SAS/ETS® 14.1 User's Guide

The UCM Procedure

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

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

SAS/ETS® 14.1 User's Guide

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

July 2015

Chapter 41
The UCM Procedure

Contents

Overview: UCM Procedure

The UCM procedure analyzes and forecasts equally spaced univariate time series data by using an unobserved components model (UCM). The UCMs are also called *structural models* in the time series literature. A UCM decomposes the response series into components such as trend, seasonals, cycles, and the regression effects due to predictor series. The components in the model are supposed to capture the salient features of the series that are useful in explaining and predicting its behavior. Harvey (1989) is a good reference for time series modeling that uses the UCMs. Harvey calls the components in a UCM the “stylized facts” about the series under consideration. Traditionally, the ARIMA models and, to some limited extent, the exponential smoothing models have been the main tools in the analysis of this type of time series data. It is fair to say that the UCMs capture the versatility of the ARIMA models while possessing the interpretability of the smoothing models. A thorough discussion of the correspondence between the ARIMA models and the UCMs, and the relative merits of UCM and ARIMA modeling, is given in Harvey (1989). The UCMs are also very similar to another set of models, called the *dynamic models*, that are popular in the Bayesian time series literature (West and Harrison 1999). In SAS/ETS, you can use PROC SSM for multivariate (and more general univariate) UCMs (see Chapter 34, “The SSM Procedure”), PROC ARIMA for ARIMA modeling (see Chapter 8, “The ARIMA Procedure”), PROC ESM for exponential smoothing modeling (see Chapter 15, “The ESM Procedure”), and the Time Series Forecasting System for a point-and-click interface to ARIMA and exponential smoothing modeling.

You can use the UCM procedure to fit a wide range of UCMs that can incorporate complex trend, seasonal, and cyclical patterns and can include multiple predictors. It provides a variety of diagnostic tools to assess the fitted model and to suggest the possible extensions or modifications. The components in the UCM provide a succinct description of the underlying mechanism governing the series. You can print, save, or plot the estimates of these component series. Along with the standard forecast and residual plots, the study of these component plots is an essential part of time series analysis using the UCMs. Once a suitable UCM is found for the series under consideration, it can be used for a variety of purposes. For example, it can be used for the following:

- forecasting the values of the response series and the component series in the model
- obtaining a model-based seasonal decomposition of the series
- obtaining a "denoised" version and interpolating the missing values of the response series in the historical period
- obtaining the full sample or "smoothed" estimates of the component series in the model

Getting Started: UCM Procedure

The analysis of time series using the UCMs involves recognizing the salient features present in the series and modeling them suitably. The UCM procedure provides a variety of models for estimating and forecasting the commonly observed features in time series. These models are discussed in detail later in the section "An Introduction to Unobserved Component Models" on page 2782. First the procedure is illustrated using an example.

A Seasonal Series with Linear Trend

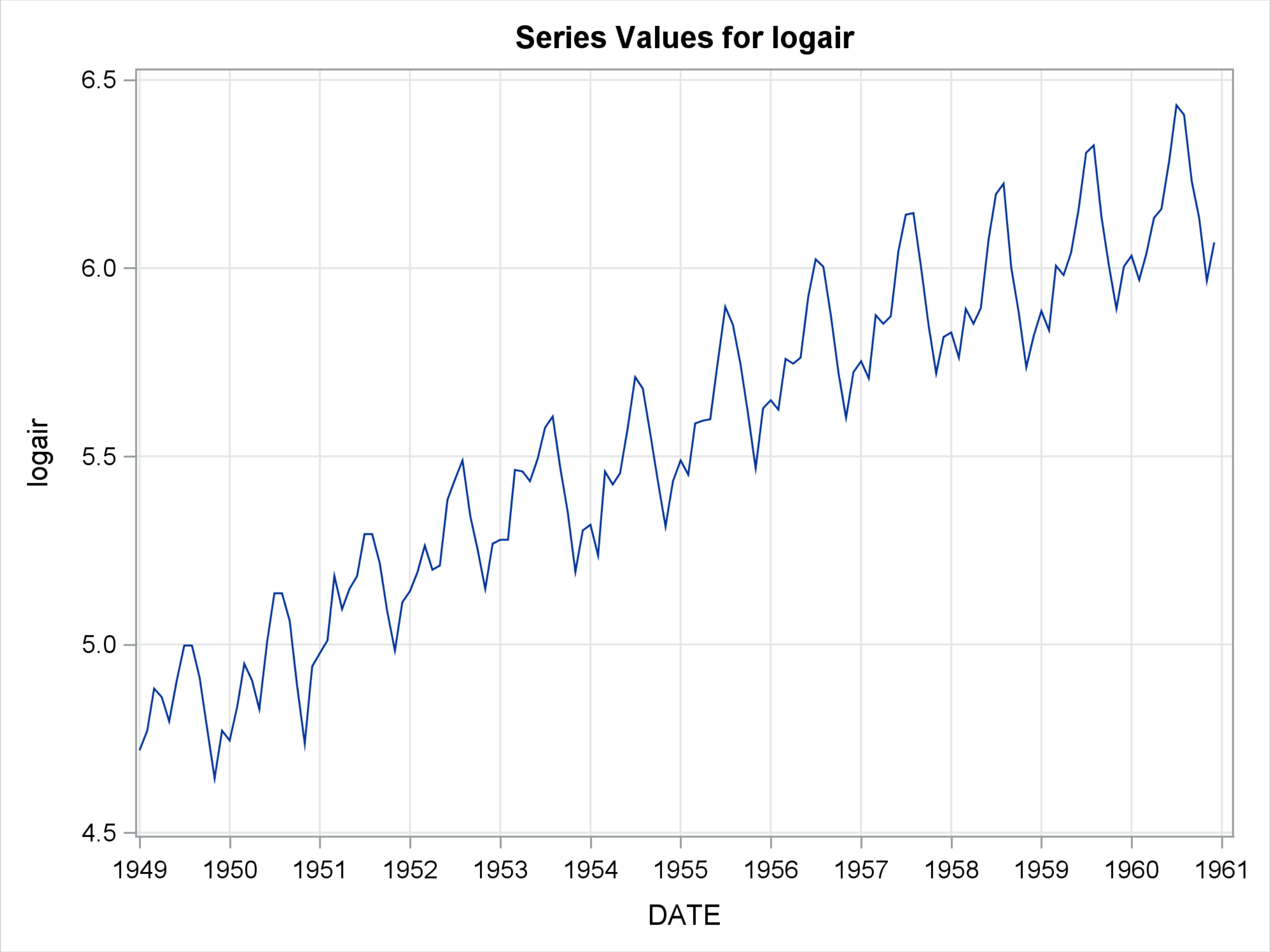
The airline passenger series, given as Series G in Box and Jenkins (1976), is often used in time series literature as an example of a nonstationary seasonal time series. This series is a monthly series consisting of the number of airline passengers who traveled during the years 1949 to 1960. Its main features are a steady rise in the number of passengers from year to year and the seasonal variation in the numbers during any given year. It also exhibits an increase in variability around the trend. A log transformation is used to stabilize this variability. The following DATA step prepares the log-transformed passenger series analyzed in this example:

```
data seriesG;
   set sashelp.air;
   logair = log( air );
run;
```

The following statements produce a time series plot of the series by using the TIMESERIES procedure (see Chapter 39, "The TIMESERIES Procedure"). The trend and seasonal features of the series are apparent in the plot in Figure 41.1.

```
proc timeseries data=seriesG plot=series;
   id date interval=month;
   var logair;
run;
```

Figure 41.1 Series Plot of Log-Transformed Airline Passenger Series



In this example this series is modeled using an unobserved component model called the basic structural model (BSM). The BSM models a time series as a sum of three stochastic components: a trend component μ_t, a seasonal component γ_t, and random error ϵ_t. Formally, a BSM for a response series y_t can be described as

$$y_t = \mu_t + \gamma_t + \epsilon_t$$

Each of the stochastic components in the model is modeled separately. The random error ϵ_t, also called the *irregular component*, is modeled simply as a sequence of independent, identically distributed (i.i.d.) zero-mean Gaussian random variables. The trend and the seasonal components can be modeled in a few different ways. The model for trend used here is called a *locally linear time trend*. This trend model can be written as follows:

$$\begin{aligned}
\mu_t &= \mu_{t-1} + \beta_{t-1} + \eta_t, \quad & \eta_t &\sim i.i.d.\ N(0, \sigma_\eta^2) \\
\beta_t &= \beta_{t-1} + \xi_t, & \xi_t &\sim i.i.d.\ N(0, \sigma_\xi^2)
\end{aligned}$$

These equations specify a trend where the level μ_t as well as the slope β_t is allowed to vary over time. This variation in slope and level is governed by the variances of the disturbance terms η_t and ξ_t in their respective equations. Some interesting special cases of this model arise when you manipulate these disturbance variances. For example, if the variance of ξ_t is zero, the slope will be constant (equal to β_0); if the variance of η_t is also zero, μ_t will be a deterministic trend given by the line $\mu_0 + \beta_0 t$. The seasonal model used in this example is called a trigonometric seasonal. The stochastic equations governing a trigonometric seasonal are explained later (see the section "Modeling Seasons" on page 2784). However, it is interesting to note here that this seasonal model reduces to the familiar regression with deterministic seasonal dummies if the variance of the disturbance terms in its equations is equal to zero. The following statements specify a BSM with these three components:

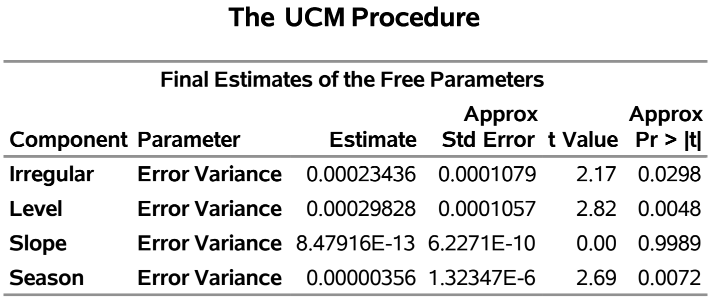
```
proc ucm data=seriesG;
   id date interval=month;
   model logair;
   irregular;
   level;
   slope;
   season length=12 type=trig print=smooth;
   estimate;
   forecast lead=24 print=decomp;
run;
```

The PROC UCM statement signifies the start of the UCM procedure, and the input data set, seriesG, containing the dependent series is specified there. The optional ID statement is used to specify a date, datetime, or time identification variable, date in this example, to label the observations. The INTERVAL=MONTH option in the ID statement indicates that the measurements were collected on a monthly basis. The model specification begins with the MODEL statement, where the response series is specified (logair in this case). After this the components in the model are specified using separate statements that enable you to control their individual properties. The irregular component ϵ_t is specified using the IRREGULAR statement and the trend component μ_t is specified using the LEVEL and SLOPE statements. The seasonal component γ_t is specified using the SEASON statement. The specifics of the seasonal characteristics such as the season length, its stochastic evolution properties, etc., are specified using the options in the SEASON statement. The seasonal component used in this example has a season length of 12, corresponding to the monthly seasonality, and is of the *trigonometric* type. Different types of seasonals are explained later (see the section "Modeling Seasons" on page 2784).

The parameters of this model are the variances of the disturbance terms in the evolution equations of μ_t, β_t, and γ_t and the variance of the irregular component ϵ_t. These parameters are estimated by maximizing the likelihood of the data. The ESTIMATE statement options can be used to specify the span of data used in parameter estimation and to display and save the results of the estimation step and the model diagnostics. You can use the estimated model to obtain the forecasts of the series as well as the components. The options in the individual component statements can be used to display the component forecasts—for example, PRINT=SMOOTH option in the SEASON statement requests the displaying of smoothed forecasts of the seasonal component γ_t. The series forecasts and forecasts of the sum of components can be requested using the FORECAST statement. The option PRINT=DECOMP in the FORECAST statement requests the printing of the smoothed trend μ_t and the trend plus seasonal component $(\mu_t + \gamma_t)$.

The parameter estimates for this model are displayed in Figure 41.2.

Figure 41.2 BSM for the Logair Series



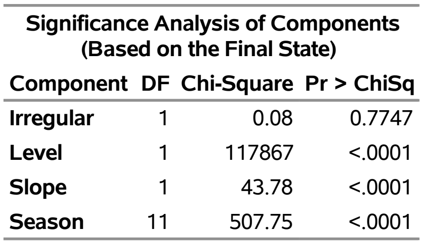
The UCM Procedure

Final Estimates of the Free Parameters

Component	Parameter	Estimate	Approx Std Error	t Value	Approx Pr > |t|
Irregular	**Error Variance**	0.00023436	0.0001079	2.17	0.0298
Level	**Error Variance**	0.00029828	0.0001057	2.82	0.0048
Slope	**Error Variance**	8.47916E-13	6.2271E-10	0.00	0.9989
Season	**Error Variance**	0.00000356	1.32347E-6	2.69	0.0072

The estimates suggest that except for the slope component, the disturbance variances of all the components are significant—that is, all these components are *stochastic*. The slope component, however, appears to be deterministic because its error variance is quite insignificant. It might then be useful to check if the slope component can be dropped from the model—that is, if $\beta_0 = 0$. This can be checked by examining the significance analysis table of the components given in Figure 41.3.

Figure 41.3 Component Significance Analysis for the Logair Series



Significance Analysis of Components (Based on the Final State)

Component	DF	Chi-Square	Pr > ChiSq
Irregular	1	0.08	0.7747
Level	1	117867	<.0001
Slope	1	43.78	<.0001
Season	11	507.75	<.0001

This table provides the significance of the components in the model at the end of the estimation span. If a component is deterministic, this analysis is equivalent to checking whether the corresponding regression effect is significant. However, if a component is stochastic, then this analysis pertains only to the portion of the series near the end of the estimation span. In this example the slope appears quite significant and should be retained in the model, possibly as a deterministic component. Note that, on the basis of this table, the irregular component's contribution appears insignificant toward the end of the estimation span; however, since it is a stochastic component, it cannot be dropped from the model on the basis of this analysis alone. The slope component can be made deterministic by holding the value of its error variance fixed at zero. This is done by modifying the SLOPE statement as follows:

```
slope variance=0 noest;
```

After a tentative model is fit, its adequacy can be checked by examining different goodness-of-fit measures and other diagnostic tests and plots that are based on the model residuals. Once the model appears satisfactory, it can be used for forecasting. An interesting feature of the UCM procedure is that, apart from the series forecasts, you can request the forecasts of the individual components in the model. The plots of component forecasts can be useful in understanding their contributions to the series. The following statements illustrate some of these features:

```
proc ucm data=seriesG;
   id date interval = month;
   model logair;
   irregular;
   level plot=smooth;
   slope variance=0 noest;
   season length=12 type=trig
       plot=smooth;
   estimate;
   forecast lead=24 plot=decomp;
run;
```

The table given in Figure 41.4 shows the goodness-of-fit statistics that are computed by using the one-step-ahead prediction errors (see the section "Statistics of Fit" on page 2820). These measures indicate a good agreement between the model and the data. Additional diagnostic measures are also printed by default but are not shown here.

