j(m)} \neq c(m-p-d, j-q) \qquad 0 \leq j-q \leq m-p-d$$

where $c(m-p-d, j-q)$ is a nonzero constant or a continuous random variable bounded by -1 and 1.

An ESACF table is then constructed using the $r_{j(m)}$ for $m = p_{min,} \ldots, p_{max}$ and $j = q_{min}+1, \ldots, q_{max}+1$ to identify the ARMA orders (see Table 7.3). The orders are tentatively identified by finding a right (maximal) triangular pattern with vertices located at $(p+d,q)$ and $(p+d,q_{max})$ and in which all elements are insignificant (based on asymptotic normality of the autocorrelation function). The vertex $(p+d,q)$ identifies the order. Table 7.4 depicts the theoretical pattern associated with an ARMA(1,2) series.

Table 7.3. ESACF Table

	MA					
AR	0	1	2	3	·	·
0	$r_{1(0)}$	$r_{2(0)}$	$r_{3(0)}$	$r_{4(0)}$	·	·
1	$r_{1(1)}$	$r_{2(1)}$	$r_{3(1)}$	$r_{4(1)}$	·	·
2	$r_{1(2)}$	$r_{2(2)}$	$r_{3(2)}$	$r_{4(2)}$	·	·
3	$r_{1(3)}$	$r_{2(3)}$	$r_{3(3)}$	$r_{4(3)}$	·	·
·	·	·	·	·	·	·
·	·	·	·	·	·	·

Table 7.4. Theoretical ESACF Table for an ARMA(1,2) Series

	MA							
AR	0	1	2	3	4	5	6	7
0	*	X	X	X	X	X	X	X
1	*	X	0	0	0	0	0	0
2	*	X	X	0	0	0	0	0
3	*	X	X	X	0	0	0	0
4	*	X	X	X	X	0	0	0
	X = significant terms 0 = insignificant terms * = no pattern							

The MINIC Method

The **MIN**imum **I**nformation **C**riterion (MINIC) method can tentatively identify the order of a *stationary and invertible* ARMA process. Note that Hannan and Rissannen (1982) proposed this method, and Box *et al* (1994) and Choi (1990) provide useful descriptions of the algorithm.

Given a stationary and invertible time series $\{z_t : 1 \leq t \leq n\}$ with mean corrected form $\tilde{z}_t = z_t - \mu_z$, with a true autoregressive order of p, and with a true moving-average order of q, you can use the MINIC method to compute information criteria (or penalty functions) for various autoregressive and moving average orders. The following paragraphs provide a brief description of the algorithm.

If the series is a stationary and invertible ARMA(*p*,*q*) process of the form

$$\Phi_{(p,q)}(B)\tilde{z}_t = \Theta_{(p,q)}(B)\epsilon_t$$

the error series can be approximated by a high-order AR process

$$\hat{\epsilon}_t = \hat{\Phi}_{(p_\epsilon,q)}(B)\tilde{z}_t \approx \epsilon_t$$

where the parameter estimates $\hat{\Phi}_{(p_\epsilon,q)}$ are obtained from the Yule-Walker estimates. The choice of the autoregressive order, p_ϵ, is determined by the order that minimizes the Akaike information criterion (AIC) in the range $p_{\epsilon,min} \leq p_\epsilon \leq p_{\epsilon,max}$

$$AIC(p_\epsilon,0) = \ln(\tilde{\sigma}^2_{(p_\epsilon,0)}) + 2(p_\epsilon + 0)/n$$

where

$$\tilde{\sigma}^2_{(p_\epsilon,0)} = \frac{1}{n}\sum_{t=p_\epsilon+1}^{n}\hat{\epsilon}_t^2$$

Note that Hannan and Rissannen (1982) use the Bayesian information criterion (BIC) to determine the autoregressive order used to estimate the error series. Box *et al* (1994) and Choi (1990) recommend the AIC.

Once the error series has been estimated for autoregressive *test* order $m = p_{min}, \ldots, p_{max}$ and for moving-average *test* order $j = q_{min}, \ldots, q_{max}$, the OLS estimates $\hat{\Phi}_{(m,j)}$ and $\hat{\Theta}_{(m,j)}$ are computed from the regression model

$$\tilde{z}_t = \sum_{i=1}^{m}\phi_i^{(m,j)}\tilde{z}_{t-i} + \sum_{k=1}^{j}\theta_k^{(m,j)}\hat{\epsilon}_{t-k} + error$$

From the preceding parameter estimates, the BIC is then computed

$$BIC(m,j) = \ln(\tilde{\sigma}^2_{(m,j)}) + 2(m+j)\ln(n)/n$$

where

$$\tilde{\sigma}^2_{(m,j)} = \frac{1}{n}\sum_{t=t_0}^{n}\left(\tilde{z}_t - \sum_{i=1}^{m}\phi_i^{(m,j)}\tilde{z}_{t-i} + \sum_{k=1}^{j}\theta_k^{(m,j)}\hat{\epsilon}_{t-k}\right)$$

where $t_0 = p_\epsilon + \max(\mathrm{m}, \mathrm{j})$.

A MINIC table is then constructed using $BIC(m, j)$ (see Table 7.5). If $p_{max} > p_{\epsilon,min}$, the preceding regression may fail due to linear dependence on the estimated error series and the mean-corrected series. Values of $BIC(m, j)$ that cannot be computed are set to missing. For large autoregressive and moving average test orders with relatively few observations, a nearly perfect fit can result. This condition can be identified by a large negative $BIC(m, j)$ value.

Table 7.5. MINIC Table

	MA					
AR	0	1	2	3	·	·
0	$BIC(0,0)$	$BIC(0,1)$	$BIC(0,2)$	$BIC(0,3)$	·	·
1	$BIC(1,0)$	$BIC(1,1)$	$BIC(1,2)$	$BIC(1,3)$	·	·
2	$BIC(2,0)$	$BIC(2,1)$	$BIC(2,2)$	$BIC(2,3)$	·	·
3	$BIC(3,0)$	$BIC(3,1)$	$BIC(3,2)$	$BIC(3,3)$	·	·
·	·	·	·	·	·	·
·	·	·	·	·	·	·

The SCAN Method

The **S**mallest **CAN**onical (SCAN) correlation method can tentatively identify the orders of a *stationary or nonstationary* ARMA process. Tsay and Tiao (1985) proposed the technique, and Box *et al* (1994) and Choi (1990) provide useful descriptions of the algorithm.

Given a stationary or nonstationary time series $\{z_t : 1 \le t \le n\}$ with mean corrected form $\tilde{z}_t = z_t - \mu_z$, with a true autoregressive order of $p + d$, and with a true moving-average order of q, you can use the SCAN method to analyze eigenvalues of the correlation matrix of the ARMA process. The following paragraphs provide a brief description of the algorithm.

For autoregressive *test* order $m = p_{min}, \ldots, p_{max}$ and for moving-average *test* order $j = q_{min}, \ldots, q_{max}$, perform the following steps.

1. Let $Y_{m,t} = (\tilde{z}_t, \tilde{z}_{t-1}, \ldots, \tilde{z}_{t-m})'$. Compute the following $(m + 1) \times (m + 1)$ matrix

$$\begin{aligned}\hat{\beta}(m, j+1) &= \left(\sum_t Y_{m,t-j-1}Y'_{m,t-j-1}\right)^{-1}\left(\sum_t Y_{m,t-j-1}Y'_{m,t}\right)\\ \hat{\beta}^*(m, j+1) &= \left(\sum_t Y_{m,t}Y'_{m,t}\right)^{-1}\left(\sum_t Y_{m,t}Y'_{m,t-j-1}\right)\end{aligned}$$

$$\hat{A}^*(m,j) \quad = \quad \hat{\beta}^*(m,j+1)\hat{\beta}(m,j+1)$$

where t ranges from $j+m+2$ to n.

2. Find the *smallest* eigenvalue, $\hat{\lambda}^*(m,j)$, of $\hat{A}^*(m,j)$ and its corresponding *normalized* eigenvector, $\Phi_{m,j} = (1, -\phi_1^{(m,j)}, -\phi_2^{(m,j)}, \ldots, -\phi_m^{(m,j)})$. The squared canonical correlation estimate is $\hat{\lambda}^*(m,j)$.

3. Using the $\Phi_{m,j}$ as AR(m) coefficients, obtain the residuals for $t = j+m+1$ to n, by following the formula: $w_t^{(m,j)} = \tilde{z}_t - \phi_1^{(m,j)}\tilde{z}_{t-1} - \phi_2^{(m,j)}\tilde{z}_{t-2} - \ldots - \phi_m^{(m,j)}\tilde{z}_{t-m}$.

4. From the sample autocorrelations of the residuals, $r_k(w)$, approximate the standard error of the squared canonical correlation estimate by

$$var(\hat{\lambda}^*(m,j)^{1/2}) \approx d(m,j)/(n-m-j)$$

where $d(m,j) = (1 + 2\sum_{i=1}^{j-1} r_k(w^{(m,j)}))$.

The test statistic to be used as an identification criterion is

$$c(m,j) = -(n-m-j)\text{ln}(1 - \hat{\lambda}^*(m,j)/d(m,j))$$

which is asymptotically χ_1^2 if $m = p+d$ and $j \geq q$ or if $m \geq p+d$ and $j = q$. For $m > p$ and $j < q$, there is more than one theoretical zero canonical correlation between $Y_{m,t}$ and $Y_{m,t-j-1}$. Since the $\hat{\lambda}^*(m,j)$ are the smallest canonical correlations for each (m,j), the percentiles of $c(m,j)$ are less than those of a χ_1^2; therefore, Tsay and Tiao (1985) state that it is safe to assume a χ_1^2. For $m < p$ and $j < q$, no conclusions about the distribution of $c(m,j)$ are made.

A SCAN table is then constructed using $c(m,j)$ to determine which of the $\hat{\lambda}^*(m,j)$ are significantly different from zero (see Table 7.6). The ARMA orders are tentatively identified by finding a (maximal) rectangular pattern in which the $\hat{\lambda}^*(m,j)$ are insignificant for all test orders $m \geq p+d$ and $j \geq q$. There may be more than one pair of values $(p+d, q)$ that permit such a rectangular pattern. In this case, parsimony and the number of insignificant items in the rectangular pattern should help determine the model order. Table 7.7 depicts the theoretical pattern associated with an ARMA(2,2) series.

Table 7.6. SCAN Table

	MA					
AR	0	1	2	3	·	·
0	$c(0,0)$	$c(0,1)$	$c(0,2)$	$c(0,3)$	·	·
1	$c(1,0)$	$c(1,1)$	$c(1,2)$	$c(1,3)$	·	·
2	$c(2,0)$	$c(2,1)$	$c(2,2)$	$c(2,3)$	·	·
3	$c(3,0)$	$c(3,1)$	$c(3,2)$	$c(3,3)$	·	·
·	·	·	·	·	·	·
·	·	·	·	·	·	·

Table 7.7. Theoretical SCAN Table for an ARMA(2,2) Series

	MA							
AR	0	1	2	3	4	5	6	7
0	*	X	X	X	X	X	X	X
1	*	X	X	X	X	X	X	X
2	*	X	0	0	0	0	0	0
3	*	X	0	0	0	0	0	0
4	*	X	0	0	0	0	0	0
	X = significant terms 0 = insignificant terms * = no pattern							

Stationarity Tests

When a time series has a unit root, the series is nonstationary and the ordinary least squares (OLS) estimator is not normally distributed. Dickey (1976) and Dickey and Fuller (1979) studied the limiting distribution of the OLS estimator of autoregressive models for time series with a simple unit root. Dickey, Hasza and Fuller (1984) obtained the limiting distribution for time series with seasonal unit roots. Hamilton (1994) discusses the various types of unit root testing.

For a description of Dickey-Fuller tests, refer to , "PROBDF Function for Dickey-Fuller Tests" in Chapter 4.

Refer to Chapter 4, “SAS Macros and Functions,” for a description of the augmented Dickey-Fuller tests. Refer to Chapter 8, “The AUTOREG Procedure,” for a description of Phillips-Perron tests.

The random walk with drift test recommends whether or not an integrated times series has a drift term. Hamilton (1994) discusses this test.

Prewhitening

If, as is usually the case, an input series is autocorrelated, the direct cross-correlation function between the input and response series gives a misleading indication of the relation between the input and response series.

One solution to this problem is called *prewhitening*. You first fit an ARIMA model for the input series sufficient to reduce the residuals to white noise; then, filter the input series with this model to get the white noise residual series. You then filter the response series with the same model and cross correlate the filtered response with the filtered input series.

The ARIMA procedure performs this prewhitening process automatically when you precede the IDENTIFY statement for the response series with IDENTIFY and ESTIMATE statements to fit a model for the input series. If a model with no inputs was previously fit to a variable specified by the CROSSCORR= option, then that model is used to prewhiten both the input series and the response series before the cross correlations are computed for the input series.

For example,

```
proc arima data=in;
   identify var=x;
   estimate p=1 q=1;
   identify var=y crosscorr=x;
```

Both X and Y are filtered by the ARMA(1,1) model fit to X before the cross correlations are computed.

Note that prewhitening is done to estimate the cross-correlation function; the unfiltered series are used in any subsequent ESTIMATE or FORECAST statements, and the correlation functions of Y with its own lags are computed from the unfiltered Y series. But initial values in the ESTIMATE statement are obtained with prewhitened data; therefore, the result with prewhitening can be different from the result without prewhitening.

To suppress prewhitening for all input variables, use the CLEAR option on the IDENTIFY statement to make PROC ARIMA forget all previous models.

Prewhitening and Differencing

If the VAR= and CROSSCORR= options specify differencing, the series are differenced before the prewhitening filter is applied. When the differencing lists specified on the VAR= option for an input and on the CROSSCORR= option for that input are not the same, PROC ARIMA combines the two lists so that the differencing operators used for prewhitening include all differences in either list (in the least common multiple sense).

Identifying Transfer Function Models

When identifying a transfer function model with multiple input variables, the cross-correlation functions may be misleading if the input series are correlated with each other. Any dependencies among two or more input series will confound their cross correlations with the response series.

The prewhitening technique assumes that the input variables do not depend on past values of the response variable. If there is feedback from the response variable to an input variable, as evidenced by significant cross-correlation at negative lags, both the input and the response variables need to be prewhitened before meaningful cross correlations can be computed.

PROC ARIMA cannot handle feedback models. The STATESPACE procedure is more appropriate for models with feedback.

Missing Values and Autocorrelations

To compute the sample autocorrelation function when missing values are present, PROC ARIMA uses only cross products that do not involve missing values and employs divisors that reflect the number of cross products used rather than the total length of the series. Sample partial autocorrelations and inverse autocorrelations are then computed using the sample autocorrelation function. If necessary, a taper is employed to transform the sample autocorrelations into a positive definite sequence before calculating the partial autocorrelation and inverse correlation functions. The confidence intervals produced for these functions may not be valid when there are missing values. The distributional properties for sample correlation functions are not clear for finite samples. See Dunsmuir (1984) for some asymptotic properties of the sample correlation functions.

Estimation Details

The ARIMA procedure primarily uses the computational methods outlined by Box and Jenkins. Marquardt's method is used for the nonlinear least-squares iterations. Numerical approximations of the derivatives of the sum-of-squares function are taken using a fixed delta (controlled by the DELTA= option).

The methods do not always converge successfully for a given set of data, particularly if the starting values for the parameters are not close to the least-squares estimates.

Back-forecasting

The unconditional sum of squares is computed exactly; thus, back-forecasting is not performed. Early versions of SAS/ETS software used the back-forecasting approximation and allowed a positive value of the BACKLIM= option to control the extent of the back-forecasting. In the current version, requesting a positive number of back-forecasting steps with the BACKLIM= option has no effect.

Preliminary Estimation

If an autoregressive or moving-average operator is specified with no missing lags, preliminary estimates of the parameters are computed using the autocorrelations computed in the IDENTIFY stage. Otherwise, the preliminary estimates are arbitrarily set to values that produce stable polynomials.