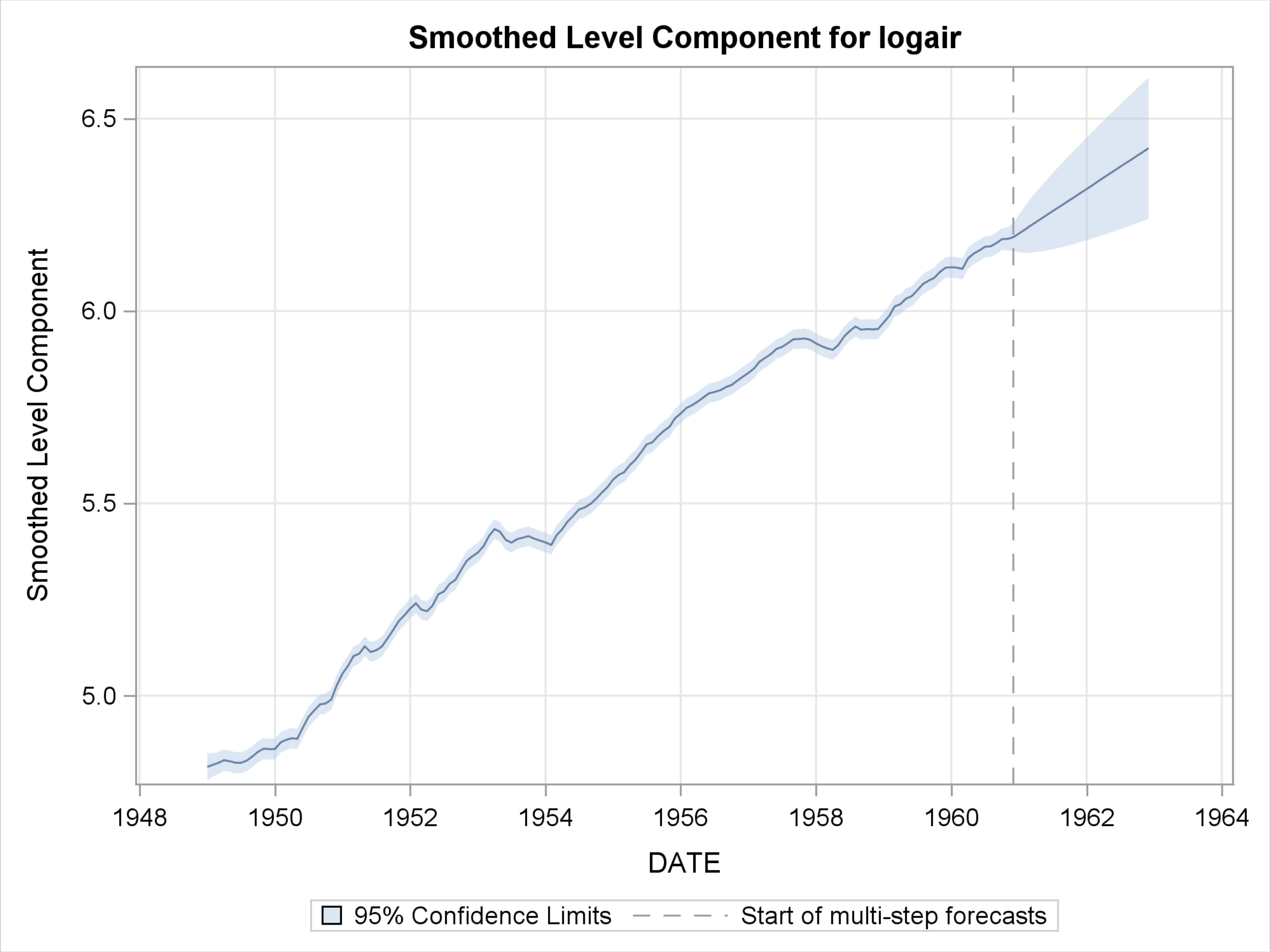
Figure 41.4 Fit Statistics for the Logair Series

The UCM Procedure

Fit Statistics Based on Residuals	
Mean Squared Error	0.00147
Root Mean Squared Error	0.03830
Mean Absolute Percentage Error	0.54132
Maximum Percent Error	2.19097
R-Square	0.99061
Adjusted R-Square	0.99046
Random Walk R-Square	0.87288
Amemiya's Adjusted R-Square	0.99017
Number of non-missing residuals used for computing the fit statistics = 131	

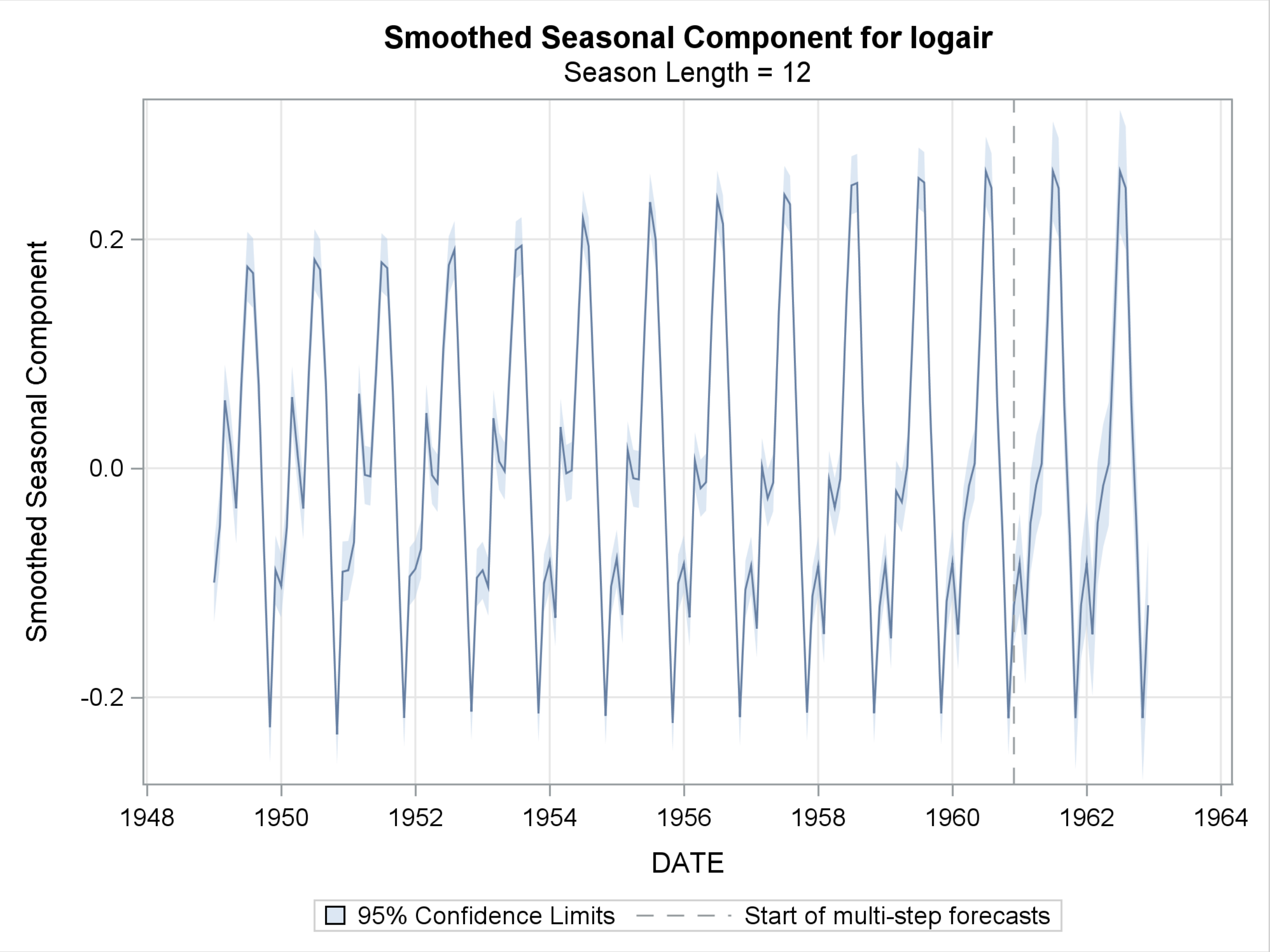
The first plot, shown in Figure 41.5, is produced by the PLOT=SMOOTH option in the LEVEL statement, it shows the smoothed level of the series.

Figure 41.5 Smoothed Trend in the Logair Series



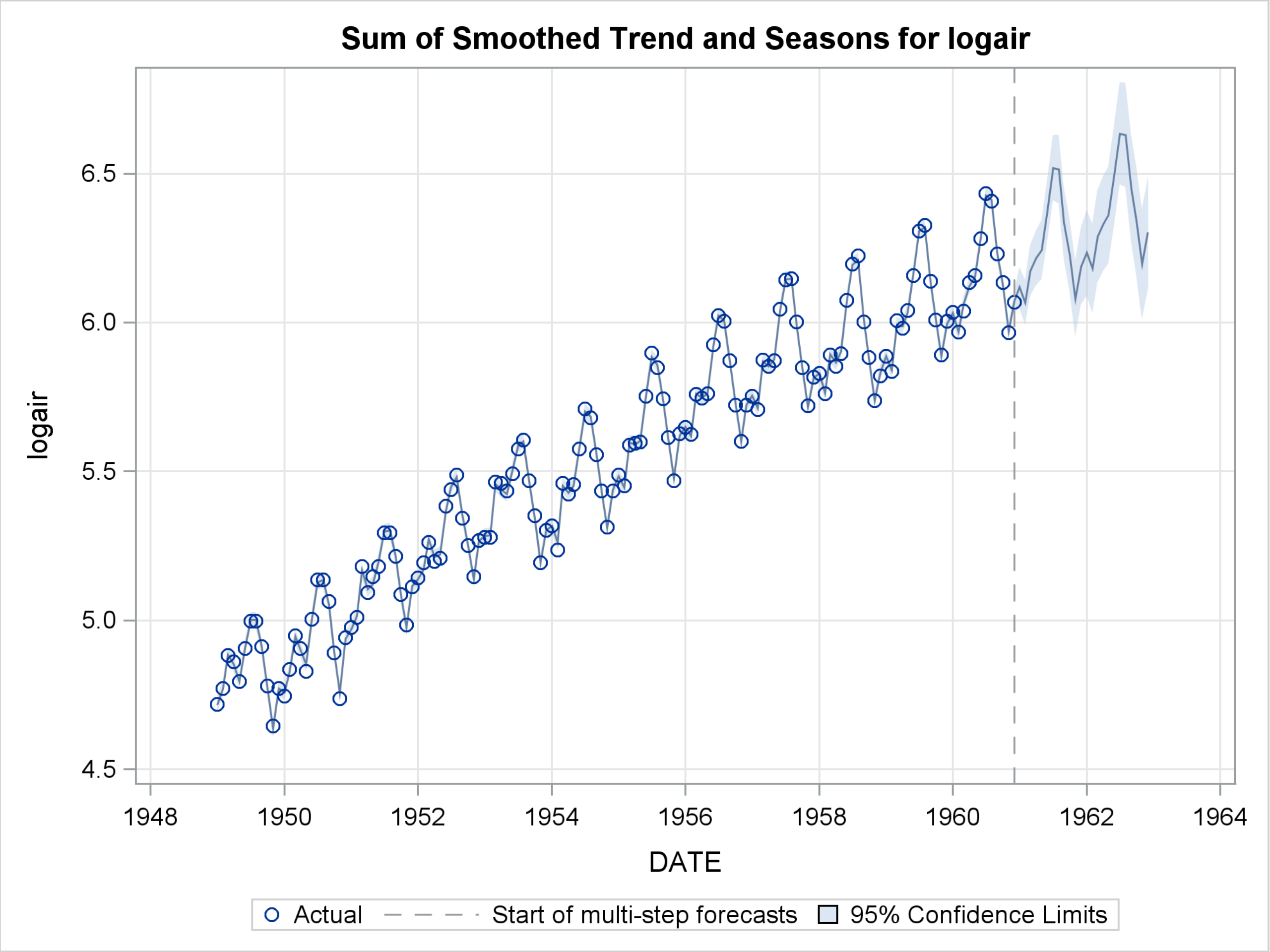
The second plot (Figure 41.6), produced by the PLOT=SMOOTH option in the SEASON statement, shows the smoothed seasonal component by itself.

Figure 41.6 Smoothed Seasonal in the Logair Series



The plot of the sum of the trend and seasonal component, produced by the PLOT=DECOMP option in the FORECAST statement, is shown in Figure 41.7. You can see that, at least visually, the model seems to fit the data well. In all these decomposition plots the component estimates are extrapolated for two years in the future based on the LEAD=24 option specified in the FORECAST statement.

Figure 41.7 Smoothed Trend plus Seasonal in the Logair Series



Syntax: UCM Procedure

The UCM procedure uses the following statements:

PROC UCM < *options* > ;
AUTOREG < *options* > ;
BLOCKSEASON *options* ;
BY *variables* ;
CYCLE < *options* > ;
DEPLAG *options* ;
ESTIMATE < *options* > ;
FORECAST < *options* > ;
ID *variable options* ;
IRREGULAR < *options* > ;
LEVEL < *options* > ;
MODEL *dependent variable* < = *regressors* > ;
NLOPTIONS *options* ;
PERFORMANCE *options* ;
OUTLIER *options* ;
RANDOMREG *regressors* < / *options* > ;
SEASON *options* ;
SLOPE < *options* > ;
SPLINEREG *regressor* < *options* > ;
SPLINESEASON *options* ;

The PROC UCM and MODEL statements are required. In addition, the model must contain at least one component with nonzero disturbance variance.

Functional Summary

The statements and options controlling the UCM procedure are summarized in the following table. Most commonly needed scenarios are listed; see the individual statements for additional details. You can use the PRINT= and PLOT= options in the individual component statements for printing and plotting the corresponding component forecasts.

Table 41.1 Functional Summary

Description	Statement	Option
Data Set Options		
specify the input data set	PROC UCM	DATA=
write parameter estimates to an output data set	ESTIMATE	OUTEST=
write series and component forecasts to an output data set	FORECAST	OUTFOR=

Table 41.1 *continued*

Description	Statement	Option
Model Specification		
specify the dependent variable and simple predictors	MODEL	
specify predictors with time-varying coefficients	RANDOMREG	
specify a nonlinear predictor	SPLINEREG	
specify the irregular component	IRREGULAR	
specify the random walk trend	LEVEL	
specify the locally linear trend	LEVEL and SLOPE	
specify a cycle component	CYCLE	
specify a dummy seasonal component	SEASON	TYPE=DUMMY
specify a trigonometric seasonal component	SEASON	TYPE=TRIG
drop some harmonics from a trigonometric seasonal component	SEASON	DROPH=
specify a list of harmonics to keep in a trigonometric seasonal component	SEASON	KEEPH=
specify a spline-season component	SPLINESEASON	
specify a block-season component	BLOCKSEASON	
specify an autoreg component	AUTOREG	
specify the lags of the dependent variable	DEPLAG	
Controlling the Likelihood Optimization Process		
request optimization of the profile likelihood	ESTIMATE	PROFILE
request optimization of the usual likelihood	ESTIMATE	NOPROFILE
specify the optimization technique	NLOPTIONS	TECH=
limit the number of iterations	NLOPTIONS	MAXITER=
Outlier Detection		
turn on the search for additive outliers		Default
turn on the search for level shifts	LEVEL	CHECKBREAK
specify the significance level for outlier tests	OUTLIER	ALPHA=
limit the number of outliers	OUTLIER	MAXNUM=
limit the number of outliers to a percentage of the series length	OUTLIER	MAXPCT=
Controlling the Series Span		
exclude some initial observations from analysis during the parameter estimation	ESTIMATE	SKIPFIRST=
exclude some observations at the end from analysis during the parameter estimation	ESTIMATE	BACK=
exclude some initial observations from analysis during forecasting	FORECAST	SKIPFIRST=
exclude some observations at the end from analysis during forecasting	FORECAST	BACK=

Table 41.1 *continued*

Description	Statement	Option
Graphical Residual Analysis		
get a panel of plots consisting of residual autocorrelation plots and residual normality plots	ESTIMATE	PLOT=PANEL
get the residual CUSUM plot	ESTIMATE	PLOT=CUSUM
get the residual cumulative sum of squares plot	ESTIMATE	PLOT=CUSUMSQ
get a plot of *p*-values for the portmanteau white noise test	ESTIMATE	PLOT=WN
get a time series plot of residuals with overlaid LOESS smoother	ESTIMATE	PLOT=LOESS
Series Decomposition and Forecasting		
specify the number of periods to forecast in the future	FORECAST	LEAD=
specify the significance level of the forecast confidence interval	FORECAST	ALPHA=
request printing of smoothed series decomposition	FORECAST	PRINT=DECOMP
request printing of one-step-ahead and multistep-ahead forecasts	FORECAST	PRINT=FORECASTS
request plotting of smoothed series decomposition	FORECAST	PLOT=DECOMP
request plotting of one-step-ahead and multistep-ahead forecasts	FORECAST	PLOT=FORECASTS
request bootstrap standard errors	FORECAST	BOOTSTRAP
BY Groups		
specify BY-group processing	BY	
Global Printing and Plotting Options		
turn off all the printing for the procedure	PROC UCM	NOPRINT
turn on all the printing options for the procedure	PROC UCM	PRINTALL
turn off all the plotting for the procedure	PROC UCM	PLOTS=NONE
turn on all the plotting options for the procedure	PROC UCM	PLOTS=ALL
turn on a variety of plotting options for the procedure	PROC UCM	PLOTS=
ID		
specify a variable that provides the time index for the series values	ID	

PROC UCM Statement

PROC UCM < *options* > **;**

The PROC UCM statement is required. The following options can be used in the PROC UCM statement:

DATA=*SAS-data-set*

specifies the name of the SAS data set containing the time series. If the DATA= option is not specified in the PROC UCM statement, the most recently created SAS data set is used.