When preliminary estimation is not performed by PROC ARIMA, then initial values of the coefficients for any given autoregressive or moving average factor are set to 0.1 if the degree of the polynomial associated with the factor is 9 or less. Otherwise, the coefficients are determined by expanding the polynomial $(1-0.1B)$ to an appropriate power using a recursive algorithm.

These preliminary estimates are the starting values in an iterative algorithm to compute estimates of the parameters.

Estimation Methods

Maximum Likelihood

The METHOD= ML option produces maximum likelihood estimates. The likelihood function is maximized via nonlinear least squares using Marquardt's method. Maximum likelihood estimates are more expensive to compute than the conditional least-squares estimates, however, they may be preferable in some cases (Ansley and Newbold 1980; Davidson 1981).

The maximum likelihood estimates are computed as follows. Let the univariate ARMA model be

$$\phi(B)(W_t - \mu_t) = \theta(B)a_t$$

where a_t is an independent sequence of normally distributed innovations with mean 0 and variance σ^2. Here μ_t is the mean parameter μ plus the transfer function inputs. The log likelihood function can be written as follows:

$$-\frac{1}{2\sigma^2}\mathbf{x}'\mathbf{\Omega}^{-1}\mathbf{x} - \frac{1}{2}\ln(|\mathbf{\Omega}|) - \frac{n}{2}\ln(\sigma^2)$$

In this equation, *n* is the number of observations, $\sigma^2\mathbf{\Omega}$ is the variance of **x** as a function of the ϕ and θ parameters, and $|\bullet|$ denotes the determinant. The vector **x** is the time series W_t minus the structural part of the model μ_t, written as a column vector, as follows:

$$\mathbf{x} = \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_n \end{bmatrix} - \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix}$$

The maximum likelihood estimate (MLE) of σ^2 is

$$s^2 = \frac{1}{n}\mathbf{x}'\mathbf{\Omega}^{-1}\mathbf{x}$$

Note that the default estimator of the variance divides by $n - r$, where *r* is the number of parameters in the model, instead of by *n*. Specifying the NODF option causes a divisor of *n* to be used.

The log likelihood concentrated with respect to σ^2 can be taken up to additive constants as

$$-\frac{n}{2}\ln(\mathbf{x}'\mathbf{\Omega}^{-1}\mathbf{x}) - \frac{1}{2}\ln(|\mathbf{\Omega}|)$$

Let **H** be the lower triangular matrix with positive elements on the diagonal such that $\mathbf{HH}' = \mathbf{\Omega}$. Let **e** be the vector $\mathbf{H}^{-1}\mathbf{x}$. The concentrated log likelihood with respect to σ^2 can now be written as

$$-\frac{n}{2}\ln(\mathbf{e}'\mathbf{e}) - \ln(|\mathbf{H}|)$$

or

$$-\frac{n}{2}\ln(|\mathbf{H}|^{1/n}\mathbf{e}'\mathbf{e}|\mathbf{H}|^{1/n})$$

The MLE is produced by using a Marquardt algorithm to minimize the following sum of squares:

$$|\mathbf{H}|^{1/n}\mathbf{e}'\mathbf{e}|\mathbf{H}|^{1/n}$$

The subsequent analysis of the residuals is done using **e** as the vector of residuals.

Unconditional Least Squares

The METHOD=ULS option produces unconditional least-squares estimates. The ULS method is also referred to as the *exact least-squares* (ELS) method. For METHOD=ULS, the estimates minimize

$$\sum_{t=1}^{n} \tilde{a}_t^2 = \sum_{t=1}^{n} (x_t - \mathbf{C}_t \mathbf{V}_t^{-1} (x_1, \cdots, x_{t-1})')^2$$

where $\mathbf{C}_t$ is the covariance matrix of x_t and $(x_1, \cdots, x_{t-1})$, and $\mathbf{V}_t$ is the variance matrix of $(x_1, \cdots, x_{t-1})$. In fact, $\sum_{t=1}^{n} \tilde{a}_t^2$ is the same as $\mathbf{x'\Omega^{-1}x}$ and, hence, $\mathbf{e'e}$. Therefore, the unconditional least-squares estimates are obtained by minimizing the sum of squared residuals rather than using the log likelihood as the criterion function.

Conditional Least Squares

The METHOD=CLS option produces conditional least-squares estimates. The CLS estimates are conditional on the assumption that the past unobserved errors are equal to 0. The series x_t can be represented in terms of the previous observations, as follows:

$$x_t = a_t + \sum_{i=1}^{\infty} \pi_i x_{t-i}$$

The π weights are computed from the ratio of the ϕ and θ polynomials, as follows:

$$\frac{\phi(B)}{\theta(B)} = 1 - \sum_{i=1}^{\infty} \pi_i B^i.$$

The CLS method produces estimates minimizing

$$\sum_{t=1}^{n} \hat{a}_t^2 = \sum_{t=1}^{n} (x_t - \sum_{i=1}^{\infty} \hat{\pi}_i x_{t-i})^2$$

where the unobserved past values of x_t are set to 0 and $\hat{\pi}_i$ are computed from the estimates of ϕ and θ at each iteration.

For METHOD=ULS and METHOD=ML, initial estimates are computed using the METHOD=CLS algorithm.

Start-up for Transfer Functions

When computing the noise series for transfer function and intervention models, the start-up for the transferred variable is done assuming that past values of the input series are equal to the first value of the series. The estimates are then obtained by applying least squares or maximum likelihood to the noise series. Thus, for transfer function models, the ML option does not generate the full (multivariate ARMA) maximum likelihood estimates, but it uses only the univariate likelihood function applied to the noise series.

Because PROC ARIMA uses all of the available data for the input series to generate the noise series, other start-up options for the transferred series can be implemented by prefixing an observation to the beginning of the real data. For example, if you fit a transfer function model to the variable Y with the single input X, then you can employ a start-up using 0 for the past values by prefixing to the actual data an observation with a missing value for Y and a value of 0 for X.

Information Criteria

PROC ARIMA computes and prints two information criteria, Akaike's information criterion (AIC) (Akaike 1974; Harvey 1981) and Schwarz's Bayesian criterion (SBC) (Schwarz 1978). The AIC and SBC are used to compare competing models fit to the same series. The model with the smaller information criteria is said to fit the data better. The AIC is computed as

$$-2\ln(L) + 2k$$

where L is the likelihood function and k is the number of free parameters. The SBC is computed as

$$-2\ln(L) + \ln(n)k$$

where n is the number of residuals that can be computed for the time series. Sometimes Schwarz's Bayesian criterion is called the Bayesian Information criterion (BIC).

If METHOD=CLS is used to do the estimation, an approximation value of L is used, where L is based on the conditional sum of squares instead of the exact sum of squares, and a Jacobian factor is left out.

Tests of Residuals

A table of test statistics for the hypothesis that the model residuals are white noise is printed as part of the ESTIMATE statement output. The chi-square statistics used in the test for lack of fit are computed using the Ljung-Box formula

$$\chi^2_m = n(n+2)\sum_{k=1}^{m}\frac{r_k^2}{(n-k)}$$

where

$$r_k = \frac{\sum_{t=1}^{n-k} a_t a_{t+k}}{\sum_{t=1}^{n} a_t^2}$$

and a_t is the residual series.

This formula has been suggested by Ljung and Box (1978) as yielding a better fit to the asymptotic chi-square distribution than the Box-Pierce Q statistic. Some simulation studies of the finite sample properties of this statistic are given by Davies, Triggs, and Newbold (1977) and by Ljung and Box (1978).

Each chi-square statistic is computed for all lags up to the indicated lag value and is not independent of the preceding chi-square values. The null hypotheses tested is that the current set of autocorrelations is white noise.

t-values

The *t* values reported in the table of parameter estimates are approximations whose accuracy depends on the validity of the model, the nature of the model, and the length of the observed series. When the length of the observed series is short and the number of estimated parameters is large with respect to the series length, the *t* approximation is usually poor. Probability values corresponding to a *t* distribution should be interpreted carefully as they may be misleading.

Cautions During Estimation

The ARIMA procedure uses a general nonlinear least-squares estimation method that can yield problematic results if your data do not fit the model. Output should be examined carefully. The GRID option can be used to ensure the validity and quality of the results. Problems you may encounter include the following:

- Preliminary moving-average estimates may not converge. Should this occur, preliminary estimates are derived as described previously in "Preliminary Estimation." You can supply your own preliminary estimates with the ESTIMATE statement options.
- The estimates can lead to an unstable time series process, which can cause extreme forecast values or overflows in the forecast.
- The Jacobian matrix of partial derivatives may be singular; usually, this happens because not all the parameters are identifiable. Removing some of the parameters or using a longer time series may help.
- The iterative process may not converge. PROC ARIMA's estimation method stops after *n* iterations, where *n* is the value of the MAXITER= option. If an iteration does not improve the SSE, the Marquardt parameter is increased by a factor of ten until parameters that have a smaller SSE are obtained or until the limit value of the Marquardt parameter is exceeded.
- For METHOD=CLS, the estimates may converge but not to least-squares estimates. The estimates may converge to a local minimum, the numerical calculations may be distorted by data whose sum-of-squares surface is not smooth, or the minimum may lie outside the region of invertibility or stationarity.
- If the data are differenced and a moving-average model is fit, the parameter estimates may try to converge exactly on the invertibility boundary. In this case, the standard error estimates that are based on derivatives may be inaccurate.

Specifying Inputs and Transfer Functions

Input variables and transfer functions for them may be specified using the INPUT= option on the ESTIMATE statement. The variables used on the INPUT= option must be included in the CROSSCORR= list in the previous IDENTIFY statement. If any differencing is specified in the CROSSCORR= list, then the differenced variable is used as the input to the transfer function.

General Syntax of the INPUT= Option

The general syntax of the INPUT= option is

ESTIMATE . . . **INPUT=***(transfer-function variable ...)*

The transfer function for an input variable is optional. The name of a variable by itself can be used to specify a pure regression term for the variable.

If specified, the syntax of the transfer function is

$$S\ \$\ (L_{1,1}, L_{1,2}, \ldots)(L_{2,1}, \ldots)\ldots/(L_{i,1}, L_{i,2}, \ldots)(L_{i+1,1}, \ldots)\ldots$$

S is the number of periods of time delay (lag) for this input series. Each term in parentheses specifies a polynomial factor with parameters at the lags specified by the $L_{i,j}$ values. The terms before the slash (/) are numerator factors. The terms after the slash (/) are denominator factors. All three parts are optional.

Commas can optionally be used between input specifications to make the INPUT= option more readable. The $ sign after the shift is also optional.

Except for the first numerator factor, each of the terms $L_{i,1}, L_{i,2}, \ldots, L_{i,k}$ indicates a factor of the form

$$(1 - \omega_{i,1}B^{L_{i,1}} - \omega_{i,2}B^{L_{i,2}} - \ldots - \omega_{i,k}B^{L_{i,k}})$$

The form of the first numerator factor depends on the ALTPARM option. By default, the constant 1 in the first numerator factor is replaced with a free parameter ω_0.

Alternative Model Parameterization

When the ALTPARM option is specified, the ω_0 parameter is factored out so it multiplies the entire transfer function, and the first numerator factor has the same form as the other factors.

The ALTPARM option does not materially affect the results; it just presents the results differently. Some people prefer to see the model written one way, while others prefer the alternative representation. Table 7.8 illustrates the effect of the ALTPARM option.

Table 7.8. The ALTPARM Option

INPUT= Option	ALTPARM	Model
INPUT=((1 2)(12)/(1)X);	No	$(\omega_0 - \omega_1 B - \omega_2 B^2)(1 - \omega_3 B^{12})/(1 - \delta_1 B)X_t$
	Yes	$\omega_0(1 - \omega_1 B - \omega_2 B^2)(1 - \omega_3 B^{12})/(1 - \delta_1 B)X_t$

Differencing and Input Variables

If you difference the response series and use input variables, take care that the differencing operations do not change the meaning of the model. For example, if you want to fit the model

$$Y_t = \frac{\omega_0}{(1-\delta_1 B)} X_t + \frac{(1-\theta_1 B)}{(1-B)(1-B^{12})} a_t$$

then the IDENTIFY statement must read

```
identify var=y(1,12) crosscorr=x(1,12);
estimate q=1 input=(/(1)x) noconstant;
```

If instead you specify the differencing as

```
identify var=y(1,12) crosscorr=x;
estimate q=1 input=(/(1)x) noconstant;
```

then the model being requested is

$$Y_t = \frac{\omega_0}{(1-\delta_1 B)(1-B)(1-B^{12})} X_t + \frac{(1-\theta_1 B)}{(1-B)(1-B^{12})} a_t$$

which is a very different model.

The point to remember is that a differencing operation requested for the response variable specified by the VAR= option is applied only to that variable and not to the noise term of the model.

Initial Values

The syntax for giving initial values to transfer function parameters in the the INITVAL= option parallels the syntax of the INPUT= option. For each transfer function in the INPUT= option, the INITVAL= option should give an initialization specification followed by the input series name. The initialization specification for each transfer function has the form

$$C \ \$ \ (V_{1,1}, V_{1,2}, \ldots)(V_{2,1}, \ldots)\ldots/(V_{i,1}, \ldots)\ldots$$

where C is the lag 0 term in the first numerator factor of the transfer function (or the overall scale factor if the ALTPARM option is specified), and $V_{i,j}$ is the coefficient of the $L_{i,j}$ element in the transfer function.

To illustrate, suppose you want to fit the model

$$Y_t = \mu + \frac{(\omega_0 - \omega_1 B - \omega_2 B^2)}{(1-\delta_1 B - \delta_2 B^2 - \delta_3 B^3)} X_{t-3} + \frac{1}{(1-\phi_1 B - \phi_2 B^3)} a_t$$

and start the estimation process with the initial values μ=10, ω_0=1, ω_1=.5, ω_2=.03, δ_1=.8, δ_2=-.1, δ_3=.002, ϕ_1=.1, ϕ_2=.01. (These are arbitrary values for illustration only.) You would use the following statements:

```
identify var=y crosscorr=x;
estimate p=(1,3) input=(3$(1,2)/(1,2,3)x)
         mu=10 ar=.1 .01 initval=(1$(.5,.03)/(.8,-.1,.002)x);
```

Note that the lags specified for a particular factor will be sorted, so initial values should be given in sorted order. For example, if the P= option had been entered as P=(3,1) instead of P=(1,3), the model would be the same and so would the AR= option. Sorting is done within all factors, including transfer function factors, so initial values should always be given in order of increasing lags.

Here is another illustration, showing initialization for a factored model with multiple inputs. The model is

$$\begin{aligned} Y_t = \mu \quad &+ \quad \frac{\omega_{1,0}}{(1-\delta_{1,1}B)}W_t + (\omega_{2,0}-\omega_{2,1}B)X_{t-3} \\ &+ \quad \frac{1}{(1-\phi_1 B)(1-\phi_2 B^6 - \phi_3 B^{12})}a_t \end{aligned}$$

and the initial values are μ=10, $\omega_{1,0}$=5, $\delta_{1,1}$=.8, $\omega_{2,0}$=1, $\omega_{2,1}$=.5, ϕ_1=.1, ϕ_2=.05, and ϕ_3=.01. You would use the following statements:

```
identify var=y crosscorr=(w x);
estimate p=(1)(6,12) input=(/(1)w, 3$(1)x)
         mu=10 ar=.1 .05 .01 initval=(5$/(.8)w 1$(.5)x);
```

Stationarity and Invertibility

By default PROC ARIMA requires that the parameter estimates for the AR and MA parts of the model always remain in the stationary and invertible regions, respectively. The NOSTABLE option removes this restriction and for high-order models may save some computer time. Note that using the NOSTABLE option does not necessarily result in an unstable model being fit, since the estimates may leave the stable region for some iterations, but still ultimately converge to stable values.

Naming of Model Parameters

In the table of parameter estimates produced by the ESTIMATE statement, model parameters are referred to using the naming convention described in this section.

The parameters in the noise part of the model are named as AR*i,j* or MA*i,j*, where AR refers to autoregressive parameters and MA to moving-average parameters. The subscript *i* refers to the particular polynomial factor, and the subscript *j* refers to the *j*th term within the *i*th factor. These terms are sorted in order of increasing lag within factors, so the subscript *j* refers to the *j*th term after sorting.