NOPRINT

turns off all the printing for the procedure. The subsequent print options in the procedure are ignored.

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

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

controls the plots produced with ODS Graphics. When you specify only one plot request, you can omit the parentheses around the plot request.

Here are some examples:

```
plots=none
plots=all
plots=residuals(acf loess)
plots(noclm)=(smooth(decomp) residual(panel loess))
```

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

```
proc ucm;
   model y = x;
   irregular;
   level;
run;

proc ucm plots=all;
   model y = x;
   irregular;
   level;
run;
```

The first PROC UCM step does not specify the PLOTS= option, so the default plot that displays the series forecasts in the forecast region is produced. The PLOTS=ALL option in the second PROC UCM step produces all the plots that are appropriate for the specified model.

In addition to the PLOTS= option in the PROC UCM statement, you can request plots by using the PLOT= option in other statements of the UCM procedure. This way of requesting plots provides finer control over the plot production. If you do not specify any specific plot request, then PROC UCM produces the plot of series forecasts in the forecast horizon by default.

Global Plot Options:

The *global-plot-options* apply to all relevant plots generated by the UCM procedure. The following *global-plot-option* is supported:

NOCLM

suppresses the confidence limits in all the component and forecast plots.

Specific Plot Options:

The following list describes the specific plots and their options:

ALL

produces all plots appropriate for the particular analysis.

NONE

suppresses all plots.

FILTER (< *filter-plot-options* **>)**

produces time series plots of the filtered component estimates. The following *filter-plot-options* are available:

ALL

produces all the filtered component estimate plots appropriate for the particular analysis.

LEVEL

produces a time series plot of the filtered level component estimate, provided the model contains the level component.

SLOPE

produces a time series plot of the filtered slope component estimate, provided the model contains the slope component.

CYCLE

produces time series plots of the filtered cycle component estimates for all cycle components in the model, if there are any.

SEASON

produces time series plots of the filtered season component estimates for all seasonal components in the model, if there are any.

DECOMP

produces time series plots of the filtered estimates of the series decomposition.

RESIDUAL (< *residual-plot-options* **>)**

produces the residuals plots. The following *residual-plot-options* are available:

ALL

produces all the residual diagnostics plots appropriate for the particular analysis.

ACF

produces the autocorrelation plot of residuals.

CUSUM
produces the plot of cumulative residuals against time.

CUSUMSQ
produces the plot of cumulative squared residuals against time.

HISTOGRAM
produces the histogram of residuals.

LOESS
produces a scatter plot of residuals against time, which has an overlaid loess-fit.

PACF
produces the partial-autocorrelation plot of residuals.

PANEL
produces a summary panel of the residual diagnostics consisting of the following:

- histogram of residuals
- normal quantile plot of residuals
- the residual-autocorrelation-plot
- the residual-partial-autocorrelation-plot

QQ
produces a normal quantile plot of residuals.

RESIDUAL
produces a needle plot of residuals against time.

WN
produces the plot of Ljung-Box white-noise test *p*-values at different lags (in log scale).

SMOOTH (< ***smooth-plot-options*** **>)**
produces time series plots of the smoothed component estimates. The following *smooth-plot-options* are available:

ALL
produces all the smoothed component estimate plots appropriate for the particular analysis.

LEVEL
produces time series plot of the smoothed level component estimate, provided the model contains the level component.

SLOPE
produces time series plot of the smoothed slope component estimate, provided the model contains the slope component.

CYCLE
produces time series plots of the smoothed cycle component estimates for all cycle components in the model, if there are any.

SEASON
produces time series plots of the smoothed season component estimates for all season components in the model, if there are any.

DECOMP
produces time series plots of the smoothed estimates of the series decomposition.

PRINTALL
turns on all the printing options for the procedure. The subsequent NOPRINT options in the procedure are ignored.

AUTOREG Statement

AUTOREG < *options* > **;**

The AUTOREG statement specifies an autoregressive component in the model. An autoregressive component is a special case of cycle that corresponds to the frequency of zero or π. It is modeled separately for easier interpretation. A stochastic equation for an autoregressive component r_t can be written as follows:

$$r_t = \rho r_{t-1} + \nu_t, \quad \nu_t \sim i.i.d. \ N(0, \sigma_\nu^2)$$

The damping factor ρ can take any value in the interval (–1, 1), including –1 but excluding 1. If $\rho = 1$, the autoregressive component cannot be distinguished from the random walk level component. If $\rho = -1$, the autoregressive component corresponds to a seasonal component with a season length of 2, or a nonstationary cycle with period 2. If $|\rho| < 1$, then the autoregressive component is stationary. The following example illustrates the AUTOREG statement. This statement includes an autoregressive component in the model. The damping factor ρ and the disturbance variance σ_ν^2 are estimated from the data.

```
   autoreg;
```

NOEST=RHO

NOEST=VARIANCE

NOEST=(RHO VARIANCE)
fixes the values of ρ and σ_ν^2 to those specified in the RHO= and VARIANCE= options.

PLOT=FILTER

PLOT=SMOOTH

PLOT=(< FILTER > < SMOOTH >)
requests plotting of the filtered or smoothed estimate of the autoreg component.

PRINT=FILTER
PRINT=SMOOTH
PRINT=(< FILTER > < SMOOTH >)

requests printing of the filtered or smoothed estimate of the autoreg component.

RHO=*value*

specifies an initial value for the damping factor ρ during the parameter estimation process. The value of ρ must be in the interval (–1, 1), including –1 but excluding 1.

VARIANCE=*value*

specifies an initial value for the disturbance variance σ_ν^2 during the parameter estimation process. Any nonnegative value, including zero, is an acceptable starting value.

BLOCKSEASON Statement

BLOCKSEASON *NBLOCKS = integer BLOCKSIZE = integer < options >* **;**

The BLOCKSEASON or BLOCKSEASONAL statement is used to specify a seasonal component γ_t that has a special block structure. The seasonal γ_t is called a *block seasonal* of block size m and number of blocks k if its season length, s, can be factored as $s = m * k$ and its seasonal effects have a block form—that is, the first m seasonal effects are all equal to some number τ_1, the next m effects are all equal to some number τ_2, and so on.

This type of seasonal structure can be appropriate in some cases; for example, consider a series that is recorded on an hourly basis. Further assume that, in this particular case, the hour-of-the-day effect and the day-of-the-week effect are additive. In this situation the hour-of-the-week seasonality, having a season length of 168, can be modeled as a sum of two components. The hour-of-the-day effect is modeled using a simple seasonal of season length 24, while the day-of-the-week is modeled as a block seasonal component that has the days of the week as blocks. This day-of-the-week block seasonal component has seven blocks, each of size 24.

A block seasonal specification requires, at the minimum, the block size m and the number of blocks in the seasonal k. These are specified using the BLOCKSIZE= and NBLOCKS= option, respectively. In addition, you might need to specify the position of the first observation of the series by using the OFFSET= option if it is not at the beginning of one of the blocks. In the example just considered, this corresponds to a situation where the first series measurement is not at the start of the day. Suppose that the first measurement of the series corresponds to the hour between 6:00 and 7:00 a.m., which is the seventh hour within that day or at the seventh position within that block. This is specified as OFFSET=7.

The other options in this statement are very similar to the options in the SEASON statement; for example, a block seasonal can also be of one of the two types, DUMMY and TRIG. There can be more than one block seasonal component in the model, each specified using a separate BLOCKSEASON statement. No two block seasonals in the model can have the same NBLOCKS= and BLOCKSIZE= specifications. The following example illustrates the use of the BLOCKSEASON statement to specify the additive, hour-of-the-week seasonal model:

```
season length=24 type=trig;
blockseason nblocks=7 blocksize=24;
```

BLOCKSIZE=*integer*

specifies the block size, m. This is a required option in this statement. The block size can be any integer larger than or equal to two. Typical examples of block sizes are 24, corresponding to the hours of the day when a day is being used as a block in hourly data, or 60, corresponding to the minutes in an hour when an hour is being used as a block in data recorded by minutes, etc.

NBLOCKS=*integer*

specifies the number of blocks, k. This is a required option in this statement. The number of blocks can be any integer greater than or equal to two.

NOEST

fixes the value of the disturbance variance parameter to the value specified in the VARIANCE= option.

OFFSET=*integer*

specifies the position of the first measurement within the block, if the first measurement is not at the start of a block. The OFFSET= value must be between one and the block size. The default value is one. The first measurement refers to the start of the estimation span and the forecast span. If these spans differ, their starting measurements must be separated by an integer multiple of the block size.

PLOT=FILTER

PLOT=SMOOTH

PLOT=F_ANNUAL

PLOT=S_ANNUAL

PLOT=(< plot request > . . . < plot request >)

requests plots of the season component. When you specify only one plot request, you can omit the parentheses around the plot request. You can use the FILTER and SMOOTH options to plot the filtered and smoothed estimates of the season component γ_t. You can use the F_ANNUAL and S_ANNUAL options to get the plots of "annual" variation in the filtered and smoothed estimates of γ_t. The annual plots are useful to see the change in the contribution of a particular month over the span of years. Here "month" and "year" are generic terms that change appropriately with the interval type being used to label the observations and the season length. For example, for monthly data with a season length of 12, the usual meaning applies, while for daily data with a season length of 7, the days of the week serve as months and the weeks serve as years. The first period in each block is plotted over the years.

PRINT=FILTER

PRINT=SMOOTH

PRINT=(< FILTER > < SMOOTH >)

requests the printing of the filtered or smoothed estimate of the block seasonal component γ_t.

TYPE=DUMMY | TRIG

specifies the type of the block seasonal component. The default type is DUMMY.

VARIANCE=*value*

specifies an initial value for the disturbance variance, σ_ω^2, in the γ_t equation at the start of the parameter estimation process. Any nonnegative value, including zero, is an acceptable starting value.

BY Statement

BY *variables* **;**

A BY statement can be used in the UCM procedure to process a data set in groups of observations defined by the BY variables. The model specified using the MODEL and other component statements is applied to all the groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. The variables are one or more variables in the input data set.

CYCLE Statement

CYCLE *< options >* **;**

The CYCLE statement is used to specify a cycle component, ψ_t, in the model. The stochastic equation governing a cycle component of period *p* and damping factor ρ is as follows

$$\begin{bmatrix} \psi_t \\ \psi_t^* \end{bmatrix} = \rho \begin{bmatrix} \cos\lambda & \sin\lambda \\ -\sin\lambda & \cos\lambda \end{bmatrix} \begin{bmatrix} \psi_{t-1} \\ \psi_{t-1}^* \end{bmatrix} + \begin{bmatrix} \nu_t \\ \nu_t^* \end{bmatrix}$$

where ν_t and ν_t^* are independent, zero-mean, Gaussian disturbances with variance σ_ν^2 and $\lambda = 2 * \pi / p$ is the angular frequency of the cycle. Any *p* strictly greater than two is an admissible value for the period, and the damping factor ρ can be any value in the interval (0, 1), including one but excluding zero. The cycles with frequency zero and π, which correspond to the periods equal to infinity and two, respectively, can be specified using the AUTOREG statement. The values of ρ less than one give rise to a stationary cycle, while $\rho = 1$ gives rise to a nonstationary cycle. As a default, values of ρ, *p*, and σ_ν^2 are estimated from the data. However, if necessary, you can fix the values of some or all of these parameters.

There can be multiple cycles in a model, each specified using a separate CYCLE statement. The examples that follow illustrate the use of the CYCLE statement.

The following statements request including two cycles in the model. The parameters of each of these cycles are estimated from the data.

```
cycle;
cycle;
```

The following statement requests inclusion of a nonstationary cycle in the model. The cycle period *p* and the disturbance variance σ_ν^2 are estimated from the data.

```
cycle rho=1 noest=rho;
```

In the following statement a nonstationary cycle with a fixed period of 12 is specified. Moreover, a starting value is supplied for σ_ν^2.

```
cycle period=12 rho=1 variance=4 noest=(rho period);
```

NOEST=PERIOD

NOEST=RHO

NOEST=VARIANCE

NOEST=(< RHO > < PERIOD > < VARIANCE >)

fixes the values of the component parameters to those specified in the RHO=, PERIOD=, and VARIANCE= options. This option enables you to fix any combination of parameter values.

PERIOD=*value*

specifies an initial value for the cycle period during the parameter estimation process. Period value must be strictly greater than 2.

PLOT=FILTER

PLOT=SMOOTH

PLOT=(< FILTER > < SMOOTH >)

requests plotting of the filtered or smoothed estimate of the cycle component.

PRINT=FILTER

PRINT=SMOOTH

PRINT=(< FILTER > < SMOOTH >)

requests the printing of a filtered or smoothed estimate of the cycle component ψ_t.

RHO=*value*

specifies an initial value for the damping factor in this component during the parameter estimation process. Any value in the interval (0, 1), including one but excluding zero, is an acceptable initial value for the damping factor.

VARIANCE=*value*

specifies an initial value for the disturbance variance parameter, σ_ν^2, to be used during the parameter estimation process. Any nonnegative value, including zero, is an acceptable starting value.

DEPLAG Statement

DEPLAG *LAGS = order < PHI = value . . . > < NOEST>* **;**

The DEPLAG statement is used to specify the lags of the dependent variable to be included as predictors in the model. The following examples illustrate the use of DEPLAG statement.

If the dependent series is denoted by y_t, the following statement specifies the inclusion of $\phi_1 y_{t-1} + \phi_2 y_{t-2}$ in the model. The parameters ϕ_1 and ϕ_2 are estimated from the data.

```
deplag lags=2;
```

The following statement requests including $\phi_1 y_{t-1} + \phi_2 y_{t-4} - \phi_1 \phi_2 y_{t-5}$ in the model. The values of ϕ_1 and ϕ_2 are fixed at 0.8 and –1.2.

```
deplag lags=(1)(4) phi=0.8 -1.2 noest;
```

The dependent lag parameters are not constrained to lie in any particular region. In particular, this implies that a UCM that contains only an irregular component and dependent lags, resulting in a traditional autoregressive model, is not constrained to be a stationary model. In the DEPLAG statement, if an initial value is supplied for any one of the parameters, the initial values must also be supplied for all other parameters.

LAGS=*order*

LAGS=(*lag, . . . , lag* **) . . . (***lag, . . . , lag* **)**

is a required option in this statement. LAGS=(l_1, l_2, . . . , l_k) defines a model with specified lags of the dependent variable included as predictors. LAGS=*order* is equivalent to LAGS=(1, 2, . . . , *order*).

A concatenation of parenthesized lists specifies a factored model. For example, LAGS=(1)(12) specifies that the lag values, 1, 12, and 13, corresponding to the following polynomial in the backward shift operator, be included in the model

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

Note that, in this case, the coefficient of the thirteenth lag is constrained to be the product of the coefficients of the first and twelfth lags.

NOEST

fixes the values of the parameters to those specified in PHI= option.

PHI=*value* **. . .**

lists starting values for the coefficients of the lagged dependent variable. The order of the values listed corresponds with the order of the lags specified in the LAGS= option.