When inputs are used in the model, the parameters of each transfer function are named NUM*i*,*j* and DEN*i*,*j*. The *j*th term in the *i*th factor of a numerator polynomial is named NUM*i*,*j*. The *j*th term in the *i*th factor of a denominator polynomial is named DEN*i*,*j*.

This naming process is repeated for each input variable, so if there are multiple inputs, parameters in transfer functions for different input series have the same name. The table of parameter estimates shows in the "Variable" column the input with which each parameter is associated. The parameter name shown in the "Parameter" column and the input variable name shown in the "Variable" column must be combined to fully identify transfer function parameters.

The lag 0 parameter in the first numerator factor for the first input variable is named NUM1. For subsequent input variables, the lag 0 parameter in the first numerator factor is named NUM*k*, where *k* is the position of the input variable in the INPUT= option list. If the ALTPARM option is specified, the NUM*k* parameter is replaced by an overall scale parameter named SCALE*k*.

For the mean and noise process parameters, the response series name is shown in the "Variable" column. The "Lag" and "Shift" for each parameter are also shown in the table of parameter estimates when inputs are used.

Missing Values and Estimation and Forecasting

Estimation and forecasting are carried out in the presence of missing values by forecasting the missing values with the current set of parameter estimates. The maximum likelihood algorithm employed was suggested by Jones (1980) and is used for both unconditional least-squares (ULS) and conditional least-squares (CLS) estimation.

The CLS algorithm simply fills in missing values with infinite memory forecast values, computed by forecasting ahead from the nonmissing past values as far as required by the structure of the missing values. These artificial values are then employed in the nonmissing value CLS algorithm. Artificial values are updated at each iteration along with parameter estimates.

For models with input variables, embedded missing values (that is, missing values other than at the beginning or end of the series) are not generally supported. Embedded missing values in input variables are supported for the special case of a multiple regression model having ARIMA errors. A multiple regression model is specified by an INPUT= option that simply lists the input variables (possibly with lag shifts) without any numerator or denominator transfer function factors. One-step-ahead forecasts are not available for the response variable when one or more of the input variables have missing values.

When embedded missing values are present for a model with complex transfer functions, PROC ARIMA uses the first continuous nonmissing piece of each series to do the analysis. That is, PROC ARIMA skips observations at the beginning of each series until it encounters a nonmissing value and then uses the data from there until it encounters another missing value or until the end of the data is reached. This makes the current version of PROC ARIMA compatible with earlier releases that did not allow embedded missing values.

Forecasting Details

If the model has input variables, a forecast beyond the end of the data for the input variables is possible only if univariate ARIMA models have previously been fit to the input variables or future values for the input variables are included in the DATA= data set.

If input variables are used, the forecast standard errors and confidence limits of the response depend on the estimated forecast error variance of the predicted inputs. If several input series are used, the forecast errors for the inputs should be independent; otherwise, the standard errors and confidence limits for the response series will not be accurate. If future values for the input variables are included in the DATA= data set, the standard errors of the forecasts will be underestimated since these values are assumed to be known with certainty.

The forecasts are generated using forecasting equations consistent with the method used to estimate the model parameters. Thus, the estimation method specified on the ESTIMATE statement also controls the way forecasts are produced by the FORECAST statement.

Infinite Memory Forecasts

If METHOD=CLS is used, the forecasts are *infinite memory forecasts*, also called *conditional forecasts*. The term *conditional* is used because the forecasts are computed by assuming that the unknown values of the response series before the start of the data are equal to the mean of the series. Thus, the forecasts are conditional on this assumption.

The series x_t can be represented as

$$x_t = a_t + \sum_{i=1}^{\infty} \pi_i x_{t-i}$$

where $\phi(B)/\theta(B) = 1 - \sum_{i=1}^{\infty} \pi_i B^i$.

The *k*-step forecast of x_{t+k} is computed as

$$\hat{x}_{t+k} = \sum_{i=1}^{k-1} \hat{\pi}_i \hat{x}_{t+k-i} + \sum_{i=k}^{\infty} \hat{\pi}_i x_{t+k-i}$$

where unobserved past values of x_t are set to zero, and $\hat{\pi}_i$ is obtained from the estimated parameters $\hat{\phi}$ and $\hat{\theta}$.

Finite Memory Forecasts

For METHOD=ULS or METHOD=ML, the forecasts are *finite memory forecasts*, also called *unconditional forecasts*. For finite memory forecasts, the covariance function of the ARMA model is used to derive the best linear prediction equation.

That is, the k-step forecast of x_{t+k}, given $(x_1, \cdots, x_{t-1})$, is

$$\tilde{x}_{t+k} = \mathbf{C}_{k,t} \mathbf{V}_t^{-1} (x_1, \cdots, x_{t-1})'$$

where $\mathbf{C}_{k,t}$ is the covariance of x_{t+k} and $(x_1, \cdots, x_{t-1})$, and $\mathbf{V}_t$ is the covariance matrix of the vector $(x_1, \cdots, x_{t-1})$. $\mathbf{C}_{k,t}$ and $\mathbf{V}_t$ are derived from the estimated parameters.

Finite memory forecasts minimize the mean-squared error of prediction if the parameters of the ARMA model are known exactly. (In most cases, the parameters of the ARMA model are estimated, so the predictors are not true best linear forecasts.)

If the response series is differenced, the final forecast is produced by summing the forecast of the differenced series. This summation, and thus the forecast, is conditional on the initial values of the series. Thus, when the response series is differenced, the final forecasts are not true finite memory forecasts because they are derived assuming that the differenced series begins in a steady-state condition. Thus, they fall somewhere between finite memory and infinite memory forecasts. In practice, there is seldom any practical difference between these forecasts and true finite memory forecasts.

Forecasting Log Transformed Data

The log transformation is often used to convert time series that are nonstationary with respect to the innovation variance into stationary time series. The usual approach is to take the log of the series in a DATA step and then apply PROC ARIMA to the transformed data. A DATA step is then used to transform the forecasts of the logs back to the original units of measurement. The confidence limits are also transformed using the exponential function.

As one alternative, you can simply exponentiate the forecast series. This procedure gives a forecast for the median of the series, but the antilog of the forecast log series underpredicts the mean of the original series. If you want to predict the expected value of the series, you need to take into account the standard error of the forecast, as shown in the following example, which uses an AR(2) model to forecast the log of a series Y:

```
data in;
   set in;
   ylog = log( y );
run;

proc arima data=in;
   identify var=ylog;
   estimate p=2;
   forecast lead=10 out=out;
run;

data out;
   set out;
   y   = exp( ylog );
```

```
      l95 = exp( l95 );
      u95 = exp( u95 );
      forecast = exp( forecast + std*std/2 );
   run;
```

Specifying Series Periodicity

The INTERVAL= option is used together with the ID= variable to describe the observations that make up the time series. For example, INTERVAL=MONTH specifies a monthly time series in which each observation represents one month. See Chapter 3, “Date Intervals, Formats, and Functions,” for details on the interval values supported.

The variable specified by the ID= option in the PROC ARIMA statement identifies the time periods associated with the observations. Usually, SAS date or datetime values are used for this variable. PROC ARIMA uses the ID= variable in the following ways:

- to validate the data periodicity. When the INTERVAL= option is specified, PROC ARIMA uses the ID variable to check the data and verify that successive observations have valid ID values corresponding to successive time intervals. When the INTERVAL= option is not used, PROC ARIMA verifies that the ID values are nonmissing and in ascending order.
- to check for gaps in the input observations. For example, if INTERVAL=MONTH and an input observation for April 1970 follows an observation for January 1970, there is a gap in the input data with two omitted observations (namely February and March 1970). A warning message is printed when a gap in the input data is found.
- to label the forecast observations in the output data set. PROC ARIMA extrapolates the values of the ID variable for the forecast observations from the ID value at the end of the input data according to the frequency specifications of the INTERVAL= option. If the INTERVAL= option is not specified, PROC ARIMA extrapolates the ID variable by incrementing the ID variable value for the last observation in the input data by 1 for each forecast period. Values of the ID variable over the range of the input data are copied to the output data set.

The ALIGN= option is used to align the ID variable to the beginning, middle or end of the time ID interval specified by the INTERVAL= option.

OUT= Data Set

The output data set produced by the OUT= option of the PROC ARIMA or FORECAST statements contains the following:

- the BY variables
- the ID variable

- the variable specified by the VAR= option in the IDENTIFY statement, which contains the actual values of the response series
- FORECAST, a numeric variable containing the one-step-ahead predicted values and the multistep forecasts
- STD, a numeric variable containing the standard errors of the forecasts
- a numeric variable containing the lower confidence limits of the forecast. This variable is named L95 by default but has a different name if the ALPHA= option specifies a different size for the confidence limits.
- RESIDUAL, a numeric variable containing the differences between actual and forecast values
- a numeric variable containing the upper confidence limits of the forecast. This variable is named U95 by default but has a different name if the ALPHA= option specifies a different size for the confidence limits.

The ID variable, the BY variables, and the time series variable are the only ones copied from the input to the output data set.

Unless the NOOUTALL option is specified, the data set contains the whole time series. The FORECAST variable has the one-step forecasts (predicted values) for the input periods, followed by *n* forecast values, where *n* is the LEAD= value. The actual and RESIDUAL values are missing beyond the end of the series.

If you specify the same OUT= data set on different FORECAST statements, the latter FORECAST statements overwrite the output from the previous FORECAST statements. If you want to combine the forecasts from different FORECAST statements in the same output data set, specify the OUT= option once on the PROC ARIMA statement and omit the OUT= option on the FORECAST statements.

When a global output data set is created by the OUT= option in the PROC ARIMA statement, the variables in the OUT= data set are defined by the first FORECAST statement that is executed. The results of subsequent FORECAST statements are vertically concatenated onto the OUT= data set. Thus, if no ID variable is specified in the first FORECAST statement that is executed, no ID variable appears in the output data set, even if one is specified in a later FORECAST statement. If an ID variable is specified in the first FORECAST statement that is executed but not in a later FORECAST statement, the value of the ID variable is the same as the last value processed for the ID variable for all observations created by the later FORECAST statement. Furthermore, even if the response variable changes in subsequent FORECAST statements, the response variable name in the output data set will be that of the first response variable analyzed.

OUTCOV= Data Set

The output data set produced by the OUTCOV= option of the IDENTIFY statement contains the following variables:

- LAG, a numeric variable containing the lags corresponding to the values of the covariance variables. The values of LAG range from 0 to N for covariance functions and from -N to N for cross-covariance functions, where N is the value of the NLAG= option.

- VAR, a character variable containing the name of the variable specified by the VAR= option.
- CROSSVAR, a character variable containing the name of the variable specified in the CROSSCORR= option, which labels the different cross-covariance functions. The CROSSVAR variable is blank for the autocovariance observations. When there is no CROSSCORR= option, this variable is not created.
- N, a numeric variable containing the number of observations used to calculate the current value of the covariance or cross-covariance function.
- COV, a numeric variable containing the autocovariance or cross-covariance function values. COV contains the autocovariances of the VAR= variable when the value of the CROSSVAR variable is blank. Otherwise COV contains the cross covariances between the VAR= variable and the variable named by the CROSSVAR variable.
- CORR, a numeric variable containing the autocorrelation or cross-correlation function values. CORR contains the autocorrelations of the VAR= variable when the value of the CROSSVAR variable is blank. Otherwise CORR contains the cross correlations between the VAR= variable and the variable named by the CROSSVAR variable.
- STDERR, a numeric variable containing the standard errors of the autocorrelations. The standard error estimate is based on the hypothesis that the process generating the time series is a pure moving-average process of order LAG-1. For the cross correlations, STDERR contains the value $1/\sqrt{n}$, which approximates the standard error under the hypothesis that the two series are uncorrelated.
- INVCORR, a numeric variable containing the inverse autocorrelation function values of the VAR= variable. For cross-correlation observations, (that is, when the value of the CROSSVAR variable is not blank), INVCORR contains missing values.
- PARTCORR, a numeric variable containing the partial autocorrelation function values of the VAR= variable. For cross-correlation observations (that is, when the value of the CROSSVAR variable is not blank), PARTCORR contains missing values.

OUTEST= Data Set

PROC ARIMA writes the parameter estimates for a model to an output data set when the OUTEST= option is specified in the ESTIMATE statement. The OUTEST= data set contains the following:

- the BY variables
- _NAME_, a character variable containing the name of the parameter for the covariance or correlation observations, or blank for the observations containing

the parameter estimates. (This variable is not created if neither OUTCOV nor OUTCORR is specified.)

- _TYPE_, a character variable that identifies the type of observation. A description of the _TYPE_ variable values is given below.
- variables for model parameters

The variables for the model parameters are named as follows:

ERRORVAR	This numeric variable contains the variance estimate. The _TYPE_=EST observation for this variable contains the estimated error variance, and the remaining observations are missing.
MU	This numeric variable contains values for the mean parameter for the model. (This variable is not created if NOCONSTANT is specified.)
MA*j_k*	These numeric variables contain values for the moving average parameters. The variables for moving average parameters are named MA*j_k*, where *j* is the factor number, and *k* is the index of the parameter within a factor.
AR*j_k*	These numeric variables contain values for the autoregressive parameters. The variables for autoregressive parameters are named AR*j_k*, where *j* is the factor number, and *k* is the index of the parameter within a factor.
I*j_k*	These variables contain values for the transfer function parameters. Variables for transfer function parameters are named I*j_k*, where *j* is the number of the INPUT variable associated with the transfer function component, and *k* is the number of the parameter for the particular INPUT variable. INPUT variables are numbered according to the order in which they appear in the INPUT= list.
STATUS	This variable describes the convergence status of the model. A value of 0_CONVERGED indicates that the model converged.

The value of the _TYPE_ variable for each observation indicates the kind of value contained in the variables for model parameters for the observation. The OUTEST= data set contains observations with the following _TYPE_ values:

EST	the observation contains parameter estimates
STD	the observation contains approximate standard errors of the estimates
CORR	the observation contains correlations of the estimates. OUTCORR must be specified to get these observations.
COV	the observation contains covariances of the estimates. OUTCOV must be specified to get these observations.

FACTOR	the observation contains values that identify for each parameter the factor that contains it. Negative values indicate denominator factors in transfer function models.
LAG	the observation contains values that identify the lag associated with each parameter
SHIFT	the observation contains values that identify the shift associated with the input series for the parameter

The values given for _TYPE_=FACTOR, _TYPE_=LAG, or _TYPE_=SHIFT observations enable you to reconstruct the model employed when provided with only the OUTEST= data set.

OUTEST= Examples

This section clarifies how model parameters are stored in the OUTEST= data set with two examples.

Consider the following example:

```
proc arima data=input;
   identify var=y cross=(x1 x2);
   estimate p=(1)(6) q=(1,3)(12) input=(x1 x2) outest=est;
quit;
proc print data=est;
run;
```

The model specified by these statements is

$$Y_t = \mu + \omega_{1,0}X_{1,t} + \omega_{2,0}X_{2,t} + \frac{(1-\theta_{11}B-\theta_{12}B^3)(1-\theta_{21}B^{12})}{(1-\phi_{11}B)(1-\phi_{21}B^6)}a_t$$

The OUTEST= data set contains the values shown in Table 7.9.

Table 7.9. OUTEST= Data Set for First Example

Obs	_TYPE_	Y	MU	MA1_1	MA1_2	MA2_1	AR1_1	AR2_1	I1_1	I2_1
1	EST	σ^2	μ	θ_{11}	θ_{12}	θ_{21}	ϕ_{11}	ϕ_{21}	$\omega_{1,0}$	$\omega_{2,0}$
2	STD	.	se μ	se θ_{11}	se θ_{12}	se θ_{21}	se ϕ_{11}	se ϕ_{21}	se $\omega_{1,0}$	se $\omega_{2,0}$
3	FACTOR	.	0	1	1	2	1	2	1	1
4	LAG	.	0	1	3	12	1	6	0	0
5	SHIFT	.	0	0	0	0	0	0	0	0

Note that the symbols in the rows for _TYPE_=EST and _TYPE_=STD in Table 7.9 would be numeric values in a real data set.