ESTIMATE Statement

ESTIMATE *< options>* **;**

The ESTIMATE statement is an optional statement used to control the overall model-fitting environment. Using this statement, you can control the span of observations used to fit the model by using the SKIPFIRST= and BACK= options. This can be useful in model diagnostics. You can request a variety of goodness-of-fit statistics and other model diagnostic information including different residual diagnostic plots. Note that the ESTIMATE statement is not used to control the nonlinear optimization process itself. That is done using the NLOPTIONS statement, where you can control the number of iterations, choose between the different optimization techniques, and so on. You can save the estimated parameters and other related information in a data set by using the OUTEST= option. You can request the optimization of the profile likelihood, the likelihood obtained by concentrating out a disturbance variance, for parameter estimation by using the PROFILE option. The following example illustrates the use of this statement:

```
estimate skipfirst=12 back=24;
```

This statement requests that the initial 12 measurements and the last 24 measurements be excluded during the model-fitting process. The actual observation span used to fit the model is decided as follows: Suppose that n_0 and n_1 are the observation numbers of the first and the last nonmissing values of the response variable, respectively. As a result of SKIPFIRST=12 and BACK=24, the measurements between observation numbers $n_0 + 12$ and $n_1 - 24$ form the estimation span. Of course, the model fitting might not take place if there are insufficient data in the resulting span. The model fitting does not take place if there are regressors in the model that have missing values in the estimation span.

BACK=*integer*

SKIPLAST=*integer*

indicates that some ending part of the data needs to be ignored during the parameter estimation. This can be useful when you want to study the forecasting performance of a model on the observed data. BACK=10 results in skipping the last 10 measurements of the response series during the parameter estimation. The default is BACK=0.

EXTRADIFFUSE=*k*

enables continuation of the diffuse filtering iterations for *k* additional iterations beyond the first instance where the initialization of the diffuse state would have otherwise taken place. If the specified *k* is larger than the sample size, the diffuse iterations continue until the end of the sample. Note that one-step-ahead residuals are produced only after the diffuse state is initialized. Delaying the initialization leads to a reduction in the number of one-step-ahead residuals available for computing the residual diagnostic measures. This option is useful when you want to ignore the first few one-step-ahead residuals that often have large variance.

NOPROFILE

requests that the usual likelihood be optimized for parameter estimation. For more information, see the section "Parameter Estimation by Profile Likelihood Optimization" on page 2798.

OUTEST=*SAS-data-set*

specifies an output data set for the estimated parameters.

In the ESTIMATE statement, the PLOT= option is used to obtain different residual diagnostic plots. The different possibilities are as follows:

PLOT=ACF

PLOT=MODEL

PLOT=LOESS

PLOT=HISTOGRAM

PLOT=PACF

PLOT=PANEL

PLOT=QQ

PLOT=RESIDUAL

PLOT=WN

PLOT=(< plot request > . . . < plot request >)

requests different residual diagnostic plots. The different options are as follows:

ACF

produces the residual-autocorrelation plot.

CUSUM

produces the plot of cumulative residuals against time.

CUSUMSQ

produces the plot of cumulative squared residuals against time.

MODEL
produces the plot of one-step-ahead forecasts in the estimation span.

HISTOGRAM
produces the histogram of residuals.

LOESS
produces a scatter plot of residuals against time, which has an overlaid loess-fit.

PACF
produces the residual-partial-autocorrelation plot.

PANEL
produces a summary panel of the residual diagnostics consisting of

- histogram of residuals
- normal quantile plot of residuals
- the residual-autocorrelation-plot
- the residual-partial-autocorrelation-plot

QQ
produces a normal quantile plot of residuals.

RESIDUAL
produces a needle plot of residuals against time.

WN
produces a plot of p-values, in log-scale, at different lags for the Ljung-Box portmanteau white noise test statistics.

PRINT=NONE
suppresses all the printed output related to the model fitting, such as the parameter estimates, the goodness-of-fit statistics, and so on.

PROFILE
requests that the profile likelihood, obtained by concentrating out one of the disturbance variances from the likelihood, be optimized for parameter estimation. By default, the profile likelihood is not optimized if any of the disturbance variance parameters is held fixed to a nonzero value. For more information see the section "Parameter Estimation by Profile Likelihood Optimization" on page 2798.

SKIPFIRST=*integer*
indicates that some early part of the data needs to be ignored during the parameter estimation. This can be useful if there is a reason to believe that the model being estimated is not appropriate for this portion of the data. SKIPFIRST=10 results in skipping the first 10 measurements of the response series during the parameter estimation. The default is SKIPFIRST=0.

FORECAST Statement

FORECAST < *options* > **;**

The FORECAST statement is an optional statement that is used to specify the overall forecasting environment for the specified model. It can be used to specify the span of observations, the historical period, to use to compute the forecasts of the future observations. This is done using the SKIPFIRST= and BACK= options. The number of periods to forecast beyond the historical period, and the significance level of the forecast confidence interval, is specified using the LEAD= and ALPHA= options. You can request one-step-ahead series and component forecasts by using the PRINT= option. You can save the series forecasts, and the model-based decomposition of the series, in a data set by using the OUTFOR= option. You can use the BOOTSTRAP option to request the computation of bootstrap prediction standard errors and the associated confidence intervals. The following example illustrates the use of this statement:

```
   forecast skipfirst=12 back=24 lead=30;
```

This statement requests that the initial 12 and the last 24 response values be excluded during the forecast computations. The forecast horizon, specified using the LEAD= option, is 30 periods; that is, multistep forecasting begins at the end of the historical period and continues for 30 periods. The actual observation span used to compute the multistep forecasting is decided as follows: Suppose that n_0 and n_1 are the observation numbers of the first and the last nonmissing values of the response variable, respectively. As a result of SKIPFIRST=12 and BACK=24, the historical period, or the forecast span, begins at $n_0 + 12$ and ends at $n_1 - 24$. Multistep forecasts are produced for the next 30 periods—that is, for the observation numbers $n_1 - 23$ to $n_1 + 6$. Of course, the forecast computations can fail if the model has regressor variables that have missing values in the forecast span. If the regressors contain missing values in the forecast horizon—that is, between the observations $n_1 - 23$ and $n_1 + 6$—the forecast horizon is reduced accordingly.

ALPHA=*value*

specifies the significance level of the forecast confidence intervals; for example, ALPHA=0.05, which is the default, results in a 95% confidence interval.

BACK=*integer*

SKIPLAST=*integer*

specifies the holdout sample for the evaluation of the forecasting performance of the model. For example, BACK=10 results in treating the last 10 observed values of the response series as unobserved. A post-sample-prediction-analysis table is produced for comparing the predicted values with the actual values in the holdout period. The default is BACK=0.

BOOTSTRAP(NREP=*integer* **< SEED=***integer* **>)** (Experimental)

enables the computation of bootstrap prediction standard errors based on the specified number of replications (NREP). The value of NREP must be at least 2. Optionally, you can specify the random number seed that is associated with the first replication by using the SEED= option. The seeds for the subsequent replications are assigned sequentially. The default seed value that is associated with the first replication is 123. The BOOTSTRAP option has no effect if the number of parameters to be estimated is zero (that is, all the model parameters are known). Note that this option is computationally expensive. The computational cost of NREP replications is comparable to the cost of estimating parameters NREP times.

EXTRADIFFUSE=*k*

enables continuation of the diffuse filtering iterations for *k* additional iterations beyond the first instance where the initialization of the diffuse state would have otherwise taken place. If the specified *k* is larger than the sample size, the diffuse iterations continue until the end of the sample. Note that one-step-ahead forecasts are produced only after the diffuse state is initialized. Delaying the initialization leads to reduction in the number of one-step-ahead forecasts. This option is useful when you want to ignore the first few one-step-ahead forecasts that often have large variance.

LEAD=*integer*

specifies the number of periods to forecast beyond the historical period defined by the SKIPFIRST= and BACK= options; for example, LEAD=10 results in the forecasting of 10 future values of the response series. The default is LEAD=12.

OUTFOR=*SAS-data-set*

specifies an output data set for the forecasts. The output data set contains the ID variable (if specified), the response and predictor series, the one-step-ahead and out-of-sample response series forecasts, the forecast confidence intervals, the smoothed values of the response series, and the smoothed forecasts produced as a result of the model-based decomposition of the series.

PLOT=DECOMP

PLOT=DECOMPVAR

PLOT=FDECOMP

PLOT=FDECOMPVAR

PLOT=FORECASTS

PLOT=TREND

PLOT=(< plot request > . . . < plot request >)

requests forecast and model decomposition plots. The FORECASTS option provides the plot of the series forecasts, the TREND and DECOMP options provide the plots of the smoothed trend and other decompositions, the DECOMPVAR option can be used to plot the variance of these components, and the FDECOMP and FDECOMPVAR options provide the same plots for the filtered decomposition estimates and their variances.

PRINT=DECOMP

PRINT=FDECOMP

PRINT=FORECASTS

PRINT=NONE

PRINT=(< print request > . . . < print request >)

controls the printing of the series forecasts and the printing of smoothed model decomposition estimates. By default, the series forecasts are printed only for the forecast horizon specified by the LEAD= option; that is, the one-step-ahead predicted values are not printed. You can request forecasts for the entire forecast span by specifying the PRINT=FORECASTS option. Using PRINT=DECOMP, you can get smoothed estimates of the following effects: trend, trend plus regression, trend plus regression plus cycle, and sum of all components except the irregular. If some of these effects are absent in the model, then they are ignored. Similarly you can get filtered estimates of these effects by using PRINT=FDECOMP. You can use PRINT=NONE to suppress the printing of all the forecast output.

SKIPFIRST=*integer*

indicates that some early part of the data needs to be ignored during the forecasting calculations. This can be useful if there is a reason to believe that the model being used for forecasting is not appropriate for this portion of the data. SKIPFIRST=10 results in skipping the first 10 measurements of the response series during the forecast calculations. The default is SKIPFIRST=0.

ID Statement

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

The ID statement names a numeric variable that identifies observations in the input and output data sets. The ID variable's values are assumed to be SAS date, time, or datetime values. In addition, the ID statement specifies the frequency associated with the time series. The ID statement options also specify how the observations are aligned to form the time series. If the ID statement is specified, the INTERVAL= option must also be specified. If the ID statement is not specified, the observation number, with respect to the BY group, is used as the time ID. The values of the ID variable are extrapolated for the forecast observations based on the values of the INTERVAL= option.

ALIGN=*value*

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

INTERVAL=*value*

specifies the time interval between observations. This option is required in the ID statement. INTERVAL=*value* is used in conjunction with the ID variable to check that the input data are in order and have no gaps. The INTERVAL= option is also used to extrapolate the ID values past the end of the input data. For a complete discussion of the intervals supported, see Chapter 5, "Date Intervals, Formats, and Functions."

IRREGULAR Statement

IRREGULAR < *options*> **;**

The IRREGULAR statement includes an irregular component in the model. There can be at most one IRREGULAR statement in the model specification. The irregular component corresponds to the overall random error ϵ_t in the model. By default the irregular component is modeled as white noise—that is, as a sequence of independent, identically distributed, zero-mean, Gaussian random variables. However, you can also model it as an autoregressive moving average (ARMA) process. The options for specifying an ARMA model for the irregular component are given in a separate subsection: "ARMA Specification" on page 2770.

The options in this statement enable you to specify the model for the irregular component and to output its estimates. Two examples of the IRREGULAR statement are given next. In the first example the statement is in its simplest form, resulting in the inclusion of an irregular component that is white noise with unknown variance:

```
irregular;
```

The following statement provides a starting value for the white noise variance σ_{ϵ}^2 to be used in the nonlinear parameter estimation process. It also requests the printing of smoothed estimates of ϵ_t. The smoothed irregulars are useful in model diagnostics.

```
irregular variance=4 print=smooth;
```

NOEST

fixes the value of σ_{ϵ}^2 to the value specified in the VARIANCE= option. Also see the NOEST= option in the subsection “ARMA Specification” on page 2770.

PLOT=FILTER
PLOT=SMOOTH
PLOT=(< FILTER > < SMOOTH >)

requests plotting of the filtered or smoothed estimate of the irregular component.

PRINT=FILTER
PRINT=SMOOTH
PRINT=(< FILTER > < SMOOTH >)

requests printing of the filtered or smoothed estimate of the irregular component.

VARIANCE=*value*

specifies an initial value for σ_{ϵ}^2 during the parameter estimation process. Any nonnegative value, including zero, is an acceptable starting value.

ARMA Specification

This section details the options for specifying an ARMA model for the irregular component. The specification of ARMA models requires some notation, which is explained first.

Let B denote the backshift operator—that is, for any sequence ϵ_t, $B\epsilon_t = \epsilon_{t-1}$. The higher powers of B represent larger shifts (for example, $B^3\epsilon_t = \epsilon_{t-3}$). A random sequence ϵ_t follows a zero-mean ARMA$(p,q)\times(P,Q)_s$ model with nonseasonal autoregressive order p, seasonal autoregressive order P, nonseasonal moving average order q, and seasonal moving average order Q, if it satisfies the following difference equation specified in terms of the polynomials in the backshift operator where a_t is a white noise sequence and s is the season length:

$$\phi(B)\Phi(B^s)\epsilon_t = \theta(B)\Theta(B^s)a_t$$

The polynomials ϕ, Φ, θ, and Θ are of orders p, P, q, and Q, respectively, which can be any nonnegative integers. The season length s must be a positive integer. For example, ϵ_t satisfies an ARMA(1,1) model (that is, $p = 1, q = 1$, $P = 0$, and $Q = 0$) if

$$\epsilon_t = \phi_1\epsilon_{t-1} + a_t - \theta_1 a_{t-1}$$

for some coefficients ϕ_1 and θ_1 and a white noise sequence a_t. Similarly ϵ_t satisfies an ARMA(1,1)×(1,1)$_{12}$ model if

$$\epsilon_t = \phi_1\epsilon_{t-1} + \Phi_1\epsilon_{t-12} - \phi_1\Phi_1\epsilon_{t-13} + a_t - \theta_1 a_{t-1} - \Theta_1 a_{t-12} + \theta_1\Theta_1 a_{t-13}$$

for some coefficients ϕ_1, Φ_1, θ_1, and Θ_1 and a white noise sequence a_t. The ARMA process is stationary and invertible if the defining polynomials ϕ, Φ, θ, and Θ have all their roots outside the unit circle—that is, their absolute values are strictly larger than 1.0. It is assumed that the ARMA model specified for the irregular component is stationary and invertible—that is, the coefficients of the polynomials ϕ, Φ, θ, and Θ are constrained so that the stationarity and invertibility conditions are satisfied. The unknown coefficients of these polynomials become part of the model parameter vector that is estimated using the data.

The notation for a closely related class of models, autoregressive integrated moving average (ARIMA) models, is also given here. A random sequence y_t is said to follow an ARIMA(p,d,q)×(P,D,Q)$_s$ model if, for some nonnegative integers d and D, the differenced series $\epsilon_t = (1-B)^d(1-B^s)^D y_t$ follows an ARMA(p,q)×(P,Q)$_s$ model. The integers d and D are called nonseasonal and seasonal differencing orders, respectively. You can specify ARIMA models by using the DEPLAG statement for specifying the differencing orders and by using the IRREGULAR statement for the ARMA specification. See Example 41.8 for an example of ARIMA(0,1,1)×(0,1,1)$_{12}$ model specification. Brockwell and Davis (1991) can be consulted for additional information about ARIMA models.

You can use options of the IRREGULAR statement to specify the desired ARMA model and to request printed and graphical output. A few examples of the IRREGULAR statement are given next.

The following statement specifies an irregular component that is modeled as an ARMA(1,1) process. It also requests plotting its smoothed estimate.

```
   irregular p=1 q=1 plot=smooth;
```

The following statement specifies an ARMA(1,1)×(1,1)$_{12}$ model. It also fixes the coefficient of the first-order seasonal moving average polynomial to 0.1. The other coefficients and the white noise variance are estimated using the data.

```
   irregular p=1 sp=1 q=1 sq=1 s=12 sma=0.1 noest=(sma);
```

AR=$\phi_1\ \phi_2 \ldots \phi_p$

lists the starting values of the coefficients of the nonseasonal autoregressive polynomial

$$\phi(B) = 1 - \phi_1 B - \ldots - \phi_p B^p$$

where the order p is specified in the P= option. The coefficients ϕ_i must define a stationary autoregressive polynomial.

MA=$\theta_1\ \theta_2 \ldots \theta_q$

lists the starting values of the coefficients of the nonseasonal moving average polynomial

$$\theta(B) = 1 - \theta_1 B - \ldots - \theta_q B^q$$

where the order q is specified in the Q= option. The coefficients θ_i must define an invertible moving average polynomial.

NOEST=(<VARIANCE> <AR> <SAR> <MA> <SMA>)

fixes the values of the ARMA parameters and the value of the white noise variance to those specified in the AR=, SAR=, MA=, SMA=, or VARIANCE= options.

P=*integer*

specifies the order of the nonseasonal autoregressive polynomial. The order can be any nonnegative integer; the default value is 0. In practice the order is a small integer such as 1, 2, or 3.

Q=*integer*

specifies the order of the nonseasonal moving average polynomial. The order can be any nonnegative integer; the default value is 0. In practice the order is a small integer such as 1, 2, or 3.

S=*integer*

specifies the season length used during the specification of the seasonal autoregressive or seasonal moving average polynomial. The season length can be any positive integer; for example, S=4 might be an appropriate value for a quarterly series. The default value is S=1.

SAR=Φ_1 Φ_2 **. . .** Φ_P

lists the starting values of the coefficients of the seasonal autoregressive polynomial

$$\Phi(B^s) = 1 - \Phi_1 B^s - \ldots - \Phi_P B^{sP}$$

where the order P is specified in the SP= option and the season length s is specified in the S= option. The coefficients Φ_i must define a stationary autoregressive polynomial.

SMA=Θ_1 Θ_2 **. . .** Θ_Q

lists the starting values of the coefficients of the seasonal moving average polynomial

$$\Theta(B^s) = 1 - \Theta_1 B^s - \ldots - \Theta_Q B^{sQ}$$

where the order Q is specified in the SQ= option and the season length s is specified in the S= option. The coefficients Θ_i must define an invertible moving average polynomial.

SP=*integer*

specifies the order of the seasonal autoregressive polynomial. The order can be any nonnegative integer; the default value is 0. In practice the order is a small integer such as 1 or 2.

SQ=*integer*

specifies the order of the seasonal moving average polynomial. The order can be any nonnegative integer; the default value is 0. In practice the order is a small integer such as 1 or 2.

LEVEL Statement

LEVEL < *options* > **;**

The LEVEL statement is used to include a level component in the model. The level component, either by itself or together with a slope component (see the SLOPE statement), forms the trend component, μ_t, of the model. If the slope component is absent, the resulting trend is a random walk (RW) specified by the following equations:

$$\mu_t = \mu_{t-1} + \eta_t, \quad \eta_t \sim i.i.d.\ N(0, \sigma_\eta^2)$$

If the slope component is present, signified by the presence of a SLOPE statement, a locally linear trend (LLT) is obtained. The equations of LLT are as follows:

$$\begin{array}{rclrcl} \mu_t & = & \mu_{t-1} + \beta_{t-1} + \eta_t, & \eta_t & \sim & i.i.d.\ N(0, \sigma_\eta^2) \\ \beta_t & = & \beta_{t-1} + \xi_t, & \xi_t & \sim & i.i.d.\ N(0, \sigma_\xi^2) \end{array}$$

In either case, the options in the LEVEL statement are used to specify the value of σ_η^2 and to request forecasts of μ_t. The SLOPE statement is used for similar purposes in the case of slope β_t. The following examples illustrate the use of the LEVEL statement. Assuming that a SLOPE statement is not added subsequently, a simple random walk trend is specified by the following statement:

```
   level;
```

The following statements specify a locally linear trend with value of σ_η^2 fixed at 4. It also requests printing of filtered values of μ_t. The value of σ_ξ^2, the disturbance variance in the slope equation, is estimated from the data.

```
   level variance=4 noest print=filter;
   slope;
```

CHECKBREAK

turns on the checking of breaks in the level component.

NOEST

fixes the value of σ_η^2 to the value specified in the VARIANCE= option.

PLOT=FILTER

PLOT=SMOOTH

PLOT=(< FILTER > < SMOOTH >)

requests plotting of the filtered or smoothed estimate of the level component.

PRINT=FILTER

PRINT=SMOOTH

PRINT=(< FILTER > < SMOOTH >)

requests printing of the filtered or smoothed estimate of the level component.

VARIANCE=*value*

specifies an initial value for σ_η^2, the disturbance variance in the μ_t equation at the start of the parameter estimation process. Any nonnegative value, including zero, is an acceptable starting value.

MODEL Statement

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

The MODEL statement specifies the response variable and, optionally, the predictor or regressor variables for the UCM model. This is a required statement in the UCM procedure. The predictors specified in the MODEL statement are assumed to have a linear and time-invariant relationship with the response. The predictors that have time-varying regression coefficients are specified separately in the RANDOMREG statement. Similarly, the predictors that have a nonlinear effect on the response variable are specified separately in the SPLINEREG statement. Only one MODEL statement can be specified.

NLOPTIONS Statement

NLOPTIONS < *options* > **;**

PROC UCM uses the nonlinear optimization (NLO) subsystem to perform the nonlinear optimization of the likelihood function during the estimation of model parameters. You can use the NLOPTIONS statement to control different aspects of this optimization process. For most problems the default settings of the optimization process are adequate. However, in some cases it might be useful to change the optimization technique or to change the maximum number of iterations. This can be done by using the TECH= and MAXITER= options in the NLOPTIONS statement as follows:

```
nloptions tech=dbldog maxiter=200;
```

This sets the maximum number of iterations to 200 and changes the optimization technique to DBLDOG rather than the default technique, TRUREG, used in PROC UCM. A discussion of the full range of options that can be used with the NLOPTIONS statement is given in Chapter 7, “Nonlinear Optimization Methods.” In PROC UCM all these options are available except the options related to the printing of the optimization history. In this version of PROC UCM all the printed output from the NLO subsystem is suppressed.

OUTLIER Statement

OUTLIER < *options* > **;**

The OUTLIER statement enables you to control the reporting of the additive outliers (AO) and level shifts (LS) in the response series. The AOs are searched by default. You can turn on the search for LSs by using the CHECKBREAK option in the LEVEL statement.

ALPHA=*significance-level*
specifies the significance level for reporting the outliers. The default is 0.05.

MAXNUM=*number*
limits the number of outliers to search. The default is MAXNUM=5.

MAXPCT=*number*

is similar to the MAXNUM= option. In the MAXPCT= option you can limit the number of outliers to search for according to a percentage of the series length. The default is MAXPCT=1. When both of these options are specified, the minimum of the two search numbers is used.

PRINT=SHORT | DETAIL

enables you to control the printed output of the outlier search. The PRINT=SHORT option, which is the default, produces an outlier summary table containing the most significant outliers, either AO or LS, discovered in the outlier search. The PRINT=DETAIL option produces, in addition to the outlier summary table, separate tables containing the AO and LS structural break chi-square statistics computed at each time point in the estimation span.

PERFORMANCE Statement

PERFORMANCE *options* **;**

The PERFORMANCE statement defines performance parameters for distributed and multithreaded computing and passes variables that describe the distributed computing environment. In the UCM procedure, this statement is applicable only if you specify the BOOTSTRAP option in the FORECAST statement. In addition, the number of nodes that you specify in the NODES= option in the PERFORMANCE statement must be strictly smaller than the number of bootstrap replications that you specify in the BOOTSTRAP option. The following statements illustrate how you can use this statement to perform bootstrap computations that use 10 nodes on a grid named *hpa.sas.com*:

```
proc ucm data=seriesG;
   id date interval=month;
   model logair;
   irregular;
   level;
   forecast lead=24 bootstrap(nrep=50 seed=1234);
   performance nodes=10 host="hpa.sas.com";
run;
```

For more information about the PERFORMANCE statement, see the section "PERFORMANCE Statement" on page 87 of Chapter 3, "Shared Concepts and Topics."

RANDOMREG Statement

RANDOMREG *regressors* < / *options* > **;**

The RANDOMREG statement is used to specify regressors with time-varying regression coefficients. Each regression coefficient—say, β_t— is assumed to evolve as a random walk:

$$\beta_t = \beta_{t-1} + \eta_t, \quad \eta_t \sim i.i.d.\ N(0, \sigma^2)$$

Of course, if the random walk disturbance variance σ^2 is zero, then the regression coefficient is not time varying, and it reduces to the standard regression setting. There can be multiple RANDOMREG statements, and each statement can contain one or more regressors. The regressors in a given RANDOMREG statement form a group that is assumed to share the same disturbance variance parameter. The random walks associated with different regressors are assumed to be independent. For an example of using this statement see Example 41.4. See the section "Reporting Parameter Estimates for Random Regressors" on page 2795 for additional information about the way parameter estimates are reported for this type of regressors.

NOEST

fixes the value of σ^2 to the value specified in the VARIANCE= option.

PLOT=FILTER

PLOT=SMOOTH

PLOT=(< FILTER > < SMOOTH >)

requests plotting of filtered or smoothed estimate of the time-varying regression coefficient.

PRINT=FILTER

PRINT=SMOOTH

PRINT=(< FILTER > < SMOOTH >)

requests printing of the filtered or smoothed estimate of the time-varying regression coefficient.

VARIANCE=*value*

specifies an initial value for σ^2 during the parameter estimation process. Any nonnegative value, including zero, is an acceptable starting value.

SEASON Statement

SEASON *LENGTH = integer < options >* **;**

The SEASON or SEASONAL statement is used to specify a seasonal component, γ_t, in the model. A seasonal component can be one of the two types, DUMMY or TRIG. A DUMMY seasonal with season length s satisfies the following stochastic equation:

$$\sum_{i=0}^{s-1} \gamma_{t-i} = \omega_t, \qquad \omega_t \sim i.i.d.\ N(0, \sigma_\omega^2)$$

The equations for a TRIG (short for trigonometric) seasonal component are as follows

$$\gamma_t = \sum_{j=1}^{[s/2]} \gamma_{j,t}$$

where $[s/2]$ equals $s/2$ if s is even and $(s-1)/2$ if it is odd. The sinusoids, also called *harmonics*, $\gamma_{j,t}$ have frequencies $\lambda_j = 2\pi j/s$ and are specified by the matrix equation

$$\begin{bmatrix} \gamma_{j,t} \\ \gamma_{j,t}^* \end{bmatrix} = \begin{bmatrix} \cos \lambda_j & \sin \lambda_j \\ -\sin \lambda_j & \cos \lambda_j \end{bmatrix} \begin{bmatrix} \gamma_{j,t-1} \\ \gamma_{j,t-1}^* \end{bmatrix} + \begin{bmatrix} \omega_{j,t} \\ \omega_{j,t}^* \end{bmatrix}$$

where the disturbances $\omega_{j,t}$ and $\omega_{j,t}^*$ are assumed to be independent and, for fixed j, $\omega_{j,t}$ and $\omega_{j,t}^* \sim N(0, \sigma_\omega^2)$. If s is even, then the equation for $\gamma_{s/2,t}^*$ is not needed and $\gamma_{s/2,t}$ is given by

$$\gamma_{s/2,t} = -\gamma_{s/2,t-1} + \omega_{s/2,t}$$

In the TRIG seasonal case, the option KEEPH= or DROPH= can be used to obtain *subset trigonometric* seasonals that contain only a subset of the full set of harmonics $\gamma_{j,t},\ j = 1, 2, \ldots, [s/2]$. This is particularly useful when the season length s is large and the seasonal pattern is relatively smooth.

Note that whether the seasonal type is DUMMY or TRIG, there is only one parameter, the disturbance variance σ^2_ω, in the seasonal model.

There can be more than one seasonal component in the model, necessarily with different season lengths if the seasons are full. You can have multiple *subset* season components with the same season length, if you need to use separate disturbance variances for different sets of harmonics. Each seasonal component is specified using a separate SEASON statement. A model with multiple seasonal components can easily become quite complex and might need a large amount of data and computing resources for its estimation and forecasting. The examples that follow illustrate the use of SEASON statement.

The following statement specifies a DUMMY type (default) seasonal component with a season length of four, corresponding to the quarterly seasonality. The disturbance variance σ^2_ω is estimated from the data.

```
season length=4;
```

The following statement specifies a trigonometric seasonal with monthly seasonality. It also provides a starting value for σ^2_ω.

```
season length=12 type=trig variance=4;
```

DROPHARMONICS|DROPH=*number-list* **|** *n* **TO** *m* **BY** *p*

enables you to drop some harmonics $\gamma_{j,t}$ from the full set of harmonics used to obtain a trigonometric seasonal. The drop list can include any integer between 1 and $[s/2]$, s being the season length. For example, the following specification results in a specification of a trigonometric seasonal with a season length 12 that consists of only the first four harmonics $\gamma_{j,t},\ j = 1, 2, 3, 4$:

```
season length=12 type=trig DROPH=5 6;
```

The last two *high* frequency harmonics are dropped. The DROPH= option cannot be used with the KEEPH= option.

KEEPHARMONICS|KEEPH=*number-list* **|** *n* **TO** *m* **BY** *p*

enables you to keep only the harmonics $\gamma_{j,t}$ listed in the option to obtain a trigonometric seasonal. The keep list can include any integer between 1 and $[s/2]$, s being the season length. For example, the following specification results in a specification of a trigonometric seasonal with a season length of 12 that consists of all the six harmonics $\gamma_{j,t},\ j = 1, \ldots 6$:

```
season length=12 type=trig KEEPH=1 to 3;
season length=12 type=trig KEEPH=4 to 6;
```

However, these six harmonics are grouped into two groups, each having its own disturbance variance parameter. The DROPH= option cannot be used with the KEEPH= option.

LENGTH=*integer*

specifies the season length, s. This is a required option in this statement. The season length can be any integer greater than or equal to 2. Typical examples of season lengths are 12, corresponding to the monthly seasonality, or 4, corresponding to the quarterly seasonality.

NOEST

fixes the value of the disturbance variance parameter to the value specified in the VARIANCE= option.

PLOT=FILTER

PLOT=SMOOTH

PLOT=F_ANNUAL

PLOT=S_ANNUAL

PLOT=(<plot request> . . . <plot request>)

requests plots of the season component. When you specify only one plot request, you can omit the parentheses around the plot request. You can use the FILTER and SMOOTH options to plot the filtered and smoothed estimates of the season component γ_t. You can use the F_ANNUAL and S_ANNUAL options to get the plots of "annual" variation in the filtered and smoothed estimates of γ_t. The annual plots are useful to see the change in the contribution of a particular month over the span of years. Here "month" and "year" are generic terms that change appropriately with the interval type being used to label the observations and the season length. For example, for monthly data with a season length of 12, the usual meaning applies, while for daily data with a season length of 7, the days of the week serve as months and the weeks serve as years.

PRINT=HARMONICS

requests printing of the summary of harmonics present in the seasonal component. This option is valid only for the trigonometric seasonal component.

PRINT=FILTER

PRINT=SMOOTH

PRINT=(< print request > . . . < print request >)

requests printing of the filtered or smoothed estimate of the seasonal component γ_t.

TYPE=DUMMY | TRIG

specifies the type of the seasonal component. The default type is DUMMY.

VARIANCE=*value*

specifies an initial value for the disturbance variance, σ^2_ω, in the γ_t equation at the start of the parameter estimation process. Any nonnegative value, including zero, is an acceptable starting value.

SLOPE Statement

SLOPE < *options* > ;

The SLOPE statement is used to include a slope component in the model. The slope component cannot be used without the level component (see the LEVEL statement). The level and slope specifications jointly define the trend component of the model. A SLOPE statement without the accompanying LEVEL statement is ignored. The equations of the trend, defined jointly by the level μ_t and slope β_t, are as follows:

$$\begin{aligned} \mu_t &= \mu_{t-1} + \beta_{t-1} + \eta_t, & \eta_t &\sim i.i.d.\ N(0, \sigma_\eta^2) \\ \beta_t &= \beta_{t-1} + \xi_t, & \xi_t &\sim i.i.d.\ N(0, \sigma_\xi^2) \end{aligned}$$

The SLOPE statement is used to specify the value of the disturbance variance, σ_ξ^2, in the slope equation, and to request forecasts of β_t. The following examples illustrate this statement:

```
level;
slope;
```

The preceding statements fit a model with a locally linear trend. The disturbance variances σ_η^2 and σ_ξ^2 are estimated from the data. You can request a locally linear trend with fixed slope by using the following statements:

```
level;
slope variance=0 noest;
```

NOEST

fixes the value of the disturbance variance, σ_ξ^2, to the value specified in the VARIANCE= option.