Next, consider the following example:

```
proc arima data=input;
   identify var=y cross=(x1(2) x2(1));
   estimate p=1 q=1 input=(2 $ (1)/(1,2)x1 1 $ /(1)x2) outest=est;
quit;
proc print data=est;
run;
```

The model specified by these statements is

$$Y_t = \mu + \frac{\omega_{10} - \omega_{11}B}{1 - \delta_{11}B - \delta_{12}B^2} X_{1,t-2} + \frac{\omega_{20}}{1 - \delta_{21}B} X_{2,t-1} + \frac{(1 - \theta_1 B)}{(1 - \phi_1 B)} a_t$$

The OUTEST= data set contains the values shown in Table 7.10.

Table 7.10. OUTEST= Data Set for Second Example

Obs	_TYPE_	Y	MU	MA1_1	AR1_1	I1_1	I1_2	I1_3	I1_4	I2_1	I2_2
1	EST	σ^2	μ	θ_1	ϕ_1	ω_{10}	ω_{11}	δ_{11}	δ_{12}	ω_{20}	δ_{21}
2	STD	.	se μ	se θ_1	se ϕ_1	se ω_{10}	se ω_{11}	se δ_{11}	se δ_{12}	se ω_{20}	se δ_{21}
3	FACTOR	.	0	1	1	1	1	-1	-1	1	-1
4	LAG	.	0	1	1	0	1	1	2	0	1
5	SHIFT	.	0	0	0	2	2	2	2	1	1

OUTMODEL= Data Set

The OUTMODEL= option in the ESTIMATE statement writes an output data set that enables you to reconstruct the model. The OUTMODEL= data set contains much the same information as the OUTEST= data set but in a transposed form that may be more useful for some purposes. In addition, the OUTMODEL= data set includes the differencing operators.

The OUTMODEL data set contains the following:

- the BY variables
- _NAME_, a character variable containing the name of the response or input variable for the observation.
- _TYPE_, a character variable that contains the estimation method that was employed. The value of _TYPE_ can be CLS, ULS, or ML.

STATUS This variable describes the convergence status of the model. A value of 0_CONVERGED indicates that the model converged.

- _PARM_, a character variable containing the name of the parameter given by the observation. _PARM_ takes on the values ERRORVAR, MU, AR, MA, NUM, DEN, and DIF.
- _VALUE_, a numeric variable containing the value of the estimate defined by the _PARM_ variable.
- _STD_, a numeric variable containing the standard error of the estimate.
- _FACTOR_, a numeric variable indicating the number of the factor to which the parameter belongs.
- _LAG_, a numeric variable containing the number of the term within the factor containing the parameter.
- _SHIFT_, a numeric variable containing the shift value for the input variable associated with the current parameter.

The values of _FACTOR_ and _LAG_ identify which particular MA, AR, NUM, or DEN parameter estimate is given by the _VALUE_ variable. The _NAME_ variable contains the response variable name for the MU, AR, or MA parameters. Otherwise, _NAME_ contains the input variable name associated with NUM or DEN parameter estimates. The _NAME_ variable contains the appropriate variable name associated with the current DIF observation as well. The _VALUE_ variable is 1 for all DIF observations, and the _LAG_ variable indicates the degree of differencing employed.

The observations contained in the OUTMODEL= data set are identified by the _PARM_ variable. A description of the values of the _PARM_ variable follows:

NUMRESID	_VALUE_ contains the number of residuals.
NPARMS	_VALUE_ contains the number of parameters in the model.
NDIFS	_VALUE_ contains the sum of the differencing lags employed for the response variable.
ERRORVAR	_VALUE_ contains the estimate of the innovation variance.
MU	_VALUE_ contains the estimate of the mean term.
AR	_VALUE_ contains the estimate of the autoregressive parameter indexed by the _FACTOR_ and _LAG_ variable values.
MA	_VALUE_ contains the estimate of a moving average parameter indexed by the _FACTOR_ and _LAG_ variable values.
NUM	_VALUE_ contains the estimate of the parameter in the numerator factor of the transfer function of the input variable indexed by the _FACTOR_, _LAG_, and _SHIFT_ variable values.
DEN	_VALUE_ contains the estimate of the parameter in the denominator factor of the transfer function of the input variable indexed by the _FACTOR_, _LAG_, and _SHIFT_ variable values.
DIF	_VALUE_ contains the difference operator defined by the difference lag given by the value in the _LAG_ variable.

OUTSTAT= Data Set

PROC ARIMA writes the diagnostic statistics for a model to an output data set when the OUTSTAT= option is specified in the ESTIMATE statement. The OUTSTAT data set contains the following:

- the BY variables.
- _TYPE_, a character variable that contains the estimation method used. _TYPE_ can have the value CLS, ULS, or ML.
- _STAT_, a character variable containing the name of the statistic given by the _VALUE_ variable in this observation. _STAT_ takes on the values AIC, SBC, LOGLIK, SSE, NUMRESID, NPARMS, NDIFS, ERRORVAR, MU, CONV, and NITER.
- _VALUE_, a numeric variable containing the value of the statistic named by the _STAT_ variable.

The observations contained in the OUTSTAT= data set are identified by the _STAT_ variable. A description of the values of the _STAT_ variable follows:

AIC	Akaike's information criterion
SBC	Schwarz's Bayesian criterion
LOGLIK	the log likelihood, if METHOD=ML or METHOD=ULS is specified
SSE	the sum of the squared residuals
NUMRESID	the number of residuals
NPARMS	the number of parameters in the model
NDIFS	the sum of the differencing lags employed for the response variable
ERRORVAR	the estimate of the innovation variance
MU	the estimate of the mean term
CONV	tells if the estimation converged
NITER	the number of iterations

Printed Output

The ARIMA procedure produces printed output for each of the IDENTIFY, ESTIMATE, and FORECAST statements. The output produced by each ARIMA statement is described in the following sections.

IDENTIFY Statement Printed Output

The printed output of the IDENTIFY statement consists of the following:

1. a table of summary statistics, including the name of the response variable, any specified periods of differencing, the mean and standard deviation of the response series after differencing, and the number of observations after differencing

2. a plot of the sample autocorrelation function for lags up to and including the NLAG= option value. Standard errors of the autocorrelations also appear to the right of the autocorrelation plot if the value of LINESIZE= option is sufficiently large. The standard errors are derived using Bartlett's approximation (Box and Jenkins 1976, p. 177). The approximation for a standard error for the estimated autocorrelation function at lag *k* is based on a null hypothesis that a pure moving-average Gaussian process of order *k*-1 generated the time series. The relative position of an approximate 95% confidence interval under this null hypothesis is indicated by the dots in the plot, while the asterisks represent the relative magnitude of the autocorrelation value.

3. a plot of the sample inverse autocorrelation function. See the section "The Inverse Autocorrelation Function" for more information on the inverse autocorrelation function.

4. a plot of the sample partial autocorrelation function

5. a table of test statistics for the hypothesis that the series is white noise. These test statistics are the same as the tests for white noise residuals produced by the ESTIMATE statement and are described in the section "Estimation Details" earlier in this chapter.

6. if the CROSSCORR= option is used, a plot of the sample cross-correlation function for each series specified in the CROSSCORR= option. If a model was previously estimated for a variable in the CROSSCORR= list, the cross correlations for that series are computed for the prewhitened input and response series. For each input variable with a prewhitening filter, the cross-correlation report for the input series includes

   (a) a table of test statistics for the hypothesis of no cross correlation between the input and response series
   (b) the prewhitening filter used for the prewhitening transformation of the predictor and response variables

7. if the ESACF option is used, ESACF tables are printed

8. if the MINIC option is used, a MINIC table is printed

9. if the SCAN option is used, SCAN table is printed

10. if the STATIONARITY option is used, STATIONARITY tests results are printed

ESTIMATE Statement Printed Output

The printed output of the ESTIMATE statement consists of the following:

1. when the PRINTALL option is specified, the preliminary parameter estimates and an iteration history showing the sequence of parameter estimates tried during the fitting process

2. a table of parameter estimates showing the following for each parameter: the parameter name, the parameter estimate, the approximate standard error, t value, approximate probability ($Pr > |t|$), the lag for the parameter, the input variable name for the parameter, and the lag or "Shift" for the input variable

3. the estimates of the constant term, the innovation variance (Variance Estimate), the innovation standard deviation (Std Error Estimate), Akaike's information criterion (AIC), Schwarz's Bayesian criterion (SBC), and the number of residuals

4. the correlation matrix of the parameter estimates

5. a table of test statistics for hypothesis that the residuals of the model are white noise titled "Autocorrelation Check of Residuals"

6. if the PLOT option is specified, autocorrelation, inverse autocorrelation, and partial autocorrelation function plots of the residuals

7. if an INPUT variable has been modeled in such a way that prewhitening is performed in the IDENTIFY step, a table of test statistics titled "Crosscorrelation Check of Residuals." The test statistic is based on the chi-square approximation suggested by Box and Jenkins (1976, pp. 395–396). The cross-correlation function is computed using the residuals from the model as one series and the prewhitened input variable as the other series.

8. if the GRID option is specified, the sum-of-squares or likelihood surface over a grid of parameter values near the final estimates

9. a summary of the estimated model showing the autoregressive factors, moving average factors, and transfer function factors in back shift notation with the estimated parameter values.

FORECAST Statement Printed Output

The printed output of the FORECAST statement consists of the following:

1. a summary of the estimated model

2. a table of forecasts, with columns for the observation numbers (Obs), the forecast values (Forecast), the forecast standard errors (Std Error), lower and upper limits of the approximate 95% confidence interval (95% confidence limits). The ALPHA= option can be used to change the confidence interval for forecasts. If the PRINTALL option is specified, the forecast table also includes columns for the actual values of the response series (Actual) and the residual values (Residual), and the table includes the input observations used to estimate the model.

ODS Table Names

PROC ARIMA assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 6, “Using the Output Delivery System.”

Table 7.11. ODS Tables Produced in PROC ARIMA

ODS Table Name	Description	Option
ODS Tables Created by the IDENTIFY Statement		
DescStats	Descriptive Statistics	
InputDescStats	Input Descriptive Statistics	
CorrGraph	Correlations graph	
StationarityTests	Stationarity tests	STATIONARITY option
TentativeOrders	Tenative Order Selection	MINIC, ESACF, or SCAN option
PACFGraph	Partial autocorrelations graph	

Table 7.11. (continued)

ODS Table Name	Description	Option
IACFGraph	Inverse autocorrelations graph	
ChiSqAuto	Chi-Square statistics table for autocorrelation	
ChiSqCross	Chi-Square statistics table for cross-correlations	CROSSCORR=
MINIC	Minimum Information Criterion	MINIC option
ESACF	Extended Sample Autocorrelation Function	ESACF option
ESACFPValues	ESACF Probability Values	ESACF option
SCAN	Squared Canonical Correlation Estimates	SCAN option
SCANPValues	SCAN Chi-Square[1] Probability Values	SCAN option
ODS Tables Created by the ESTIMATE Statement		
FitStatistics	Fit Statistics	
ARPolynomial	Filter Equations	
MAPolynomial	Filter Equations	
NumPolynomial	Filter Equations	
DenPolynomial	Filter Equations	
ParameterEstimates	ParameterEstimates	
ChiSqAuto	Chi-Square statistics table for autocorrelation	
ChiSqCross	Chi-Square statistics table for cross-correlations	
InitialAREstimates	Initial autoregressive parameter estimates	
InitialMAEstimates	Initial moving average parameter estimates	
PrelimEstimates	Preliminary Estimation	
IterHistory	Conditional Least Squares Estimation	METHOD=CLS
OptSummary	ARIMA Estimation Optimization	PRINTALL option
ModelDescription	Model description	
InputDescription	Input description	
ObjectiveGrid	Objective function grid matrix	GRID option
CorrB	Correlations of the Estimates	
ODS Tables Created by the FORECAST Statement		
Forecasts	Forecast	

Examples

Example 7.1. Simulated IMA Model

This example illustrates the ARIMA procedure results for a case where the true model is known. An integrated moving average model is used for this illustration.

The following DATA step generates a pseudo-random sample of 100 periods from the ARIMA(0,1,1) process $u_t = u_{t-1} + a_t - .8a_{t-1}$, a_t iid $\mathrm{N}(0, 1)$.

```
title1 'Simulated IMA(1,1) Series';
data a;
  u1 = 0.9; a1 = 0;
  do i = -50 to 100;
     a = rannor( 32565 );
     u = u1 + a - .8 * a1;
     if i > 0 then output;
     a1 = a;
     u1 = u;
     end;
run;
```

The following ARIMA procedure statements identify and estimate the model.

```
proc arima data=a;
  identify var=u nlag=15;
  run;
  identify var=u(1) nlag=15;
  run;
  estimate q=1 ;
  run;
quit;
```

The results of the first IDENTIFY statement are shown in Output 7.1.1. The output shows the behavior of the sample autocorrelation function when the process is nonstationary. Note that in this case the estimated autocorrelations are not very high, even at small lags. Nonstationarity is reflected in a pattern of significant autocorrelations that do not decline quickly with increasing lag, not in the size of the autocorrelations.

Output 7.1.1. Output from the First IDENTIFY Statement

```
                           Simulated IMA(1,1) Series

                              The ARIMA Procedure

                              Name of Variable = u

                       Mean of Working Series    0.099637
                       Standard Deviation        1.115604
                       Number of Observations         100


                                Autocorrelations

Lag   Covariance   Correlation   -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

  0     1.244572       1.00000   |                    |********************|
  1     0.547457       0.43988   |                .   |*********           |
  2     0.534787       0.42970   |               .    |*********           |
  3     0.569849       0.45787   |               .    |*********           |
  4     0.384428       0.30888   |              .     |******              |
  5     0.405137       0.32552   |              .     |*******             |
  6     0.253617       0.20378   |              .     |**** .              |
  7     0.321830       0.25859   |             .      |***** .             |
  8     0.363871       0.29237   |             .      |******.             |
  9     0.271180       0.21789   |             .      |****  .             |
 10     0.419208       0.33683   |             .      |*******             |
 11     0.298127       0.23954   |             .      |***** .             |
 12     0.186460       0.14982   |             .      |***   .             |
 13     0.313270       0.25171   |             .      |***** .             |
 14     0.314594       0.25277   |            .       |*****  .            |
 15     0.156329       0.12561   |            .       |***    .            |

                         "." marks two standard errors
                       The ARIMA Procedure

                     Inverse Autocorrelations

Lag    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

  1       -0.12382    |                . **|   .                |
  2       -0.17396    |                .***|   .                |
  3       -0.19966    |                ****|   .                |
  4       -0.01476    |                .   |   .                |
  5       -0.02895    |                .  *|   .                |
  6        0.20612    |                .   |****                |
  7        0.01258    |                .   |   .                |
  8       -0.09616    |                . **|   .                |
  9        0.00025    |                .   |   .                |
 10       -0.16879    |                .***|   .                |
 11        0.05680    |                .   |*  .                |
 12        0.14306    |                .   |***.                |
 13       -0.02466    |                .   |   .                |
 14       -0.15549    |                .***|   .                |
 15        0.08247    |                .   |** .                |


                     Partial Autocorrelations

Lag    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

  1        0.43988    |                .   |*********           |
  2        0.29287    |                .   |******              |
  3        0.26499    |                .   |*****               |
  4       -0.00728    |                .   |   .                |
  5        0.06473    |                .   |*  .                |
  6       -0.09926    |                . **|   .                |
  7        0.10048    |                .   |** .                |
  8        0.12872    |                .   |***.                |
  9        0.03286    |                .   |*  .                |
 10        0.16034    |                .   |***.                |
 11       -0.03794    |                .  *|   .                |
 12       -0.14469    |                .***|   .                |
 13        0.06415    |                .   |*  .                |
 14        0.15482    |                .   |***.                |
 15       -0.10989    |                . **|   .                |
```

```
                           The ARIMA Procedure

                   Autocorrelation Check for White Noise

 To      Chi-         Pr >
Lag    Square   DF   ChiSq   ---------------Autocorrelations---------------

  6     87.22    6  <.0001   0.440   0.430   0.458   0.309   0.326   0.204
 12    131.39   12  <.0001   0.259   0.292   0.218   0.337   0.240   0.150
```

The second IDENTIFY statement differences the series. The results of the second IDENTIFY statement are shown in Output 7.1.2. This output shows autocorrelation, inverse autocorrelation, and partial autocorrelation functions typical of MA(1) processes.

Output 7.1.2. Output from the Second IDENTIFY Statement

```
                              The ARIMA Procedure

                             Name of Variable = u

             Period(s) of Differencing                           1
             Mean of Working Series                       0.019752
             Standard Deviation                           1.160921
             Number of Observations                             99
             Observation(s) eliminated by differencing           1


                                Autocorrelations

Lag   Covariance   Correlation   -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

  0     1.347737       1.00000   |                    |********************|
  1    -0.699404       -.51895   |          **********|   .                |
  2    -0.036142       -.02682   |               .   *|    .               |
  3     0.245093       0.18186   |               .    |****.               |
  4    -0.234167       -.17375   |               . ***|    .               |
  5     0.181778       0.13488   |               .    |*** .               |
  6    -0.184601       -.13697   |               . ***|    .               |
  7    0.0088659       0.00658   |               .    |    .               |
  8     0.146372       0.10861   |               .    |**  .               |
  9    -0.241579       -.17925   |               .****|    .               |
 10     0.240512       0.17846   |               .    |****.               |
 11     0.031005       0.02301   |              .     |     .              |
 12    -0.250954       -.18620   |              . ****|     .              |
 13     0.095295       0.07071   |              .     |*    .              |
 14     0.194110       0.14403   |              .     |***  .              |
 15    -0.219688       -.16300   |              .  ***|     .              |

                         "." marks two standard errors
```

```
                            The ARIMA Procedure

                          Inverse Autocorrelations

     Lag    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

       1        0.72538    |                .   |***************     |
       2        0.48987    |                .   |**********          |
       3        0.35415    |                .   |*******             |
       4        0.34169    |                .   |*******             |
       5        0.33466    |                .   |*******             |
       6        0.34003    |                .   |*******             |
       7        0.24192    |                .   |*****               |
       8        0.12899    |                .   |***.                |
       9        0.06597    |                .   |*  .                |
      10        0.01654    |                .   |   .                |
      11        0.06434    |                .   |*  .                |
      12        0.08659    |                .   |** .                |
      13        0.02485    |                .   |   .                |
      14       -0.03545    |                .  *|   .                |
      15       -0.00113    |                .   |   .                |


                          Partial Autocorrelations

     Lag    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

       1       -0.51895    |          **********|   .                |
       2       -0.40526    |            ********|   .                |
       3       -0.07862    |                . **|   .                |
       4       -0.14588    |                .***|   .                |
       5        0.02735    |                .   |*  .                |
       6       -0.13782    |                .***|   .                |
       7       -0.16741    |                .***|   .                |
       8       -0.06041    |                .  *|   .                |
       9       -0.18372    |                ****|   .                |
      10       -0.01478    |                .   |   .                |
      11        0.14277    |                .   |***.                |
      12       -0.04345    |                .  *|   .                |
      13       -0.19959    |                ****|   .                |
      14        0.08302    |                .   |** .                |
      15        0.00278    |                .   |   .                |
```

```
                           The ARIMA Procedure

                   Autocorrelation Check for White Noise

 To      Chi-         Pr >
Lag    Square   DF   ChiSq  ---------------Autocorrelations---------------

  6     38.13    6  <.0001  -0.519  -0.027   0.182  -0.174   0.135  -0.137
 12     50.62   12  <.0001   0.007   0.109  -0.179   0.178   0.023  -0.186
```

The ESTIMATE statement fits an ARIMA(0,1,1) model to the simulated data. Note that in this case the parameter estimates are reasonably close to the values used to generate the simulated data. ($\mu = 0,\ \hat{\mu} = .02.\ \ \theta_1 = .8,\ \hat{\theta}_1 = .79.\ \ \sigma^2 = 1,\ \hat{\sigma}^2 = .82.$)

The ESTIMATE statement results are shown in Output 7.1.3.

Output 7.1.3. Output from Fitting ARIMA(0,1,1) Model

```
                              The ARIMA Procedure

                     Conditional Least Squares Estimation

                               Approx Std
    Parameter      Estimate         Error       t Value    Pr > |t|     Lag

    MU              0.02056       0.01972          1.04      0.2997       0
    MA1,1           0.79142       0.06474         12.22      <.0001       1


                        Constant Estimate      0.020558
                        Variance Estimate      0.819807
                        Std Error Estimate     0.905432
                        AIC                    263.2594
                        SBC                    268.4497
                        Number of Residuals          99
                 * AIC and SBC do not include log determinant.


                           Correlations of Parameter
                                   Estimates

                         Parameter        MU     MA1,1

                         MU            1.000    -0.124
                         MA1,1        -0.124     1.000


                       Autocorrelation Check of Residuals

    To      Chi-         Pr >
   Lag    Square   DF   ChiSq  ---------------Autocorrelations---------------

     6      6.48    5  0.2623  -0.033   0.030   0.153  -0.096   0.013  -0.163
    12     13.11   11  0.2862  -0.048   0.046  -0.086   0.159   0.027  -0.145
    18     20.12   17  0.2680   0.069   0.130  -0.099   0.006   0.164  -0.013
    24     24.73   23  0.3645   0.064   0.032   0.076  -0.077  -0.075   0.114


                             Model for variable u

                     Estimated Mean               0.020558
                     Period(s) of Differencing           1


                             Moving Average Factors

                          Factor 1:  1 - 0.79142 B**(1)
```

Example 7.2. Seasonal Model for the Airline Series

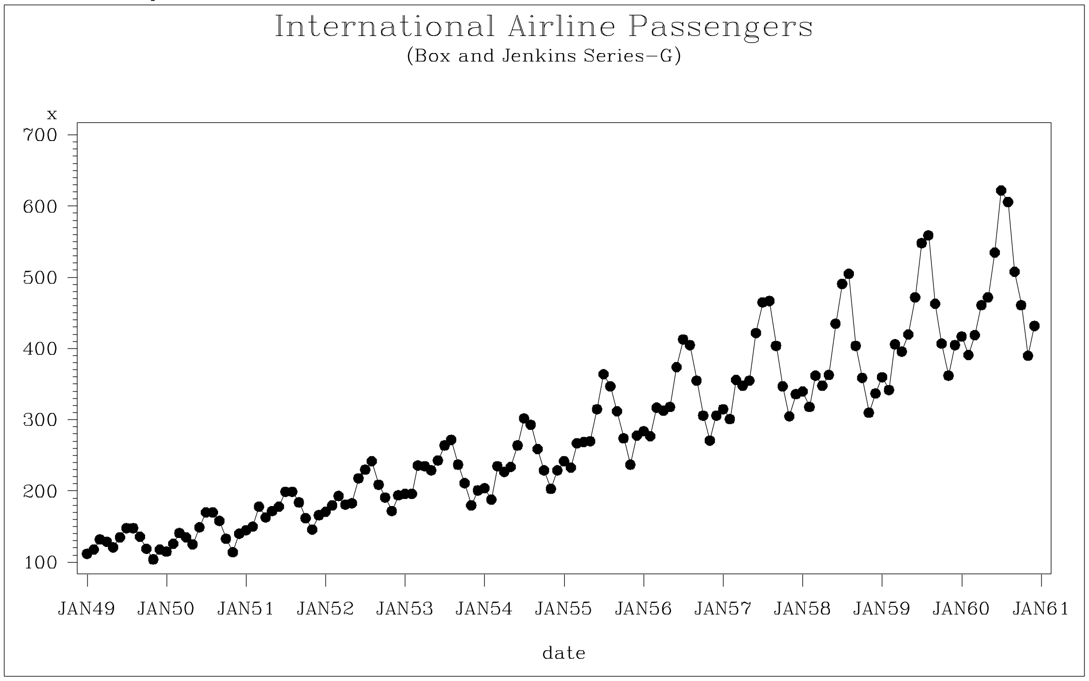
The airline passenger data, given as Series G in Box and Jenkins (1976), has been used in time series analysis literature as an example of a nonstationary seasonal time series. This example uses PROC ARIMA to fit the "airline model," ARIMA(0,1,1)$\times$(0,1,1)$_{12}$, to Box and Jenkins' Series G.

The following statements read the data and log transform the series. The PROC GPLOT step plots the series, as shown in Output 7.2.1.

```
title1 'International Airline Passengers';
title2 '(Box and Jenkins Series-G)';
data seriesg;
   input x @@;
   xlog = log( x );
   date = intnx( 'month', '31dec1948'd, _n_ );
   format date monyy.;
   datalines;
112 118 132 129 121 135 148 148 136 119 104 118
115 126 141 135 125 149 170 170 158 133 114 140
145 150 178 163 172 178 199 199 184 162 146 166
171 180 193 181 183 218 230 242 209 191 172 194
196 196 236 235 229 243 264 272 237 211 180 201
204 188 235 227 234 264 302 293 259 229 203 229
242 233 267 269 270 315 364 347 312 274 237 278
284 277 317 313 318 374 413 405 355 306 271 306
315 301 356 348 355 422 465 467 404 347 305 336
340 318 362 348 363 435 491 505 404 359 310 337
360 342 406 396 420 472 548 559 463 407 362 405
417 391 419 461 472 535 622 606 508 461 390 432
;

symbol1 i=join  v=dot;
proc gplot data=seriesg;
   plot x * date = 1 / haxis= '1jan49'd to '1jan61'd by year;
run;
```

Output 7.2.1. Plot of Data



The following PROC ARIMA step fits an ARIMA(0,1,1)$\times$(0,1,1)$_{12}$ model without a mean term to the logarithms of the airline passengers series. The model is forecast, and the results stored in the data set B.

```
proc arima data=seriesg;
   identify var=xlog(1,12) nlag=15;
   run;
   estimate q=(1)(12) noconstant method=uls;
   run;
   forecast out=b lead=24 id=date interval=month noprint;
quit;
```

The printed output from the IDENTIFY statement is shown in Output 7.2.2. The autocorrelation plots shown are for the twice differenced series $(1-B)(1-B^{12})X$. Note that the autocorrelation functions have the pattern characteristic of a first-order moving average process combined with a seasonal moving average process with lag 12.

Output 7.2.2. IDENTIFY Statement Output

```
                              The ARIMA Procedure

                            Name of Variable = xlog

             Period(s) of Differencing                        1,12
             Mean of Working Series                       0.000291
             Standard Deviation                           0.045673
             Number of Observations                            131
             Observation(s) eliminated by differencing          13


                                Autocorrelations

Lag   Covariance   Correlation   -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

  0    0.0020860       1.00000   |                    |********************|
  1   -0.0007116       -.34112   |             *******|  .                 |
  2   0.00021913       0.10505   |                .   |** .                |
  3   -0.0004217       -.20214   |                ****|   .                |
  4   0.00004456       0.02136   |                .   |   .                |
  5   0.00011610       0.05565   |                .   |*  .                |
  6   0.00006426       0.03080   |                .   |*  .                |
  7   -0.0001159       -.05558   |                .  *|   .                |
  8   -1.5867E-6       -.00076   |                .   |   .                |
  9   0.00036791       0.17637   |                .   |****                |
 10   -0.0001593       -.07636   |                . **|   .                |
 11   0.00013431       0.06438   |                .   |*  .                |
 12   -0.0008065       -.38661   |            ********|   .                |
 13   0.00031624       0.15160   |               .    |*** .               |
 14   -0.0001202       -.05761   |               .   *|    .               |
 15   0.00031200       0.14957   |               .    |*** .               |

                         "." marks two standard errors
```

```
                             The ARIMA Procedure

                           Inverse Autocorrelations

      Lag    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

        1        0.41027    |                 .  |********            |
        2        0.12711    |                 .  |***                 |
        3        0.10189    |                 .  |**.                 |
        4        0.01978    |                 .  |  .                 |
        5       -0.10310    |                 .**|  .                 |
        6       -0.11886    |                 .**|  .                 |
        7       -0.04088    |                 . *|  .                 |
        8       -0.05086    |                 . *|  .                 |
        9       -0.06022    |                 . *|  .                 |
       10        0.06460    |                 .  |* .                 |
       11        0.19907    |                 .  |****                |
       12        0.31709    |                 .  |******              |
       13        0.12434    |                 .  |**.                 |
       14        0.06583    |                 .  |* .                 |
       15        0.01515    |                 .  |  .                 |


                           Partial Autocorrelations

      Lag    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

        1       -0.34112    |             *******|  .                 |
        2       -0.01281    |                 .  |  .                 |
        3       -0.19266    |                ****|  .                 |
        4       -0.12503    |                 ***|  .                 |
        5        0.03309    |                 .  |* .                 |
        6        0.03468    |                 .  |* .                 |
        7       -0.06019    |                 . *|  .                 |
        8       -0.02022    |                 .  |  .                 |
        9        0.22558    |                 .  |*****               |
       10        0.04307    |                 .  |* .                 |
       11        0.04659    |                 .  |* .                 |
       12       -0.33869    |             *******|  .                 |
       13       -0.10918    |                 .**|  .                 |
       14       -0.07684    |                 .**|  .                 |
       15       -0.02175    |                 .  |  .                 |
```

```
                             The ARIMA Procedure

                     Autocorrelation Check for White Noise

   To      Chi-         Pr >
  Lag    Square   DF   ChiSq   ---------------Autocorrelations---------------

    6     23.27    6  0.0007  -0.341   0.105  -0.202   0.021   0.056   0.031
   12     51.47   12  <.0001  -0.056  -0.001   0.176  -0.076   0.064  -0.387
```

The results of the ESTIMATE statement are shown in Output 7.2.3.