PLOT=FILTER

PLOT=SMOOTH

PLOT=(< FILTER > < SMOOTH >)

requests plotting of the filtered or smoothed estimate of the slope component.

PRINT=FILTER

PRINT=SMOOTH

PRINT=(< FILTER > < SMOOTH >)

requests printing of the filtered or smoothed estimate of the slope component β_t.

VARIANCE=*value*

specifies an initial value for the disturbance variance, σ_ξ^2, in the β_t equation at the start of the parameter estimation process. Any nonnegative value, including zero, is an acceptable starting value.

SPLINEREG Statement

SPLINEREG *regressor < options >* **;**

The SPLINEREG statement is used to specify a regressor that has a nonlinear relationship with the dependent series that can be approximated by a given B-spline. If the specified spline has degree d and is based on n internal knots, then it is known that it can be written as a linear combination of $(n + d + 1)$ regressors that are derived from the original regressor. The span of these $(n + d + 1)$ derived regressors includes constant; therefore, to avoid multicollinearity with the level component, one of these regressors is dropped. Specifying the SPLINEREG statement is equivalent to specifying a RANDOMREG statement with these derived regressors. There can be multiple SPLINEREG statements. You must specify at least one interior knot, either using the NKNOTS= option or the KNOTS= option. For more information about splines, see Chapter 117, "The TRANSREG Procedure" (*SAS/STAT User's Guide*). For an example of using this statement, see Example 41.6. See the section "Reporting Parameter Estimates for Random Regressors" on page 2795 for additional information about the way parameter estimates are reported for this type of regressors.

DEGREE=*integer*

specifies the degree of the spline. It can be any integer larger than or equal to zero. The default value is 3. The polynomial degree should be a small integer, usually 0, 1, 2, or 3. Larger values are rarely useful. If you have any doubt as to what degree to specify, use the default.

KNOTS=*number-list* **|** *n* **TO** *m* **BY** *p*

specifies the interior knots or break points. The values in the knot list must be nondecreasing and must lie between the minimum and the maximum of the spline regressor values in the input data set. The first time you specify a value in the knot list, it indicates a discontinuity in the nth (from DEGREE=n) derivative of the transformation function at the value of the knot. The second mention of a value indicates a discontinuity in the $(n-1)$th derivative of the transformation function at the value of the knot. Knots can be repeated any number of times for decreasing smoothness at the break points, but the values in the knot list can never decrease.

You cannot use the KNOTS= option with the NKNOTS= option. You should keep the number of knots small.

NKNOTS=*m*

creates m knots, the first at the $100/(m+1)$ percentile, the second at the $200/(m+1)$ percentile, and so on. Knots are always placed at data values; there is no interpolation. For example, if NKNOTS=3, knots are placed at the 25th percentile, the median, and the 75th percentile. The value specified for the NKNOTS= option must be ≥ 1. You cannot use the NKNOTS=option with the KNOTS= option.

NOTE: Specifying knots by using the NKNOTS= option can result in different sets of knots in the estimation and forecast stages if the distributions of regressor values in the estimation and forecast spans differ. The estimation span is based on the BACK= and SKIPFIRST= options in the ESTIMATE statement, and the forecast span is based on the BACK= and SKIPFIRST= options in the FORECAST statement.

NOEST

fixes the value of the regression coefficient random walk disturbance variance to the value specified in the VARIANCE= option.

PLOT=FILTER

PLOT=SMOOTH

PLOT=(< FILTER > < SMOOTH >)

requests plotting of filtered or smoothed estimate of the time-varying regression coefficient.

PRINT=FILTER

PRINT=SMOOTH

PRINT=(< FILTER > < SMOOTH >)

requests printing of filtered or smoothed estimate of the time-varying regression coefficient.

VARIANCE=*value*

specifies an initial value for the regression coefficient random walk disturbance variance during the parameter estimation process. Any nonnegative value, including zero, is an acceptable starting value.

SPLINESEASON Statement

SPLINESEASON *LENGTH = integer KNOTS=* $integer_1$ $integer_2$ $\ldots$ *< options >* **;**

The SPLINESEASON statement is used to specify a seasonal pattern that is to be approximated by a given B-spline. If the specified spline has degree d and is based on n internal knots, then it can be written as a linear combination of $(n + d)$ regressors that are derived from the seasonal dummy regressors. The SPLINESEASON specification is equivalent to specifying a RANDOMREG specification with these derived regressors. Such approximation is useful only if the season length is relatively large, at least larger than $(n + d)$. For additional information about splines, see Chapter 117, "The TRANSREG Procedure" (*SAS/STAT User's Guide*). For an example of using this statement, see Example 41.3.

DEGREE=*integer*

specifies the degree of the spline. It can be any integer greater than or equal to zero. The default value is 3.

KNOTS=$integer_1$ $integer_2$ **...**

lists the *internal* knots. This list of values must be a nondecreasing sequence of integers within the range of 2 to $(s - 1)$, where s is the season length specified in the LENGTH= option. This is a required option in this statement.

LENGTH=*integer*

specifies the season length, *s*. This is a required option in this statement. The length can be any integer greater than or equal to three.

NOEST

fixes the value of the regression coefficient random walk disturbance variance to the value specified in the VARIANCE= option.

OFFSET=*integer*

specifies the position of the first measurement within the season, if the first measurement is not at the start of the season. The OFFSET= value must be between one and the season length. The default value is one. The first measurement refers to the start of the estimation span and the forecast span. If these spans differ, their starting measurements must be separated by an integer multiple of the season length.

PLOT=FILTER

PLOT=SMOOTH

PLOT=(< FILTER > < SMOOTH >)

requests plots of the season component. When you specify only one plot request, you can omit the parentheses around the plot request. You can use the FILTER and SMOOTH options to plot the filtered and smoothed estimates of the season component.

PRINT=FILTER
PRINT=SMOOTH
PRINT=(< FILTER > < SMOOTH >)

requests the printing of the filtered or smoothed estimate of the spline season component.

RKNOTS=(*knot, . . . , knot* **) . . . (***knot, . . . , knot* **)**

specifies a grouping of knots such that the knots within the same group have identical seasonal values. The knots specified in this option must already be present in the list specified by the KNOTS= option. The knot groups must be non-overlapping and without any repeated knots.

VARIANCE=*value*

specifies an initial value for the regression coefficient random walk disturbance variance during the parameter estimation process. Any nonnegative value, including zero, is an acceptable starting value.

Details: UCM Procedure

An Introduction to Unobserved Component Models

A UCM decomposes the response series into components such as trend, seasons, cycles, and the regression effects due to predictor series. The following model shows a possible scenario:

$$
\begin{aligned}
y_t &= \mu_t + \gamma_t + \psi_t + \sum_{j=1}^{m} \beta_j x_{jt} + \epsilon_t \\
\epsilon_t &\sim i.i.d.\ N(0, \sigma_\epsilon^2)
\end{aligned}
$$

The terms μ_t, γ_t, and ψ_t represent the trend, seasonal, and cyclical components, respectively. In fact the model can contain multiple seasons and cycles, and the seasons can be of different types. For simplicity of discussion the preceding model contains only one of each of these components. The regression term, $\sum_{j=1}^{m} \beta_j x_{jt}$, includes contribution of regression variables with *fixed* regression coefficients. A model can also contain regression variables that have *time varying* regression coefficients or that have a nonlinear relationship with the dependent series (see "Incorporating Predictors of Different Kinds" on page 2794). The disturbance term ϵ_t, also called the *irregular* component, is usually assumed to be Gaussian white noise. In some cases it is useful to model the irregular component as a stationary ARMA process. See the section "Modeling the Irregular Component" on page 2786 for additional information.

By controlling the presence or absence of various terms and by choosing the proper flavor of the included terms, the UCMs can generate a rich variety of time series patterns. A UCM can be applied to variables after transforming them by transforms such as *log* and *difference*.

The components μ_t, γ_t, and ψ_t model structurally different aspects of the time series. For example, the trend μ_t models the natural tendency of the series in the absence of any other perturbing effects such as seasonality, cyclical components, and the effects of exogenous variables, while the seasonal component γ_t models the correction to the level due to the seasonal effects. These components are assumed to be statistically independent of each other and independent of the irregular component. All of the component models can be thought of as stochastic generalizations of the relevant deterministic patterns in time. This way the deterministic cases emerge as special cases of the stochastic models. The different models available for these unobserved components are discussed next.

Modeling the Trend

As mentioned earlier, the trend in a series can be loosely defined as the natural tendency of the series in the absence of any other perturbing effects. The UCM procedure offers two ways to model the trend component μ_t. The first model, called the random walk (RW) model, implies that the trend remains roughly constant throughout the life of the series without any persistent upward or downward drift. In the second model the trend is modeled as a locally linear time trend (LLT). The RW model can be described as

$$\mu_t = \mu_{t-1} + \eta_t, \quad \eta_t \sim i.i.d. \ N(0, \sigma_\eta^2)$$

Note that if $\sigma_\eta^2 = 0$, then the model becomes $\mu_t = constant$. In the LLT model the trend is locally linear, consisting of both the *level* and *slope*. The LLT model is

$$\begin{aligned} \mu_t &= \mu_{t-1} + \beta_{t-1} + \eta_t, & \eta_t &\sim i.i.d. \ N(0, \sigma_\eta^2) \\ \beta_t &= \beta_{t-1} + \xi_t, & \xi_t &\sim i.i.d. \ N(0, \sigma_\xi^2) \end{aligned}$$

The disturbances η_t and ξ_t are assumed to be independent. There are some interesting special cases of this model obtained by setting one or both of the disturbance variances σ_η^2 and σ_ξ^2 equal to zero. If σ_ξ^2 is set equal to zero, then you get a linear trend model with fixed slope. If σ_η^2 is set to zero, then the resulting model usually has a smoother trend. If both the variances are set to zero, then the resulting model is the deterministic linear time trend: $\mu_t = \mu_0 + \beta_0 t$.

You can incorporate these trend patterns in your model by using the LEVEL and SLOPE statements.

Modeling a Cycle

A deterministic cycle ψ_t with frequency λ, $0 < \lambda < \pi$, can be written as

$$\psi_t = \alpha \cos(\lambda t) + \beta \sin(\lambda t)$$

If the argument t is measured on a continuous scale, then ψ_t is a periodic function with period $2\pi/\lambda$, amplitude $\gamma = (\alpha^2 + \beta^2)^{1/2}$, and phase $\phi = \tan^{-1}(\beta/\alpha)$. Equivalently, the cycle can be written in terms of the amplitude and phase as

$$\psi_t = \gamma \cos(\lambda t - \phi)$$

Note that when ψ_t is measured only at the integer values, it is not exactly periodic, unless $\lambda = (2\pi j)/k$ for some integers j and k. The cycles in their pure form are not used very often in practice. However, they are very useful as building blocks for more complex periodic patterns. It is well known that the periodic pattern of any complexity can be written as a sum of pure cycles of different frequencies and amplitudes. In time series situations it is useful to generalize this simple cyclical pattern to a stochastic cycle that has a fixed period but time-varying amplitude and phase. The stochastic cycle considered here is motivated by the following recursive formula for computing ψ_t:

$$\begin{bmatrix} \psi_t \\ \psi_t^* \end{bmatrix} = \begin{bmatrix} \cos\lambda & \sin\lambda \\ -\sin\lambda & \cos\lambda \end{bmatrix} \begin{bmatrix} \psi_{t-1} \\ \psi_{t-1}^* \end{bmatrix}$$

starting with $\psi_0 = \alpha$ and $\psi_0^* = \beta$. Note that ψ_t and ψ_t^* satisfy the relation

$$\psi_t^2 + \psi_t^{*2} = \alpha^2 + \beta^2 \qquad \text{for all } \; t$$

A stochastic generalization of the cycle ψ_t can be obtained by adding random noise to this recursion and by introducing a damping factor, ρ, for additional modeling flexibility. This model can be described as follows

$$\begin{bmatrix} \psi_t \\ \psi_t^* \end{bmatrix} = \rho \begin{bmatrix} \cos\lambda & \sin\lambda \\ -\sin\lambda & \cos\lambda \end{bmatrix} \begin{bmatrix} \psi_{t-1} \\ \psi_{t-1}^* \end{bmatrix} + \begin{bmatrix} \nu_t \\ \nu_t^* \end{bmatrix}$$

where $0 \le \rho \le 1$, and the disturbances ν_t and ν_t^* are independent $N(0, \sigma_\nu^2)$ variables. The resulting stochastic cycle has a fixed period but time-varying amplitude and phase. The stationarity properties of the random sequence ψ_t depend on the damping factor ρ. If $\rho < 1$, ψ_t has a stationary distribution with mean zero and variance $\sigma_\nu^2/(1-\rho^2)$. If $\rho = 1$, ψ_t is nonstationary.

You can incorporate a cycle in a UCM by specifying a CYCLE statement. You can include multiple cycles in the model by using separate CYCLE statements for each included cycle.

As mentioned before, the cycles are very useful as building blocks for constructing more complex periodic patterns. Periodic patterns of almost any complexity can be created by superimposing cycles of different periods and amplitudes. In particular, the seasonal patterns, general periodic patterns with integer periods, can be constructed as sums of cycles. This important topic of modeling the seasonal components is considered next.

Modeling Seasons

The seasonal fluctuations are a common source of variation in time series data. These fluctuations arise because of the regular changes in seasons or some other periodic events. The seasonal effects are regarded as corrections to the general trend of the series due to the seasonal variations, and these effects sum to zero when summed over the full season cycle. Therefore the seasonal component γ_t is modeled as a stochastic periodic pattern of an integer period s such that the sum $\sum_{i=0}^{s-1} \gamma_{t-i}$ is always zero in the mean. The period s is called the season length. Two different models for the seasonal component are considered here. The first model is called the *dummy* variable form of the seasonal component. It is described by the equation

$$\sum_{i=0}^{s-1} \gamma_{t-i} = \omega_t, \qquad \omega_t \sim i.i.d. \; N(0, \sigma_\omega^2)$$

The other model is called the *trigonometric* form of the seasonal component. In this case γ_t is modeled as a sum of cycles of different frequencies. This model is given as follows

$$\gamma_t = \sum_{j=1}^{[s/2]} \gamma_{j,t}$$

where $[s/2]$ equals $s/2$ if s is even and $(s-1)/2$ if it is odd. The cycles $\gamma_{j,t}$ have frequencies $\lambda_j = 2\pi j/s$ and are specified by the matrix equation

$$\begin{bmatrix} \gamma_{j,t} \\ \gamma_{j,t}^* \end{bmatrix} = \begin{bmatrix} \cos\lambda_j & \sin\lambda_j \\ -\sin\lambda_j & \cos\lambda_j \end{bmatrix} \begin{bmatrix} \gamma_{j,t-1} \\ \gamma_{j,t-1}^* \end{bmatrix} + \begin{bmatrix} \omega_{j,t} \\ \omega_{j,t}^* \end{bmatrix}$$

where the disturbances $\omega_{j,t}$ and $\omega^*_{j,t}$ are assumed to be independent and, for fixed j, $\omega_{j,t}$ and $\omega^*_{j,t} \sim N(0, \sigma^2_\omega)$. If s is even, then the equation for $\gamma^*_{s/2,t}$ is not needed and $\gamma_{s/2,t}$ is given by

$$\gamma_{s/2,t} = -\gamma_{s/2,t-1} + \omega_{s/2,t}$$

The cycles $\gamma_{j,t}$ are called *harmonics*. If the seasonal component is deterministic, the decomposition of the seasonal effects into these harmonics is identical to its Fourier decomposition. In this case the sum of squares of the seasonal factors equals the sum of squares of the amplitudes of these harmonics. In many practical situations, the contribution of the high-frequency harmonics is negligible and can be ignored, giving rise to a simpler description of the seasonal. In the case of stochastic seasonals, the situation might not be so transparent; however, similar considerations still apply. Note that if the disturbance variance $\sigma^2_\omega = 0$, then both the dummy and the trigonometric forms of seasonal components reduce to constant seasonal effects. That is, the seasonal component reduces to a deterministic function that is completely determined by its first $s - 1$ values.