Output 7.2.3. ESTIMATE Statement Output

```
                              The ARIMA Procedure

                    Unconditional Least Squares Estimation

                                 Approx Std
    Parameter      Estimate          Error       t Value    Pr > |t|     Lag

    MA1,1           0.39594        0.08149          4.86      <.0001       1
    MA2,1           0.61331        0.07961          7.70      <.0001      12


                        Variance Estimate      0.001363
                        Std Error Estimate     0.036921
                        AIC                    -484.755
                        SBC                    -479.005
                        Number of Residuals         131


                           Correlations of Parameter
                                   Estimates

                         Parameter     MA1,1     MA2,1

                         MA1,1         1.000    -0.055
                         MA2,1        -0.055     1.000


                       Autocorrelation Check of Residuals

    To      Chi-         Pr >
   Lag    Square   DF   ChiSq  ---------------Autocorrelations---------------

     6      5.56    4  0.2349   0.022   0.024  -0.125  -0.129   0.057   0.065
    12      8.49   10  0.5816  -0.065  -0.042   0.102  -0.060   0.023   0.007
    18     13.23   16  0.6560   0.022   0.039   0.045  -0.162   0.035   0.001
    24     24.99   22  0.2978  -0.106  -0.104  -0.037  -0.027   0.219   0.040


                          Model for variable xlog

                       Period(s) of Differencing    1,12


                           Moving Average Factors

                         Factor 1:  1 - 0.39594 B**(1)
                         Factor 2:  1 - 0.61331 B**(12)
```

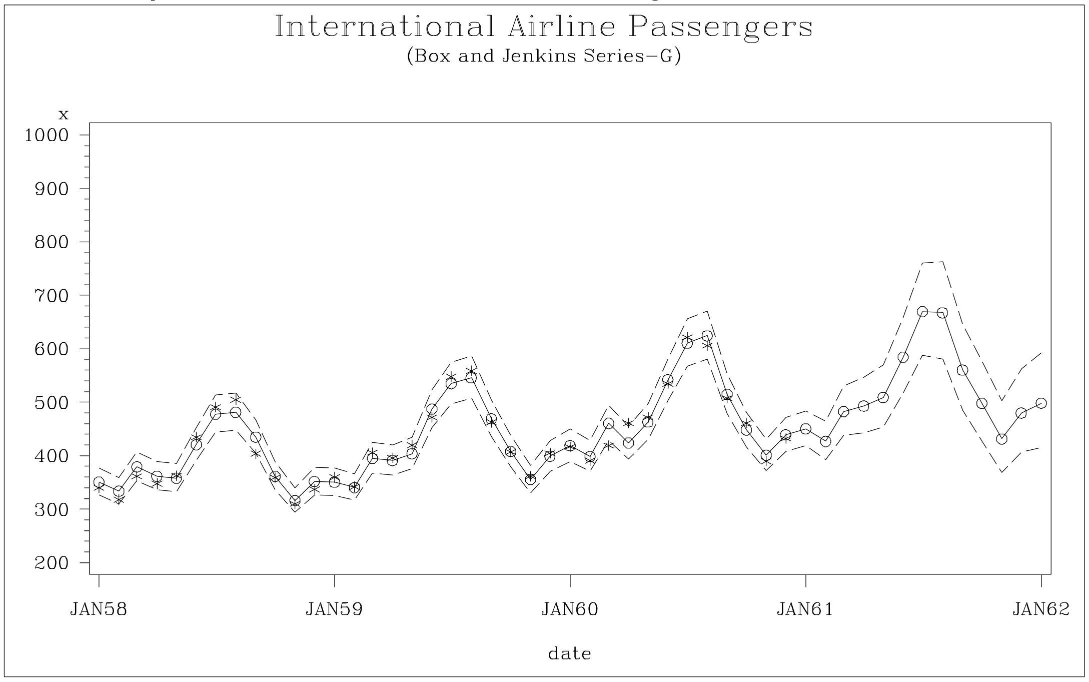
The following statements retransform the forecast values to get forecasts in the original scales. See the section "Forecasting Log Transformed Data" earlier in this chapter for more information.

```
data c;
   set b;
   x        = exp( xlog );
   forecast = exp( forecast + std*std/2 );
   l95      = exp( l95 );
   u95      = exp( u95 );
run;
```

The forecasts and their confidence limits are plotted using the following PROC GPLOT step. The plot is shown in Output 7.2.4.

```
   symbol1 i=none  v=star;
   symbol2 i=join  v=circle;
   symbol3 i=join  v=none l=3;
   proc gplot data=c;
      where date >= ’1jan58’d;
      plot x * date = 1 forecast * date = 2
           l95 * date = 3 u95 * date = 3 /
           overlay haxis= ’1jan58’d to ’1jan62’d by year;
   run;
```

Output 7.2.4. Plot of the Forecast for the Original Series



Example 7.3. Model for Series J Data from Box and Jenkins

This example uses the Series J data from Box and Jenkins (1976). First the input series, *X*, is modeled with a univariate ARMA model. Next, the dependent series, *Y*, is cross correlated with the input series. Since a model has been fit to *X*, both *Y* and *X* are prewhitened by this model before the sample cross correlations are computed. Next, a transfer function model is fit with no structure on the noise term. The residuals from this model are identified by means of the PLOT option; then, the full model, transfer function and noise is fit to the data.

The following statements read Input Gas Rate and Output CO_2 from a gas furnace. (Data values are not shown. See "Series J" in Box and Jenkins (1976) for the values.)

```
title1 'Gas Furnace Data';
title2 '(Box and Jenkins, Series J)';
data seriesj;
   input x y @@;
   label x = 'Input Gas Rate'
         y = 'Output CO2';
datalines;
;
```

The following statements produce Output 7.3.1 through Output 7.3.5.

```
proc arima data=seriesj;

   /*--- Look at the input process -------------------*/
   identify var=x nlag=10;
   run;

   /*--- Fit a model for the input -------------------*/
   estimate p=3;
   run;

   /*--- Crosscorrelation of prewhitened series ------*/
   identify var=y crosscorr=(x) nlag=10;
   run;

   /*--- Fit transfer function - look at residuals ---*/
   estimate input=( 3 $ (1,2)/(1,2) x ) plot;
   run;

   /*--- Estimate full model ------------------------*/
   estimate p=2 input=( 3 $ (1,2)/(1) x );
   run;

quit;
```

The results of the first IDENTIFY statement for the input series X are shown in Output 7.3.1.

Output 7.3.1. IDENTIFY Statement Results for X

```
                                Gas Furnace Data
                          (Box and Jenkins, Series J)

                              The ARIMA Procedure

                              Name of Variable = x

                       Mean of Working Series    -0.05683
                       Standard Deviation        1.070952
                       Number of Observations         296


                                Autocorrelations

Lag   Covariance   Correlation   -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

  0     1.146938       1.00000   |                    |********************|
  1     1.092430       0.95247   |                  . |******************* |
  2     0.956652       0.83409   |                .   |*****************   |
  3     0.782051       0.68186   |               .    |**************      |
  4     0.609291       0.53123   |               .    |***********         |
  5     0.467380       0.40750   |              .     |********            |
  6     0.364957       0.31820   |              .     |******              |
  7     0.298427       0.26019   |              .     |*****.              |
  8     0.260943       0.22751   |              .     |*****.              |
  9     0.244378       0.21307   |              .     |**** .              |
 10     0.238942       0.20833   |              .     |**** .              |

                         "." marks two standard errors


                            Inverse Autocorrelations

       Lag    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

         1       -0.71090    |      **************| .                  |
         2        0.26217    |                  . |*****               |
         3       -0.13005    |                 ***| .                  |
         4        0.14777    |                  . |***                 |
         5       -0.06803    |                  .*| .                  |
         6       -0.01147    |                  . | .                  |
         7       -0.01649    |                  . | .                  |
         8        0.06108    |                  . |*.                  |
         9       -0.04490    |                  .*| .                  |
        10        0.01100    |                  . | .                  |
                              The ARIMA Procedure

                            Partial Autocorrelations

      Lag    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

        1        0.95247    |                   . |******************* |
        2       -0.78796    |    ****************| .                   |
        3        0.33897    |                   . |*******             |
        4        0.12121    |                   . |**                  |
        5        0.05896    |                   . |*.                  |
        6       -0.11147    |                   **| .                  |
        7        0.04862    |                   . |*.                  |
        8        0.09945    |                   . |**                  |
        9        0.01587    |                   . | .                  |
       10       -0.06973    |                   .*| .                  |


                      Autocorrelation Check for White Noise

  To      Chi-          Pr >
 Lag    Square   DF   ChiSq  ---------------Autocorrelations---------------

   6    786.35    6  <.0001   0.952   0.834   0.682   0.531   0.408   0.318
```

The ESTIMATE statement results for the AR(3) model for the input series X are shown in Output 7.3.2.

Output 7.3.2. Estimates of the AR(3) Model for X

```
                              The ARIMA Procedure

                     Conditional Least Squares Estimation

                                Approx Std
 Parameter      Estimate         Error       t Value    Pr > |t|     Lag

 MU             -0.12280       0.10902         -1.13      0.2609       0
 AR1,1           1.97607       0.05499         35.94      <.0001       1
 AR1,2          -1.37499       0.09967        -13.80      <.0001       2
 AR1,3           0.34336       0.05502          6.24      <.0001       3


                    Constant Estimate      -0.00682
                    Variance Estimate      0.035797
                    Std Error Estimate       0.1892
                    AIC                    -141.667
                    SBC                    -126.906
                    Number of Residuals         296
                * AIC and SBC do not include log determinant.


                      Correlations of Parameter Estimates

            Parameter         MU     AR1,1     AR1,2     AR1,3

            MU             1.000    -0.017     0.014    -0.016
            AR1,1         -0.017     1.000    -0.941     0.790
            AR1,2          0.014    -0.941     1.000    -0.941
            AR1,3         -0.016     0.790    -0.941     1.000


                       Autocorrelation Check of Residuals

 To      Chi-         Pr >
Lag     Square   DF   ChiSq  ---------------Autocorrelations--------------

  6      10.30    3  0.0162  -0.042   0.068   0.056  -0.145  -0.009   0.059
 12      19.89    9  0.0186   0.014   0.002  -0.055   0.035   0.143  -0.079
 18      27.92   15  0.0221   0.099   0.043  -0.082   0.017   0.066  -0.052
 24      31.05   21  0.0729  -0.078   0.024   0.015   0.030   0.045   0.004
 30      34.58   27  0.1499  -0.007  -0.004   0.073  -0.038  -0.062   0.003
 36      38.84   33  0.2231   0.010   0.002   0.082   0.045   0.056  -0.023
 42      41.18   39  0.3753   0.002   0.033  -0.061  -0.003  -0.006  -0.043
 48      42.73   45  0.5687   0.018   0.051  -0.012   0.015  -0.027   0.020
```

```
                              The ARIMA Procedure

                              Model for variable x

                           Estimated Mean     -0.1228


                             Autoregressive Factors

         Factor 1:  1 - 1.97607 B**(1) + 1.37499 B**(2) - 0.34336 B**(3)
```

The IDENTIFY statement results for the dependent series Y cross correlated with the input series X is shown in Output 7.3.3. Since a model has been fit to X, both Y and X are prewhitened by this model before the sample cross correlations are computed.

Output 7.3.3. IDENTIFY Statement for Y Cross Correlated with X

```
                              The ARIMA Procedure

                            Partial Autocorrelations

      Lag    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

        1        0.97076    |                   . |******************* |
        2       -0.80388    |    ****************| .                   |
        3        0.18833    |                   . |****                |
        4        0.25999    |                   . |*****               |
        5        0.05949    |                   . |*.                  |
        6       -0.06258    |                   .*| .                  |
        7       -0.01435    |                   . | .                  |
        8        0.05490    |                   . |*.                  |
        9        0.00545    |                   . | .                  |
       10        0.03141    |                   . |*.                  |


                        Autocorrelation Check for White Noise

  To       Chi-           Pr >
 Lag     Square    DF    ChiSq    ---------------Autocorrelations---------------

   6    1023.15     6   <.0001    0.971    0.896    0.793    0.680    0.574    0.485
```

```
                              The ARIMA Procedure

                             Correlation of y and x

                  Number of Observations                 296
                  Variance of transformed series y    0.131438
                  Variance of transformed series x    0.035357

                       Both series have been prewhitened.


                               Crosscorrelations

Lag    Covariance    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

-10     0.0015683        0.02301     |                 . | .                 |
 -9    0.00013502        0.00198     |                 . | .                 |
 -8    -0.0060480        -.08872     |                 **| .                 |
 -7    -0.0017624        -.02585     |                 .*| .                 |
 -6    -0.0080539        -.11814     |                 **| .                 |
 -5    -0.0000944        -.00138     |                 . | .                 |
 -4    -0.0012802        -.01878     |                 . | .                 |
 -3    -0.0031078        -.04559     |                 .*| .                 |
 -2    0.00065212        0.00957     |                 . | .                 |
 -1    -0.0019166        -.02811     |                 .*| .                 |
  0    -0.0003673        -.00539     |                 . | .                 |
  1     0.0038939        0.05712     |                 . |*.                 |
  2    -0.0016971        -.02489     |                 . | .                 |
  3     -0.019231        -.28210     |             ******| .                 |
  4     -0.022479        -.32974     |            *******| .                 |
  5     -0.030909        -.45341     |          *********| .                 |
  6     -0.018122        -.26583     |              *****| .                 |
  7     -0.011426        -.16761     |                ***| .                 |
  8    -0.0017355        -.02546     |                 .*| .                 |
  9     0.0022590        0.03314     |                 . |*.                 |
 10    -0.0035152        -.05156     |                 .*| .                 |

                         "." marks two standard errors


                      Crosscorrelation Check Between Series

    To      Chi-         Pr >
   Lag    Square    DF   ChiSq  ---------------Crosscorrelations--------------

     5    117.75     6  <.0001  -0.005   0.057  -0.025  -0.282  -0.330  -0.453
```

```
                              The ARIMA Procedure

         Both variables have been prewhitened by the following filter:

                              Prewhitening Filter


                             Autoregressive Factors

         Factor 1:  1 - 1.97607 B**(1) + 1.37499 B**(2) - 0.34336 B**(3)
```

The ESTIMATE statement results for the transfer function model with no structure on the noise term is shown in Output 7.3.4. The PLOT option prints the residual autocorrelation functions from this model.

Output 7.3.4. Estimates of the Transfer Function Model

```
                              The ARIMA Procedure

                      Conditional Least Squares Estimation

                        Approx Std
Parameter    Estimate        Error     t Value  Pr > |t|   Lag  Variable  Shift

MU           53.32237      0.04932     1081.24    <.0001     0  y             0
NUM1         -0.62868      0.25385       -2.48    0.0138     0  x             3
NUM1,1        0.47258      0.62253        0.76    0.4484     1  x             3
NUM1,2        0.73660      0.81006        0.91    0.3640     2  x             3
DEN1,1        0.15411      0.90483        0.17    0.8649     1  x             3
DEN1,2        0.27774      0.57345        0.48    0.6285     2  x             3


                        Constant Estimate      53.32237
                        Variance Estimate      0.704241
                        Std Error Estimate     0.839191
                        AIC                    729.7249
                        SBC                    751.7648
                        Number of Residuals         291
                 * AIC and SBC do not include log determinant.


                      Correlations of Parameter Estimates

Variable                  y         x         x         x         x         x
Parameter                MU      NUM1    NUM1,1    NUM1,2    DEN1,1    DEN1,2

y              MU     1.000     0.013     0.002    -0.002     0.004    -0.006
x            NUM1     0.013     1.000     0.755    -0.447     0.089    -0.065
x          NUM1,1     0.002     0.755     1.000     0.121    -0.538     0.565
x          NUM1,2    -0.002    -0.447     0.121     1.000    -0.892     0.870
x          DEN1,1     0.004     0.089    -0.538    -0.892     1.000    -0.998
x          DEN1,2    -0.006    -0.065     0.565     0.870    -0.998     1.000
```

```
                              The ARIMA Procedure

                       Autocorrelation Check of Residuals

   To      Chi-         Pr >
  Lag    Square   DF   ChiSq   ---------------Autocorrelations---------------

    6    496.45    6  <.0001    0.893   0.711   0.502   0.312   0.167   0.064
   12    498.58   12  <.0001   -0.003  -0.040  -0.054  -0.040  -0.022  -0.021
   18    539.38   18  <.0001   -0.045  -0.083  -0.131  -0.170  -0.196  -0.195
   24    561.87   24  <.0001   -0.163  -0.102  -0.026   0.047   0.106   0.142
   30    585.90   30  <.0001    0.158   0.156   0.131   0.081   0.013  -0.037
   36    592.42   36  <.0001   -0.048  -0.018   0.038   0.070   0.079   0.067
   42    593.44   42  <.0001    0.042   0.025   0.013   0.004   0.006   0.019
   48    601.94   48  <.0001    0.043   0.068   0.084   0.082   0.061   0.023
```

```
                              The ARIMA Procedure

                       Autocorrelation Plot of Residuals

Lag   Covariance   Correlation   -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

  0     0.704241       1.00000   |                    |********************|
  1     0.628846       0.89294   |                  . |******************  |
  2     0.500490       0.71068   |                .   |**************      |
  3     0.353404       0.50182   |                .   |**********          |
  4     0.219895       0.31224   |               .    |******              |
  5     0.117330       0.16660   |               .    |*** .               |
  6     0.044967       0.06385   |               .    |*   .               |
  7   -0.0023551       -.00334   |               .    |    .               |
  8    -0.028030       -.03980   |               .   *|    .               |
  9    -0.037891       -.05380   |               .   *|    .               |
 10    -0.028378       -.04030   |               .   *|    .               |

                         "." marks two standard errors


                            Inverse Autocorrelations

       Lag    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

         1       -0.57346    |         ***********| .                  |
         2        0.02264    |                  . | .                  |
         3        0.03631    |                  . |*.                  |
         4        0.03941    |                  . |*.                  |
         5       -0.01256    |                  . | .                  |
         6       -0.01618    |                  . | .                  |
         7        0.02680    |                  . |*.                  |
         8       -0.05895    |                  .*| .                  |
         9        0.07043    |                  . |*.                  |
        10       -0.02987    |                  .*| .                  |
                              The ARIMA Procedure

                            Partial Autocorrelations

       Lag    Correlation    -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1

         1        0.89294    |                   . |******************  |
         2       -0.42765    |           *********| .                 |
         3       -0.13463    |                 ***| .                 |
         4        0.02199    |                   . | .                 |
         5        0.03891    |                   . |*.                 |
         6       -0.02219    |                   . | .                 |
         7       -0.02249    |                   . | .                 |
         8        0.01538    |                   . | .                 |
         9        0.00634    |                   . | .                 |
        10        0.07737    |                   . |**                 |


                 Crosscorrelation Check of Residuals with Input x

   To      Chi-         Pr >
  Lag    Square   DF   ChiSq  ---------------Crosscorrelations--------------

    5      0.48    2  0.7855  -0.009  -0.005   0.026   0.013  -0.017  -0.022
   11      0.93    8  0.9986  -0.006   0.008   0.022   0.023  -0.017  -0.013
   17      2.63   14  0.9996   0.012   0.035   0.037   0.039  -0.005  -0.040
   23     19.19   20  0.5092  -0.076  -0.108  -0.122  -0.122  -0.094  -0.041
   29     20.12   26  0.7857  -0.039  -0.013   0.010  -0.020  -0.031  -0.005
   35     24.22   32  0.8363  -0.022  -0.031  -0.074  -0.036   0.014   0.076
   41     30.66   38  0.7953   0.108   0.091   0.046   0.018   0.003   0.009
   47     31.65   44  0.9180   0.008  -0.011  -0.040  -0.030  -0.002   0.028


                             Model for variable y

                       Estimated Intercept    53.32237


                               Input Number 1

                              Input Variable    x
                              Shift             3


                               Numerator Factors

              Factor 1:  -0.6287 - 0.47258 B**(1) - 0.7366 B**(2)


                              Denominator Factors

                Factor 1:  1 - 0.15411 B**(1) - 0.27774 B**(2)
```

The ESTIMATE statement results for the final transfer function model with AR(2) noise are shown in Output 7.3.5.

Output 7.3.5. Estimates of the Final Model

```
                              The ARIMA Procedure

                      Conditional Least Squares Estimation

                        Approx Std
Parameter    Estimate       Error     t Value   Pr > |t|   Lag  Variable  Shift

MU           53.26307     0.11926      446.63    <.0001     0  y             0
AR1,1         1.53292     0.04754       32.25    <.0001     1  y             0
AR1,2        -0.63297     0.05006      -12.64    <.0001     2  y             0
NUM1         -0.53522     0.07482       -7.15    <.0001     0  x             3
NUM1,1        0.37602     0.10287        3.66    0.0003     1  x             3
NUM1,2        0.51894     0.10783        4.81    <.0001     2  x             3
DEN1,1        0.54842     0.03822       14.35    <.0001     1  x             3


                        Constant Estimate      5.329371
                        Variance Estimate      0.058828
                        Std Error Estimate     0.242544
                        AIC                    8.292811
                        SBC                    34.00607
                        Number of Residuals         291
                 * AIC and SBC do not include log determinant.


                      Correlations of Parameter Estimates

          Variable                  y          y          y          x
          Parameter                MU      AR1,1      AR1,2       NUM1

          y             MU      1.000     -0.063      0.047     -0.008
          y          AR1,1     -0.063      1.000     -0.927     -0.003
          y          AR1,2      0.047     -0.927      1.000      0.023
          x           NUM1     -0.008     -0.003      0.023      1.000
          x         NUM1,1     -0.016      0.007     -0.005      0.713
          x         NUM1,2      0.017     -0.002      0.005     -0.178
          x         DEN1,1     -0.049      0.015     -0.022     -0.013

                      Correlations of Parameter Estimates

               Variable                  x          x          x
               Parameter            NUM1,1     NUM1,2     DEN1,1

               y             MU     -0.016      0.017     -0.049
               y          AR1,1      0.007     -0.002      0.015
               y          AR1,2     -0.005      0.005     -0.022
               x           NUM1      0.713     -0.178     -0.013
               x         NUM1,1      1.000     -0.467     -0.039
               x         NUM1,2     -0.467      1.000     -0.720
               x         DEN1,1     -0.039     -0.720      1.000
```

```
                              The ARIMA Procedure

                       Autocorrelation Check of Residuals

 To        Chi-          Pr >
Lag      Square     DF   ChiSq   ---------------Autocorrelations---------------

  6        8.61      4  0.0717    0.024    0.055   -0.073   -0.054   -0.054    0.119
 12       15.43     10  0.1172    0.032    0.028   -0.081    0.047    0.022    0.107
 18       21.13     16  0.1734   -0.038    0.052   -0.093   -0.013   -0.073   -0.005
 24       27.52     22  0.1922   -0.118   -0.002   -0.007    0.076    0.024   -0.004
 30       36.94     28  0.1202    0.034   -0.021    0.020    0.094   -0.118    0.065
 36       44.26     34  0.1119   -0.025   -0.057    0.113    0.022    0.030    0.065
 42       45.62     40  0.2500   -0.017   -0.036   -0.029   -0.013   -0.033    0.017
 48       48.60     46  0.3689    0.024    0.069    0.024    0.017    0.022   -0.044


                Crosscorrelation Check of Residuals with Input x

 To        Chi-          Pr >
Lag      Square     DF   ChiSq   ---------------Crosscorrelations--------------

  5        0.93      3  0.8191    0.008    0.004    0.010    0.008   -0.045    0.030
 11        6.60      9  0.6784    0.075   -0.024   -0.019   -0.026   -0.111    0.013
 17       13.86     15  0.5365    0.050    0.043    0.014    0.014   -0.141   -0.028
 23       18.55     21  0.6142   -0.074   -0.078    0.023   -0.016    0.021    0.060
 29       27.99     27  0.4113   -0.071   -0.001    0.038   -0.156    0.031    0.035
 35       35.18     33  0.3654   -0.014    0.015   -0.039    0.028    0.046    0.142
 41       37.15     39  0.5544    0.031   -0.029   -0.070   -0.006    0.012   -0.004
 47       42.42     45  0.5818    0.036   -0.038   -0.053    0.107    0.029    0.021
```

```
                              The ARIMA Procedure

                             Model for variable y

                       Estimated Intercept    53.26307


                            Autoregressive Factors

                 Factor 1:  1 - 1.53292 B**(1) + 0.63297 B**(2)


                               Input Number 1

                              Input Variable    x
                              Shift             3


                              Numerator Factors

              Factor 1:  -0.5352 - 0.37602 B**(1) - 0.51894 B**(2)


                             Denominator Factors

                         Factor 1:  1 - 0.54842 B**(1)
```

Example 7.4. An Intervention Model for Ozone Data

This example fits an intervention model to ozone data as suggested by Box and Tiao (1975). Notice that since the response variable, OZONE, is differenced, the innovation, X1, must also be differenced to generate a step function change in the response. If X1 had not been differenced, the change in the response caused by X1 would be a (seasonal) ramp and not a step function. Notice that the final model for the differenced data is a multiple regression model with a moving-average structure assumed for the residuals.

The model is fit by maximum likelihood. The seasonal moving-average parameter and its standard error are fairly sensitive to which method is chosen to fit the model, in agreement with the observations of Davidson (1981) and Ansley and Newbold (1980); thus, fitting the model by the unconditional or conditional least squares methods produce somewhat different estimates for these parameters.

Some missing values are appended to the end of the input data to generate additional values for the independent variables. Since the independent variables are not modeled, values for them must be available for any times at which predicted values are desired. In this case, predicted values are requested for 12 periods beyond the end of the data. Thus, values for X1, WINTER, and SUMMER must be given for 12 periods ahead.

The following statements read in the data and compute dummy variables for use as intervention inputs:

```
title1 'Intervention Data for Ozone Concentration';
title2 '(Box and Tiao, JASA 1975 P.70)';

data air;
   input ozone @@;
   label ozone  = 'Ozone Concentration'
         x1     = 'Intervention for post 1960 period'
         summer = 'Summer Months Intervention'
         winter = 'Winter Months Intervention';
   date = intnx( 'month', '31dec1954'd, _n_ );
   format date monyy.;
   month = month( date );
   year = year( date );
   x1 = year >= 1960;
   summer = ( 5 < month < 11 ) * ( year > 1965 );
   winter = ( year > 1965 ) - summer;
datalines;
2.7  2.0  3.6  5.0  6.5  6.1  5.9  5.0  6.4  7.4  8.2  3.9
4.1  4.5  5.5  3.8  4.8  5.6  6.3  5.9  8.7  5.3  5.7  5.7
3.0  3.4  4.9  4.5  4.0  5.7  6.3  7.1  8.0  5.2  5.0  4.7
3.7  3.1  2.5  4.0  4.1  4.6  4.4  4.2  5.1  4.6  4.4  4.0
2.9  2.4  4.7  5.1  4.0  7.5  7.7  6.3  5.3  5.7  4.8  2.7
1.7  2.0  3.4  4.0  4.3  5.0  5.5  5.0  5.4  3.8  2.4  2.0
2.2  2.5  2.6  3.3  2.9  4.3  4.2  4.2  3.9  3.9  2.5  2.2
2.4  1.9  2.1  4.5  3.3  3.4  4.1  5.7  4.8  5.0  2.8  2.9
1.7  3.2  2.7  3.0  3.4  3.8  5.0  4.8  4.9  3.5  2.5  2.4
```

```
1.6  2.3  2.5  3.1  3.5  4.5  5.7  5.0  4.6  4.8  2.1  1.4
2.1  2.9  2.7  4.2  3.9  4.1  4.6  5.8  4.4  6.1  3.5  1.9
1.8  1.9  3.7  4.4  3.8  5.6  5.7  5.1  5.6  4.8  2.5  1.5
1.8  2.5  2.6  1.8  3.7  3.7  4.9  5.1  3.7  5.4  3.0  1.8
2.1  2.6  2.8  3.2  3.5  3.5  4.9  4.2  4.7  3.7  3.2  1.8
2.0  1.7  2.8  3.2  4.4  3.4  3.9  5.5  3.8  3.2  2.3  2.2
1.3  2.3  2.7  3.3  3.7  3.0  3.8  4.7  4.6  2.9  1.7  1.3
1.8  2.0  2.2  3.0  2.4  3.5  3.5  3.3  2.7  2.5  1.6  1.2
1.5  2.0  3.1  3.0  3.5  3.4  4.0  3.8  3.1  2.1  1.6  1.3
;
```

The following statements produce Output 7.4.1 and Output 7.4.2:

```
proc arima data=air;

   /*--- Identify and seasonally difference ozone series ---*/
   identify var=ozone(12) crosscorr=( x1(12) summer winter ) noprint;

   /*--- Fit a multiple regression with a seasonal MA model ---*/
   /*---     by the maximum likelihood method ---*/
   estimate q=(1)(12) input=( x1 summer winter )
            noconstant method=ml itprint;

   /*--- Forecast ---*/
   forecast  lead=12 id=date interval=month;

run;
```

The ESTIMATE statement results are shown in Output 7.4.1.

Output 7.4.1. Parameter Estimates

```
                   Intervention Data for Ozone Concentration
                         (Box and Tiao, JASA 1975 P.70)

                              The ARIMA Procedure

                            Initial Moving Average
                                   Estimates

                                          Estimate

                               1          -0.29241


                            Initial Moving Average
                                   Estimates

                                          Estimate

                              12           0.40740


                      White Noise Variance Est    0.944969
```

```
                              The ARIMA Procedure

                      Conditional Least Squares Estimation

Iteration       SSE     MA1,1     MA2,1      NUM1      NUM2      NUM3    Lambda

        0    154.53  -0.29241   0.40740  -1.13490  -0.11731   0.05581   0.00001
        1    146.20  -0.29256   0.59844  -1.20292  -0.29784  -0.11572      1E-6
        2    145.88  -0.30071   0.59239  -1.26173  -0.26252  -0.08247      1E-7
        3    145.88  -0.29976   0.59242  -1.26246  -0.26150  -0.08197      1E-8
        4    145.88  -0.29983   0.59234  -1.26243  -0.26154  -0.08196      1E-9

                               Conditional Least
                               Squares Estimation

                              Iteration    R Crit

                                      0         1
                                      1  0.230552
                                      2  0.046601
                                      3  0.001345
                                      4  0.000125


                          Maximum Likelihood Estimation

 Iter    Loglike    MA1,1    MA2,1     NUM1     NUM2     NUM3   Lambda   R Crit

    0 -249.07778 -0.29983  0.59234 -1.26243 -0.26154 -0.08196  0.00001        1
    1 -245.89135 -0.26830  0.76634 -1.34490 -0.23984 -0.07578     1E-6 0.169445
    2 -245.88484 -0.26653  0.76623 -1.33046 -0.23939 -0.08025     1E-7 0.008044
    3 -245.88482 -0.26689  0.76661 -1.33070 -0.23936 -0.08020     1E-8 0.000603
    4 -245.88481 -0.26684  0.76665 -1.33062 -0.23936 -0.08021     1E-9 0.000073


                     ARIMA Estimation Optimization Summary

Estimation Method                                             Maximum Likelihood
Parameters Estimated                                                           5
Termination Criteria                          Maximum Relative Change in Estimates
Iteration Stopping Value                                                   0.001
Criteria Value                                                          0.000195
Alternate Criteria                           Relative Change in Objective Function
Alternate Criteria Value                                                1.247E-8
Maximum Absolute Value of Gradient                                       0.00712
R-Square Change from Last Iteration                                     0.000073
Objective Function                                        Log Gaussian Likelihood
Objective Function Value                                                -245.885
Marquardt's Lambda Coefficient                                              1E-9
Numerical Derivative Perturbation Delta                                    0.001
Iterations                                                                     4
```

```
                                 The ARIMA Procedure

                            Maximum Likelihood Estimation

                        Approx Std
Parameter    Estimate        Error     t Value  Pr > |t|   Lag  Variable  Shift

MA1,1        -0.26684      0.06710       -3.98    <.0001     1  ozone         0
MA2,1         0.76665      0.05973       12.83    <.0001    12  ozone         0
NUM1         -1.33062      0.19236       -6.92    <.0001     0  x1            0
NUM2         -0.23936      0.05952       -4.02    <.0001     0  summer        0
NUM3         -0.08021      0.04978       -1.61    0.1071     0  winter        0


                        Variance Estimate      0.634506
                        Std Error Estimate     0.796559
                        AIC                    501.7696
                        SBC                    518.3602
                        Number of Residuals         204


                      Correlations of Parameter Estimates

     Variable              ozone      ozone        x1    summer    winter
     Parameter             MA1,1      MA2,1      NUM1      NUM2      NUM3

     ozone       MA1,1     1.000      0.090    -0.039     0.062    -0.034
     ozone       MA2,1     0.090      1.000    -0.169     0.211     0.022
     x1           NUM1    -0.039     -0.169     1.000    -0.124    -0.107
     summer       NUM2     0.062      0.211    -0.124     1.000     0.097
     winter       NUM3    -0.034      0.022    -0.107     0.097     1.000


                       Autocorrelation Check of Residuals

   To      Chi-         Pr >
  Lag    Square   DF   ChiSq  ---------------Autocorrelations--------------

    6      7.47    4  0.1132   0.017   0.054   0.043   0.101  -0.022   0.140
   12     10.21   10  0.4220  -0.024  -0.059  -0.047   0.014   0.032   0.072
   18     14.53   16  0.5593   0.054   0.006  -0.110   0.028  -0.042   0.043
   24     19.99   22  0.5834   0.003  -0.074  -0.074   0.098  -0.038   0.043
   30     27.00   28  0.5180  -0.072  -0.035   0.023  -0.028  -0.107   0.100
   36     32.65   34  0.5336   0.022  -0.099  -0.006   0.087  -0.046   0.053
```

```
                           The ARIMA Procedure

                        Model for variable ozone

                     Period(s) of Differencing    12


                          Moving Average Factors

                      Factor 1:  1 + 0.26684 B**(1)
                      Factor 2:  1 - 0.76665 B**(12)


                             Input Number 1

                   Input Variable                    x1
                   Period(s) of Differencing         12
                   Overall Regression Factor   -1.33062


                             Input Number 2

                   Input Variable                summer
                   Overall Regression Factor   -0.23936


                             Input Number 3

                   Input Variable                winter
                   Overall Regression Factor   -0.08021
```

The FORECAST statement results are shown in Output 7.4.2.