In the UCM procedure you can specify a seasonal component in a variety of ways, the SEASON statement being the simplest of these. The dummy and the trigonometric seasonal components discussed so far can be considered as *saturated* seasonal components that put no restrictions on the $s - 1$ seasonal values. In some cases a more parsimonious representation of the seasonal might be more appropriate. This is particularly useful for seasonal components with large season lengths. In the UCM procedure you can obtain parsimonious representations of the seasonal components by one of the following ways:

- Use a *subset* trigonometric seasonal component obtained by deleting a few of the $[s/2]$ harmonics used in its sum. For example, a slightly smoother seasonal component of length 12, corresponding to the monthly seasonality, can be obtained by deleting the highest-frequency harmonic of period 2. That is, such a seasonal component will be a sum of five stochastic cycles that have periods 12, 6, 4, 3, and 2.4. You can specify such subset seasonal components by using the KEEPH= or DROPH= option in the SEASON statement.

- Approximate the seasonal pattern by a suitable spline approximation. You can do this by using the SPLINESEASON statement.

- A *block-seasonal* pattern is a seasonal pattern where the pattern is divided into a few blocks of equal length such that the season values within a block are the same—for example, a monthly seasonal pattern that has only four different values, one for each quarter. In some situations a long seasonal pattern can be approximated by the sum of block season and a simple season, the length of the simple season being equal to the block length of the block season. You can obtain such approximation by using a combination of BLOCKSEASON and SEASON statements.

- Consider a seasonal component of a large season length as a sum of two or more seasonal components that are each of much smaller season lengths. This can be done by specifying more than one SEASON statements.

Note that the preceding techniques of obtaining parsimonious seasonal components can also enable you to specify seasonal components that are more *general* than the simple saturated seasonal components. For example, you can specify a saturated trigonometric seasonal component that has some of its harmonics evolving according to one disturbance variance parameter while the others evolve with another disturbance variance parameter.

Modeling an Autoregression

An autoregression of order one can be thought of as a special case of a cycle when the frequency λ is either 0 or π. Modeling this special case separately helps interpretation and parameter estimation. The autoregression component r_t is modeled as follows

$$r_t = \rho r_{t-1} + \nu_t, \quad \nu_t \sim i.i.d. \; N(0, \sigma_\nu^2)$$

where $-1 \leq \rho < 1$. An autoregression can also provide an alternative to the IRREGULAR component when the model errors show some autocorrelation. You can incorporate an autoregression in your model by using the AUTOREG statement.

Modeling Regression Effects

A predictor variable can affect the response variable in a variety of ways. The UCM procedure enables you to model several different types of predictor-response relationships:

- The predictor-response relationship is *linear*, and the regression coefficient does not change with time. This is the simplest kind of relationship and such predictors are specified in the MODEL statement.
- The predictor-response relationship is *linear*, but the regression coefficient does change with time. Such predictors are specified in the RANDOMREG statement. Here the regression coefficient is assumed to evolve as a random walk.
- The predictor-response relationship is *nonlinear* and the relationship can change with time. This type of relationship can be approximated by an appropriate time-varying spline. Such predictors are specified in the SPLINEREG statement.

A response variable can depend on its own past values—that is, lagged dependent values. Such a relationship can be specified in the DEPLAG statement.

Modeling the Irregular Component

The components—such as trend, seasonal and regression effects, and nonstationary cycles—are used to capture the structural dynamics of a response series. In contrast, the stationary cycles and the autoregression are used to capture the transient aspects of the response series that are important for its short-range prediction but have little impact on its long-term forecasts. The irregular component represents the residual variation remaining in the response series that is modeled using an appropriate selection of structural and transient effects. In most cases, the irregular component can be assumed to be simply Gaussian white noise. In some other cases, however, the residual variation can be more complicated. In such situations, it might be necessary to model the irregular component as a stationary ARMA process. Moreover, you can use the ARMA irregular component together with the dependent lag specification (see the DEPLAG statement) to specify an ARIMA$(p,d,q)\times(P,D,Q)_s$ model for the response series. See the IRREGULAR statement for the explanation of the ARIMA notation. See Example 41.8 for an example of modeling a series by using an ARIMA$(0,1,1)\times(0,1,1)_{12}$ model.

The Model Parameters

The parameter vector in a UCM consists of the variances of the disturbance terms of the unobserved components, the damping coefficients and frequencies in the cycles, the damping coefficient in the autoregression, and the regression coefficients in the regression terms. These parameters are estimated by maximizing the likelihood. It is possible to restrict the values of the model parameters to user-specified values.

Model Specification

A UCM is specified by describing the components in the model. For example, consider the model

$$y_t = \mu_t + \gamma_t + \epsilon_t$$

consisting of the irregular, level, slope, and seasonal components. This model is called the basic structural model (BSM) by Harvey (1989). The syntax for a BSM with monthly seasonality of trigonometric type is as follows:

```
model y;
irregular;
level;
slope;
season length=12 type=trig;
```

Similarly the following syntax specifies a BSM with a response variable *y*, a regressor *x*, and dummy-type monthly seasonality:

```
model y = x;
irregular;
level;
slope variance=0 noest;
season length=12 type=dummy;
```

Moreover, the disturbance variance of the slope component is restricted to zero, giving rise to a local linear trend with fixed slope.

A model can contain multiple cycle and seasonal components. In such cases the model syntax contains a separate statement for each of these multiple cycle or seasonal components; for example, the syntax for a model containing irregular and level components along with two cycle components could be as follows:

```
model y = x;
irregular;
level;
cycle;
cycle;
```

The UCMs as State Space Models

The UCMs considered in PROC UCM can be thought of as special cases of more general models, called (linear) Gaussian state space models (GSSM). A GSSM can be described as follows:

$$\begin{aligned} y_t &= Z_t \alpha_t \\ \alpha_{t+1} &= T_t \alpha_t + \zeta_{t+1}, \qquad \zeta_t \sim \mathrm{N}(0, Q_t) \\ \alpha_1 &\sim \mathrm{N}(0, P) \end{aligned}$$

The first equation, called the *observation equation*, relates the response series y_t to a state vector α_t that is usually unobserved. The second equation, called the *state equation*, describes the evolution of the state vector in time. The system matrices Z_t and T_t are of appropriate dimensions and are known, except possibly for some unknown elements that become part of the parameter vector of the model. The noise series ζ_t consists of independent, zero-mean, Gaussian vectors with covariance matrices Q_t. For most of the UCMs considered here, the system matrices Z_t and T_t, and the noise covariances Q_t, are time invariant—that is, they do not depend on time. In a few cases, however, some or all of them can depend on time. The initial state vector α_1 is assumed to be independent of the noise series, and its covariance matrix P can be partially diffuse. A random vector has a partially diffuse covariance matrix if it can be partitioned such that one part of the vector has a properly defined probability distribution, while the covariance matrix of the other part is infinite—that is, you have no prior information about this part of the vector. The covariance of the initial state α_1 is assumed to have the following form:

$$P = P_* + \kappa P_\infty$$

where P_* and P_∞ are nonnegative definite, symmetric matrices and κ is a constant that is assumed to be close to ∞. In the case of UCMs considered here, P_∞ is always a diagonal matrix that consists of zeros and ones, and, if a particular diagonal element of P_∞ is one, then the corresponding row and column in P_* are zero.

The state space formulation of a UCM has many computational advantages. In this formulation there are convenient algorithms for estimating and forecasting the unobserved states $\{\alpha_t\}$ by using the observed series $\{y_t\}$. These algorithms also yield the in-sample and out-of-sample forecasts and the likelihood of $\{y_t\}$. The state space representation of a UCM does not need to be unique. In the representation used here, the unobserved components in the UCM often appear as elements of the state vector. This makes the elements of the state interpretable and, more important, the sample estimates and forecasts of these unobserved components are easily obtained. For additional information about the computational aspects of the state space modeling, see Durbin and Koopman (2001). Next, some notation is developed to describe the essential quantities computed during the analysis of the state space models.

Let $\{y_t, t = 1, \ldots, n\}$ be the observed sample from a series that satisfies a state space model. Next, for $1 \le t \le n$, let the one-step-ahead forecasts of the series, the states, and their variances be defined as follows, using the usual notation to denote the conditional expectation and conditional variance:

$$\begin{aligned} \hat{\alpha}_t &= \mathrm{E}(\alpha_t | y_1, y_2, \ldots, y_{t-1}) \\ \Gamma_t &= \mathrm{Var}(\alpha_t | y_1, y_2, \ldots, y_{t-1}) \\ \hat{y}_t &= \mathrm{E}(y_t | y_1, y_2, \ldots, y_{t-1}) \\ F_t &= \mathrm{Var}(y_t | y_1, y_2, \ldots, y_{t-1}) \end{aligned}$$

These are also called the *filtered* estimates of the series and the states. Similarly, for $t \geq 1$, let the following denote the full-sample estimates of the series and the state values at time *t*:

$$\begin{aligned}
\tilde{\alpha}_t &= \mathrm{E}(\alpha_t | y_1, y_2, \ldots, y_n) \\
\Delta_t &= \mathrm{Var}(\alpha_t | y_1, y_2, \ldots, y_n) \\
\tilde{y}_t &= \mathrm{E}(y_t | y_1, y_2, \ldots, y_n) \\
G_t &= \mathrm{Var}(y_t | y_1, y_2, \ldots, y_n)
\end{aligned}$$

If the time *t* is in the historical period— that is, if $1 \leq t \leq n$— then the full-sample estimates are called the *smoothed* estimates, and if *t* lies in the future then they are called out-of-sample forecasts. Note that if $1 \leq t \leq n$, then $\tilde{y}_t = y_t$ and $G_t = 0$, unless y_t is missing.

All the filtered and smoothed estimates ($\hat{\alpha}_t, \tilde{\alpha}_t, \ldots, G_t$, and so on) are computed by using the Kalman filtering and smoothing (KFS) algorithm, which is an iterative process. If the initial state is diffuse, as is often the case for the UCMs, its treatment requires modification of the traditional KFS, which is called the diffuse KFS (DKFS). The details of DKFS implemented in the UCM procedure can be found in De Jong and Chu-Chun-Lin (2003). Additional information on the state space models can be found in Durbin and Koopman (2001). The likelihood formulas described in this section are taken from the latter reference.

In the case of diffuse initial condition, the effect of the improper prior distribution of α_1 manifests itself in the first few filtering iterations. During these initial filtering iterations the distribution of the filtered quantities remains diffuse; that is, during these iterations the one-step-ahead series and state forecast variances F_t and Γ_t have the following form:

$$\begin{aligned}
F_t &= F_{*t} + \kappa F_{\infty t} \\
\Gamma_t &= \Gamma_{*t} + \kappa \Gamma_{\infty t}
\end{aligned}$$

The actual number of iterations—say, *I*— affected by this improper prior depends on the nature of the vectors Z_t, the number of nonzero diagonal elements of P_∞, and the pattern of missing values in the dependent series. After *I* iterations, $\Gamma_{\infty t}$ and $F_{\infty t}$ become zero and the one-step-ahead series and state forecasts have proper distributions. These first *I* iterations constitute the *initialization* phase of the DKFS algorithm. The post-initialization phase of the DKFS and the traditional KFS is the same. In the state space modeling literature the pre-initialization and post-initialization phases are some times called *pre-collapse* and *post-collapse* phases of the diffuse Kalman filtering. In certain missing value patterns it is possible for *I* to exceed the sample size; that is, the sample information can be insufficient to create a proper prior for the filtering process. In these cases, parameter estimation and forecasting is done on the basis of this improper prior, and some or all of the series and component forecasts can have infinite variances (or zero precision). The forecasts that have infinite variance are set to missing. The same situation can occur if the specified model contains components that are essentially multicollinear. In these situations no residual analysis is possible; in particular, no residuals-based goodness-of-fit statistics are produced.

The log likelihood of the sample (L_∞), which takes account of this diffuse initialization step, is computed by using the one-step-ahead series forecasts as follows

$$L_\infty(y_1, \ldots, y_n) = -\frac{(n-d)}{2} \log 2\pi - \frac{1}{2} \sum_{t=1}^{I} w_t - \frac{1}{2} \sum_{t=I+1}^{n} (\log F_t + \frac{\nu_t^2}{F_t})$$

where d is the number of diffuse elements in the initial state α_1, $\nu_t = y_t - Z_t\hat{\alpha}_t$ are the one-step-ahead residuals, and

$$\begin{aligned} w_t &= \log F_{\infty t} && \text{if } F_{\infty t} > 0 \\ &= \log F_{*t} + \frac{\nu_t^2}{F_{*t}} && \text{if } F_{\infty t} = 0 \end{aligned}$$

If y_t is missing at some time t, then the corresponding summand in the log likelihood expression is deleted, and the constant term is adjusted suitably. Moreover, if the initialization step does not complete—that is, if I exceeds the sample size— then the value of d is reduced to the number of diffuse states that are successfully initialized.

The portion of the log likelihood that corresponds to the post-initialization period is called the nondiffuse log likelihood (L_0). The nondiffuse log likelihood is given by

$$L_0(y_1, \ldots, y_n) = -\frac{1}{2}\sum_{t=I+1}^{n}(\log F_t + \frac{\nu_t^2}{F_t})$$

In the case of UCMs considered in PROC UCM, it often happens that the diffuse part of the likelihood, $\sum_{t=1}^{I} w_t$, does not depend on the model parameters, and in these cases the maximization of nondiffuse and diffuse likelihoods is equivalent. However, in some cases, such as when the model consists of dependent lags, the diffuse part does depend on the model parameters. In these cases the maximization of the diffuse and nondiffuse likelihood can produce different parameter estimates.

In some situations it is convenient to reparameterize the nondiffuse initial state covariance P_* as $\sigma^2 P_*$ and the state noise covariance Q_t as $\sigma^2 Q_t$ for some common scalar parameter σ^2. In this case the preceding log-likelihood expression, up to a constant, can be written as

$$L_\infty(y_1, \ldots, y_n) = -\frac{1}{2}\sum_{t=1}^{I} w_t - \frac{1}{2}\sum_{t=I+1}^{n}\log F_t - \frac{1}{2\sigma^2}\sum_{t=I+1}^{n}\frac{\nu_t^2}{F_t} - \frac{(n-d)}{2}\log\sigma^2$$

Solving analytically for the optimum, the maximum likelihood estimate of σ^2 can be shown to be

$$\hat{\sigma}^2 = \frac{1}{(n-d)}\sum_{t=I+1}^{n}\frac{\nu_t^2}{F_t}$$

When this expression of σ^2 is substituted back into the likelihood formula, an expression called the *profile likelihood* ($L_{profile}$) of the data is obtained:

$$-2L_{profile}(y_1, \ldots, y_n) = \sum_{t=1}^{I} w_t + \sum_{t=I+1}^{n}\log F_t + (n-d)\log(\sum_{t=I+1}^{n}\frac{\nu_t^2}{F_t})$$

In some situations the parameter estimation is done by optimizing the profile likelihood (see the section “Parameter Estimation by Profile Likelihood Optimization” on page 2798 and the PROFILE option in the ESTIMATE statement).

In the remainder of this section the state space formulation of UCMs is further explained by using some particular UCMs as examples. The examples show that the state space formulation of the UCMs depends on the components in the model in a simple fashion; for example, the system matrix T is usually a block diagonal matrix with blocks that correspond to the components in the model. The only exception to this pattern is the UCMs that consist of the lags of dependent variable. This case is considered at the end of the section.

In what follows, $Diag\,[a, b, \ldots]$ denotes a diagonal matrix with diagonal entries $[a, b, \ldots]$, and the transpose of a matrix T is denoted as $T^{'}$.