Output 7.4.2. Forecasts

```
                           The ARIMA Procedure

                        Forecasts for variable ozone

          Obs       Forecast    Std Error       95% Confidence Limits

          217         1.4205       0.7966        -0.1407         2.9817
          218         1.8446       0.8244         0.2287         3.4604
          219         2.4567       0.8244         0.8408         4.0725
          220         2.8590       0.8244         1.2431         4.4748
          221         3.1501       0.8244         1.5342         4.7659
          222         2.7211       0.8244         1.1053         4.3370
          223         3.3147       0.8244         1.6989         4.9306
          224         3.4787       0.8244         1.8629         5.0946
          225         2.9405       0.8244         1.3247         4.5564
          226         2.3587       0.8244         0.7429         3.9746
          227         1.8588       0.8244         0.2429         3.4746
          228         1.2898       0.8244        -0.3260         2.9057
```

Example 7.5. Using Diagnostics to Identify ARIMA models

Fitting ARIMA models is as much an art as it is a science. The ARIMA procedure has diagnostic options to help tentatively identify the orders of both stationary and nonstationary ARIMA processes.

Consider the Series A in Box *et al* (1994), which consists of 197 concentration readings taken every two hours from a chemical process. Let SeriesA be a data set containing these readings in a variable named X. The following SAS statements use the SCAN option of the IDENTIFY statement to generate Output 7.5.1 and Output 7.5.2. See “The SCAN Method” for details of the SCAN method.

```
proc arima data=SeriesA;
   identify var=x scan;
run;
```

Output 7.5.1. Example of SCAN Tables

```
               SERIES A: Chemical Process Concentration Readings

                              The ARIMA Procedure

                    Squared Canonical Correlation Estimates

        Lags      MA 0      MA 1      MA 2      MA 3      MA 4      MA 5

        AR 0    0.3263    0.2479    0.1654    0.1387    0.1183    0.1417
        AR 1    0.0643    0.0012    0.0028    <.0001    0.0051    0.0002
        AR 2    0.0061    0.0027    0.0021    0.0011    0.0017    0.0079
        AR 3    0.0072    <.0001    0.0007    0.0005    0.0019    0.0021
        AR 4    0.0049    0.0010    0.0014    0.0014    0.0039    0.0145
        AR 5    0.0202    0.0009    0.0016    <.0001    0.0126    0.0001


                     SCAN Chi-Square[1] Probability Values

        Lags      MA 0      MA 1      MA 2      MA 3      MA 4      MA 5

        AR 0    <.0001    <.0001    <.0001    0.0007    0.0037    0.0024
        AR 1    0.0003    0.6649    0.5194    0.9235    0.3993    0.8528
        AR 2    0.2754    0.5106    0.5860    0.7346    0.6782    0.2766
        AR 3    0.2349    0.9812    0.7667    0.7861    0.6810    0.6546
        AR 4    0.3297    0.7154    0.7113    0.6995    0.5807    0.2205
        AR 5    0.0477    0.7254    0.6652    0.9576    0.2660    0.9168
```

In Output 7.5.1, there is one (maximal) rectangular region in which all the elements are insignificant with 95% confidence. This region has a vertex at (1,1). Output 7.5.2 gives recommendations based on the significance level specified by the ALPHA=*siglevel* option.

Output 7.5.2. Example of SCAN Option Tentative Order Selection

```
                              The ARIMA Procedure

                                  ARMA(p+d,q)
                                   Tentative
                                     Order
                                   Selection
                                     Tests

                                  ----SCAN---
                                  p+d       q

                                    1       1

                            (5% Significance Level)
```

Another order identification diagnostic is the extended sample autocorrelation function or ESACF method. See "The ESACF Method" for details of the ESACF method.

The following statements generate Output 7.5.3 and Output 7.5.4.

```
proc arima data=SeriesA;
   identify var=x esacf;
run;
```

Output 7.5.3. Example of ESACF Tables

```
                              The ARIMA Procedure

                    Extended Sample Autocorrelation Function

        Lags      MA 0      MA 1      MA 2      MA 3      MA 4      MA 5

        AR 0    0.5702    0.4951    0.3980    0.3557    0.3269    0.3498
        AR 1   -0.3907    0.0425   -0.0605   -0.0083   -0.0651   -0.0127
        AR 2   -0.2859   -0.2699   -0.0449    0.0089   -0.0509   -0.0140
        AR 3   -0.5030   -0.0106    0.0946   -0.0137   -0.0148   -0.0302
        AR 4   -0.4785   -0.0176    0.0827   -0.0244   -0.0149   -0.0421
        AR 5   -0.3878   -0.4101   -0.1651    0.0103   -0.1741   -0.0231


                            ESACF Probability Values

        Lags      MA 0      MA 1      MA 2      MA 3      MA 4      MA 5

        AR 0    <.0001    <.0001    0.0001    0.0014    0.0053    0.0041
        AR 1    <.0001    0.5974    0.4622    0.9198    0.4292    0.8768
        AR 2    <.0001    0.0002    0.6106    0.9182    0.5683    0.8592
        AR 3    <.0001    0.9022    0.2400    0.8713    0.8930    0.7372
        AR 4    <.0001    0.8380    0.3180    0.7737    0.8913    0.6213
        AR 5    <.0001    <.0001    0.0765    0.9142    0.1038    0.8103
```

In Output 7.5.3, there are three right-triangular regions in which all elements are insignificant at the 5% level. The triangles have vertices (1,1), (3,1), and (4,1). Since the triangle at (1,1) covers more insignificant terms, it is recommended first. Similarly, the remaining recommendations are ordered by the number of insignificant terms contained in the triangle. Output 7.5.4 gives recommendations based on the significance level specified by the ALPHA=*siglevel* option.

Output 7.5.4. Example of ESACF Option Tentative Order Selection

```
                              The ARIMA Procedure

                                  ARMA(p+d,q)
                                   Tentative
                                     Order
                                   Selection
                                     Tests

                                  ---ESACF---
                                  p+d       q

                                    1       1
                                    3       1
                                    4       1

                            (5% Significance Level)
```

If you also specify the SCAN option in the same IDENTIFY statement, the two recommendations are printed side by side.

```
   proc arima data=SeriesA;
      identify var=x scan esacf;
   run;
```

Output 7.5.5. Example of SCAN and ESACF Option Combined

```
                              The ARIMA Procedure

                              ARMA(p+d,q) Tentative
                              Order Selection Tests

                             ---SCAN--    --ESACF--
                             p+d     q    p+d     q

                               1     1      1     1
                                            3     1
                                            4     1

                            (5% Significance Level)
```

From above, the autoregressive and moving average orders are tentatively identified by both SCAN and ESACF tables to be $(p+d,q)$=(1,1). Because both the SCAN and ESACF indicate a $p+d$ term of 1, a unit root test should be used to determine whether this autoregressive term is a unit root. Since a moving average term appears to be present, a large autoregressive term is appropriate for the Augmented Dickey-Fuller test for a unit root.

Submitting the following code generates Output 7.5.6.

```
   proc arima data=SeriesA;
      identify var=x stationarity=(adf=(5,6,7,8));
   run;
```

Output 7.5.6. Example of STATIONARITY Option Output

```
                              The ARIMA Procedure

                    Augmented Dickey-Fuller Unit Root Tests

Type          Lags         Rho    Pr < Rho       Tau    Pr < Tau          F    Pr > F

Zero Mean        5      0.0403      0.6913      0.42      0.8024
                 6      0.0479      0.6931      0.63      0.8508
                 7      0.0376      0.6907      0.49      0.8200
                 8      0.0354      0.6901      0.48      0.8175
Single Mean      5    -18.4550      0.0150     -2.67      0.0821       3.67    0.1367
                 6    -10.8939      0.1043     -2.02      0.2767       2.27    0.4931
                 7    -10.9224      0.1035     -1.93      0.3172       2.00    0.5605
                 8    -10.2992      0.1208     -1.83      0.3650       1.81    0.6108
Trend            5    -18.4360      0.0871     -2.66      0.2561       3.54    0.4703
                 6    -10.8436      0.3710     -2.01      0.5939       2.04    0.7694
                 7    -10.7427      0.3773     -1.90      0.6519       1.91    0.7956
                 8    -10.0370      0.4236     -1.79      0.7081       1.74    0.8293
```

The preceding test results show that a unit root is very likely and that the series should be differenced. Based on this test and the previous results, an ARIMA(0,1,1) would be a good choice for a tentative model for Series A.

Using the recommendation that the series be differenced, the following statements generate Output 7.5.7.

```
   proc arima data=SeriesA;
      identify var=x(1) minic;
   run;
```

Output 7.5.7. Example of MINIC Table

```
                              The ARIMA Procedure

                         Minimum Information Criterion

        Lags      MA 0      MA 1      MA 2      MA 3      MA 4      MA 5

        AR 0  -2.05761   -2.3497  -2.32358  -2.31298  -2.30967  -2.28528
        AR 1  -2.23291  -2.32345  -2.29665  -2.28644  -2.28356  -2.26011
        AR 2  -2.23947  -2.30313  -2.28084  -2.26065  -2.25685  -2.23458
        AR 3  -2.25092  -2.28088  -2.25567  -2.23455  -2.22997  -2.20769
        AR 4  -2.25934   -2.2778  -2.25363  -2.22983  -2.20312  -2.19531
        AR 5   -2.2751  -2.26805  -2.24249  -2.21789  -2.19667  -2.17426
```

The error series is estimated using an AR(7) model, and the minimum of this MINIC table is $BIC(0,1)$. This diagnostic confirms the previous result indicating that an ARIMA(0,1,1) is a tentative model for Series A.

If you also specify the SCAN or MINIC option in the same IDENTIFY statement, the BIC associated with the SCAN table and ESACF table recommendations are listed.

```
   proc arima data=SeriesA;
      identify var=x(1) minic scan esacf;
   run;
```

Output 7.5.8. Example of SCAN, ESACF, MINIC Options Combined

```
                              The ARIMA Procedure

                   ARMA(p+d,q) Tentative Order Selection Tests

                 ---------SCAN--------    --------ESACF--------
                 p+d     q         BIC    p+d     q         BIC

                   0     1     -2.3497      0     1     -2.3497
                                            1     1    -2.32345

                            (5% Significance Level)
```

References

Akaike, H. (1974), "A New Look at the Statistical Model Identification," *IEEE Transaction on Automatic Control*, AC-19, 716-723.

Anderson, T.W. (1971), *The Statistical Analysis of Time Series*, New York: John Wiley & Sons, Inc.

Andrews and Herzberg (1985), *A collection of problems from many fields for the student and research worker*, New York: Springer Verlag.

Ansley, C. (1979), "An Algorithm for the Exact Likelihood of a Mixed Autoregressive-Moving Average Process," *Biometrika*, 66, 59.

Ansley, C. and Newbold, P. (1980), "Finite Sample Properties of Estimators for Autoregressive Moving Average Models," *Journal of Econometrics*, 13, 159.

Bhansali, R.J. (1980), "Autoregressive and Window Estimates of the Inverse Correlation Function," *Biometrika*, 67, 551-566.

Box, G.E.P. and Jenkins, G.M. (1976), *Time Series Analysis: Forecasting and Control*, San Francisco: Holden-Day.

Box, G.E.P., Jenkins, G.M., and Reinsel, G.C. (1994), *Time Series Analysis: Forecasting and Control*, Third Edition, Englewood Cliffs, NJ: Prentice Hall, 197-199.

Box, G.E.P. and Tiao, G.C. (1975), "Intervention Analysis with Applications to Economic and Environmental Problems," *JASA*, 70, 70-79.

Brocklebank, J.C. and Dickey, D.A. (1986), *SAS System for Forecasting Time Series, 1986 Edition*, Cary, North Carolina: SAS Institute Inc.

Chatfield, C. (1980), "Inverse Autocorrelations," *Journal of the Royal Statistical Society*, A142, 363-377.

Choi, ByoungSeon (1992), *ARMA Model Identification*, New York: Springer-Verlag, 129-132.

Cleveland, W.S. (1972), "The Inverse Autocorrelations of a Time Series and Their Applications," *Technometrics*, 14, 277.

Davidson, J. (1981), "Problems with the Estimation of Moving Average Models," *Journal of Econometrics*, 16, 295.

Davies, N., Triggs, C.M., and Newbold, P. (1977), "Significance Levels of the Box-Pierce Portmanteau Statistic in Finite Samples," *Biometrika*, 64, 517-522.

Dickey, D. A., and Fuller, W.A. (1979), "Distribution of the Estimators for Autoregressive Time Series With a Unit Root," *Journal of the American Statistical Association*, 74 (366), 427-431.

Dickey, D. A., Hasza, D. P., and Fuller, W.A. (1984), "Testing for Unit Roots in Seasonal Time Series," *Journal of the American Statistical Association*, 79 (386), 355-367.

Dunsmuir, William (1984), "Large sample properties of estimation in time series observed at unequally spaced times," in *Time Series Analysis of Irregularly Observed Data*, Emanuel Parzen, ed., New York: Springer-Verlag.

Fuller, W.A. (1976), *Introduction to Statistical Time Series*, New York: John Wiley & Sons, Inc.

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

Hannan, E.J. and Rissanen, J. (1982), "Recursive Estimation of Mixed Autoregressive Moving Average Order," *Biometrika*, 69 (1), 81-94.

Harvey, A.C. (1981), *Time Series Models*, New York: John Wiley & Sons, Inc.

Jones, Richard H. (1980), "Maximum Likelihood Fitting of ARMA Models to Time Series with Missing Observations," *Technometrics*, 22, 389-396.

Ljung, G.M. and Box, G.E.P. (1978), "On a Measure of Lack of Fit in Time Series Models," *Biometrika*, 65, 297-303.

Montgomery, D.C. and Johnson, L.A. (1976), *Forecasting and Time Series Analysis*, New York: McGraw-Hill Book Co.

Morf, M., Sidhu, G.S., and Kailath, T. (1974), "Some New Algorithms for Recursive Estimation on Constant Linear Discrete Time Systems," *I.E.E.E. Transactions on Automatic Control*, AC-19, 315-323.

Nelson, C.R. (1973), *Applied Time Series for Managerial Forecasting*, San Francisco: Holden-Day.

Newbold, P. (1981), "Some Recent Developments in Time Series Analysis," *International Statistical Review*, 49, 53-66.

Newton, H. Joseph and Pagano, Marcello (1983), "The Finite Memory Prediction of Covariance Stationary Time Series," *SIAM Journal of Scientific and Statistical Computing*, 4, 330-339.

Pankratz, Alan (1983), *Forecasting with Univariate Box-Jenkins Models: Concepts and Cases*, New York: John Wiley & Sons, Inc.

Pankratz, Alan (1991), *Forecasting with Dynamic Regression Models*, New York: John Wiley & Sons, Inc.

Pearlman, J.G. (1980), "An Algorithm for the Exact Likelihood of a High-order Autoregressive-Moving Average Process," *Biometrika*, 67, 232-233.

Priestly, M.B. (1981), *Spectra Analysis and Time Series, Volume 1: Univariate Series*, New York: Academic Press, Inc.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *Annals of Statistics*, 6, 461-464.

Tsay, R.S. and Tiao, G.C. (1984), "Consistent Estimates of Autoregressive Parameters and Extended Sample Autocorrelation Function for Stationary and Nonstationary ARMA Models," *JASA*, 79 (385), 84-96.

Tsay, R.S. and Tiao, G.C. (1985), "Use of Canonical Analysis in Time Series Model Identification," *Biometrika*, 72 (2), 299-315.

Woodfield, T.J. (1987), "Time Series Intervention Analysis Using SAS Software," *Proceedings of the Twelfth Annual SAS Users Group International Conference,* 331-339. Cary, NC: SAS Institute Inc.