Locally Linear Trend Model

Recall that the dynamics of the locally linear trend model are

$$\begin{aligned} y_t &= \mu_t + \epsilon_t \\ \mu_t &= \mu_{t-1} + \beta_{t-1} + \eta_t \\ \beta_t &= \beta_{t-1} + \xi_t \end{aligned}$$

Here y_t is the response series and ϵ_t, η_t, and ξ_t are independent, zero-mean Gaussian disturbance sequences with variances $\sigma_\epsilon^2, \sigma_\eta^2$, and σ_ξ^2, respectively. This model can be formulated as a state space model where the state vector $\alpha_t = [\ \epsilon_t\ \mu_t\ \beta_t\ ]^{'}$ and the state noise $\zeta_t = [\ \epsilon_t\ \eta_t\ \xi_t\ ]^{'}$. Note that the elements of the state vector are precisely the unobserved components in the model. The system matrices T and Z and the noise covariance Q corresponding to this choice of state and state noise vectors can be seen to be time invariant and are given by

$$Z = [\ 1\ 1\ 0\ ],\quad T = \begin{bmatrix} 0\ 0\ 0 \\ 0\ 1\ 1 \\ 0\ 0\ 1 \end{bmatrix} \text{ and } Q = Diag\left[\sigma_\epsilon^2, \sigma_\eta^2, \sigma_\xi^2\right]$$

The distribution of the initial state vector α_1 is diffuse, with $P_* = Diag\left[\sigma_\epsilon^2, 0, 0\right]$ and $P_\infty = Diag\,[0, 1, 1]$. The parameter vector θ consists of all the disturbance variances—that is, $\theta = (\sigma_\epsilon^2, \sigma_\eta^2, \sigma_\xi^2)$.

Basic Structural Model

The basic structural model (BSM) is obtained by adding a seasonal component, γ_t, to the local level model. In order to economize on the space, the state space formulation of a BSM with a relatively short season length, season length = 4 (quarterly seasonality), is considered here. The pattern for longer season lengths such as 12 (monthly) and 52 (weekly) is easy to see.

Let us first consider the dummy form of seasonality. In this case the state and state noise vectors are $\alpha_t = \left[\ \epsilon_t\ \mu_t\ \beta_t\ \gamma_{1,t}\ \gamma_{2,t}\ \gamma_{3,t}\ \right]^{'}$ and $\zeta_t = [\ \epsilon_t\ \eta_t\ \xi_t\ \omega_t\ 0\ 0\ ]^{'}$, respectively. The first three elements of the state vector are the irregular, level, and slope components, respectively. The remaining elements, $\gamma_{i,t}$, are lagged versions of the seasonal component γ_t. $\gamma_{1,t}$ corresponds to lag zero—that is, the same as γ_t, $\gamma_{2,t}$ to lag 1 and $\gamma_{3,t}$ to lag 2. The system matrices are

$$Z = [\ 1\ 1\ 0\ 1\ 0\ 0\ ],\quad T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

and $Q = Diag\left[\sigma_\epsilon^2, \sigma_\eta^2, \sigma_\xi^2, \sigma_\omega^2, 0, 0\right]$. The distribution of the initial state vector α_1 is diffuse, with $P_* = Diag\left[\sigma_\epsilon^2, 0, 0, 0, 0, 0\right]$ and $P_\infty = Diag\,[0, 1, 1, 1, 1, 1]$.

In the case of the trigonometric type of seasonality, $\alpha_t = \left[\ \epsilon_t\ \mu_t\ \beta_t\ \gamma_{1,t}\ \gamma_{1,t}^*\ \gamma_{2,t}\ \right]'$ and $\zeta_t = \left[\ \epsilon_t\ \eta_t\ \xi_t\ \omega_{1,t}\ \omega_{1,t}^*\ \omega_{2,t}\ \right]'$. The disturbance sequences, $\omega_{j,t}, 1 \le j \le 2$, and $\omega_{1,t}^*$, are independent, zero-mean, Gaussian sequences with variance σ_ω^2. The system matrices are

$$Z = [\ 1\ 1\ 0\ 1\ 0\ 1\ ], \quad T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\lambda_1 & \sin\lambda_1 & 0 \\ 0 & 0 & 0 & -\sin\lambda_1 & \cos\lambda_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cos\lambda_2 \end{bmatrix}$$

and $Q = Diag\left[\sigma_\epsilon^2, \sigma_\eta^2, \sigma_\xi^2, \sigma_\omega^2, \sigma_\omega^2, \sigma_\omega^2\right]$. Here $\lambda_j = (2\pi j)/4$. The distribution of the initial state vector α_1 is diffuse, with $P_* = Diag\left[\sigma_\epsilon^2, 0, 0, 0, 0, 0\right]$ and $P_\infty = Diag\,[0, 1, 1, 1, 1, 1]$. The parameter vector in both the cases is $\theta = (\sigma_\epsilon^2, \sigma_\eta^2, \sigma_\xi^2, \sigma_\omega^2)$.

Seasons with Blocked Seasonal Values

Block seasonals are special seasonal components that impose a special block structure on the seasonal effects. Let us consider a BSM with monthly seasonality that has a quarterly block structure—that is, months within the same quarter are assumed to have identical effects except for some random perturbation. Such a seasonal component is a block seasonal with block size m equal to 3 and the number of blocks k equal to 4. The state space structure for such a model with dummy-type seasonality is as follows: The state and state noise vectors are $\alpha_t = \left[\ \epsilon_t\ \mu_t\ \beta_t\ \gamma_{1,t}\ \gamma_{2,t}\ \gamma_{3,t}\ \right]'$ and $\zeta_t = [\ \epsilon_t\ \eta_t\ \xi_t\ \omega_t\ 0\ 0\ ]'$, respectively. The first three elements of the state vector are the irregular, level, and slope components, respectively. The remaining elements, $\gamma_{i,t}$, are lagged versions of the seasonal component γ_t. $\gamma_{1,t}$ corresponds to lag zero—that is, the same as γ_t, $\gamma_{2,t}$ to lag m and $\gamma_{3,t}$ to lag $2m$. All the system matrices are time invariant, except the matrix T. They can be seen to be $Z = [\ 1\ 1\ 0\ 1\ 0\ 0\ ]$, $Q = Diag\left[\sigma_\epsilon^2, \sigma_\eta^2, \sigma_\xi^2, \sigma_\omega^2, 0, 0\right]$, and

$$T_t = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

when t is a multiple of the block size m, and

$$T_t = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

otherwise. Note that when t is not a multiple of m, the portion of the T_t matrix corresponding to the seasonal is identity. The distribution of the initial state vector α_1 is diffuse, with $P_* = Diag\left[\sigma_\epsilon^2, 0, 0, 0, 0, 0\right]$ and $P_\infty = Diag\,[0, 1, 1, 1, 1, 1]$.

Similarly in the case of the trigonometric form of seasonality, $\alpha_t = \left[\begin{array}{cccccc} \epsilon_t & \mu_t & \beta_t & \gamma_{1,t} & \gamma_{1,t}^* & \gamma_{2,t} \end{array}\right]'$ and $\zeta_t = \left[\begin{array}{cccccc} \epsilon_t & \eta_t & \xi_t & \omega_{1,t} & \omega_{1,t}^* & \omega_{2,t} \end{array}\right]'$. The disturbance sequences, $\omega_{j,t}, 1 \leq j \leq 2$, and $\omega_{1,t}^*$, are independent, zero-mean, Gaussian sequences with variance σ_ω^2. $Z = [\,1\ 1\ 0\ 1\ 0\ 1\,]$, $Q = Diag\left[\sigma_\epsilon^2, \sigma_\eta^2, \sigma_\xi^2, \sigma_\omega^2, \sigma_\omega^2, \sigma_\omega^2\right]$, and

$$T_t = \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\lambda_1 & \sin\lambda_1 & 0 \\ 0 & 0 & 0 & -\sin\lambda_1 & \cos\lambda_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cos\lambda_2 \end{array}\right]$$

when t is a multiple of the block size m, and

$$T_t = \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

otherwise. As before, when t is not a multiple of m, the portion of the T_t matrix corresponding to the seasonal is identity. Here $\lambda_j = (2\pi j)/4$. The distribution of the initial state vector α_1 is diffuse, with $P_* = Diag\left[\sigma_\epsilon^2, 0, 0, 0, 0, 0\right]$ and $P_\infty = Diag\,[0, 1, 1, 1, 1, 1]$. The parameter vector in both the cases is $\theta = (\sigma_\epsilon^2, \sigma_\eta^2, \sigma_\xi^2, \sigma_\omega^2)$.

Cycles and Autoregression

The preceding examples have illustrated how to build a state space model corresponding to a UCM that includes components such as irregular, trend, and seasonal. There you can see that the state vector and the system matrices have a simple block structure with blocks corresponding to the components in the model. Therefore, here only a simple model consisting of a single cycle and an irregular component is considered. The state space form for more complex UCMs consisting of multiple cycles and other components can be easily deduced from this example.

Recall that a stochastic cycle ψ_t with frequency λ, $0 < \lambda < \pi$, and damping coefficient ρ can be modeled as

$$\left[\begin{array}{c} \psi_t \\ \psi_t^* \end{array}\right] = \rho \left[\begin{array}{cc} \cos\lambda & \sin\lambda \\ -\sin\lambda & \cos\lambda \end{array}\right] \left[\begin{array}{c} \psi_{t-1} \\ \psi_{t-1}^* \end{array}\right] + \left[\begin{array}{c} \nu_t \\ \nu_t^* \end{array}\right]$$

where ν_t and ν_t^* are independent, zero-mean, Gaussian disturbances with variance σ_ν^2. In what follows, a state space form for a model consisting of such a stochastic cycle and an irregular component is given.

The state vector $\alpha_t = \left[\begin{array}{ccc} \epsilon_t & \psi_t & \psi_t^* \end{array}\right]'$, and the state noise vector $\zeta_t = \left[\begin{array}{ccc} \epsilon_t & \nu_t & \nu_t^* \end{array}\right]'$. The system matrices are

$$Z = [\,1\ 1\ 0\,] \quad T = \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & \rho\cos\lambda & \rho\sin\lambda \\ 0 & -\rho\sin\lambda & \rho\cos\lambda \end{array}\right] \quad Q = Diag\left[\sigma_\epsilon^2, \sigma_\nu^2, \sigma_\nu^2\right]$$

The distribution of the initial state vector α_1 is proper, with $P_* = Diag\left[\sigma_\epsilon^2, \sigma_\psi^2, \sigma_\psi^2\right]$, where $\sigma_\psi^2 = \sigma_\nu^2(1-\rho^2)^{-1}$. The parameter vector $\theta = (\sigma_\epsilon^2, \rho, \lambda, \sigma_\nu^2)$.

An autoregression r_t can be considered as a special case of cycle with frequency λ equal to 0 or π. In this case the equation for ψ_t^* is not needed. Therefore, for a UCM consisting of an autoregressive component and an irregular component, the state space model simplifies to the following form.

The state vector $\alpha_t = [\ \epsilon_t\ r_t\ ]'$, and the state noise vector $\zeta_t = [\ \epsilon_t\ \nu_t\ ]'$. The system matrices are

$$Z = [\ 1\ 1\ ],\quad T = \begin{bmatrix} 0 & 0 \\ 0 & \rho \end{bmatrix} \text{ and } Q = Diag\left[\sigma_\epsilon^2, \sigma_\nu^2\right]$$

The distribution of the initial state vector α_1 is proper, with $P_* = Diag\left[\sigma_\epsilon^2, \sigma_r^2\right]$, where $\sigma_r^2 = \sigma_\nu^2(1-\rho^2)^{-1}$. The parameter vector $\theta = (\sigma_\epsilon^2, \rho, \sigma_\nu^2)$.

Incorporating Predictors of Different Kinds

In the UCM procedure, predictors can be incorporated in a UCM in a variety of ways: simple time-invariant linear predictors are specified in the MODEL statement, predictors with time-varying coefficients can be specified in the RANDOMREG statement, and predictors that have a nonlinear relationship with the response variable can be specified in the SPLINEREG statement. As with earlier examples, how to obtain a state space form of a UCM consisting of such variety of predictors is illustrated using a simple special case. Consider a random walk trend model with predictors x, u_1, u_2, and v. Let us assume that x is a simple regressor specified in the MODEL statement, u_1 and u_2 are random regressors with time-varying regression coefficients that are specified in the same RANDOMREG statement, and v is a nonlinear regressor specified on a SPLINEREG statement. Let us further assume that the spline associated with v has degree one and is based on two internal knots. As explained in the section "SPLINEREG Statement" on page 2779, using v is equivalent to using $(nknots + degree) = (2+1) = 3$ derived (random) regressors: say, s_1, s_2, s_3. In all there are $(1+2+3) = 6$ regressors, the first one being a simple regressor and the others being time-varying coefficient regressors. The time-varying regressors are in two groups, the first consisting of u_1 and u_2 and the other consisting of s_1, s_2, and s_3. The dynamics of this model are as follows:

$$\begin{aligned}
y_t &= \mu_t + \beta x_t + \kappa_{1t} u_{1t} + \kappa_{2t} u_{2t} + \sum_{i=1}^{3} \gamma_{it} s_{it} + \epsilon_t \\
\mu_t &= \mu_{t-1} + \eta_t \\
\kappa_{1t} &= \kappa_{1(t-1)} + \xi_{1t} \\
\kappa_{2t} &= \kappa_{2(t-1)} + \xi_{2t} \\
\gamma_{1t} &= \gamma_{1(t-1)} + \zeta_{1t} \\
\gamma_{2t} &= \gamma_{2(t-1)} + \zeta_{2t} \\
\gamma_{3t} &= \gamma_{3(t-1)} + \zeta_{3t}
\end{aligned}$$

All the disturbances $\epsilon_t, \eta_t, \xi_{1t}, \xi_{2t}, \zeta_{1t}, \zeta_{2t}$, and ζ_{3t} are independent, zero-mean, Gaussian variables, where ξ_{1t}, ξ_{2t} share a common variance parameter σ_ξ^2 and $\zeta_{1t}, \zeta_{2t}, \zeta_{3t}$ share a common variance σ_ζ^2. These dynamics can be captured in the state space form by taking state $\alpha_t = [\ \epsilon_t\ \mu_t\ \beta\ \kappa_{1t}\ \kappa_{2t}\ \gamma_{1t}\ \gamma_{2t}\ \gamma_{3t}\ ]'$, state disturbance $\zeta_t = [\, \epsilon_t \; \eta_t \; 0 \; \xi_{1t} \; \xi_{2t} \; \zeta_{1t} \; \zeta_{2t} \; \zeta_{3t} \,]^{'}$, and the system matrices

$$\begin{aligned}
Z_t &= [\, 1 \; 1 \; x_t \; u_{1t} \; u_{2t} \; s_{1t} \; s_{2t} \; s_{3t} \,] \\
T &= Diag\,[0,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1] \\
Q &= Diag\left[\sigma^2_\epsilon,\ \sigma^2_\eta,\ 0,\ \sigma^2_\xi,\ \sigma^2_\xi,\ \sigma^2_\zeta,\ \sigma^2_\zeta,\ \sigma^2_\zeta\right]
\end{aligned}$$

Note that the regression coefficients are elements of the state vector and that the system vector Z_t is not time invariant. The distribution of the initial state vector α_1 is diffuse, with $P_* = Diag\left[\sigma^2_\epsilon, 0, 0, 0, 0, 0, 0, 0\right]$ and $P_\infty = Diag\,[0, 1, 1, 1, 1, 1, 1, 1]$. The parameters of this model are the disturbance variances, σ^2_ϵ, σ^2_η, σ^2_ξ, and σ^2_ζ, which get estimated by maximizing the likelihood. The regression coefficients, time-invariant β and time-varying $\kappa_{1t}, \kappa_{2t}, \gamma_{1t}, \gamma_{2t}$ and γ_{3t}, get implicitly estimated during the state estimation (smoothing).

Reporting Parameter Estimates for Random Regressors

If the random walk disturbance variance associated with a random regressor is held fixed at zero, then its coefficient is no longer time-varying. In the UCM procedure the random regressor parameter estimates are reported differently if the random walk disturbance variance associated with a random regressor is held fixed at zero. The following points explain how the parameter estimates are reported in the parameter estimates table and in the OUTEST= data set.

- If the random walk disturbance variance associated with a random regressor is not held fixed, then its estimate is reported in the parameter estimates table and in the OUTEST= data set.

- If more that one random regressor is specified in a RANDOMREG statement, then the first regressor in the list is used as a representative of the list while reporting the corresponding common variance parameter estimate.

- If the random walk disturbance variance is held fixed at zero, then the parameter estimates table and the OUTEST= data set contain the corresponding regression parameter estimate rather than the variance parameter estimate.

- Similar considerations apply in the case of the derived random regressors associated with a spline-regressor.

Forecasting with Predictor Variables

If regression effects are included in the model (via a MODEL statement or RANDOMREG and SPLINEREG statements) and the FORECAST statement is used to compute multistep forecasts, then future values of the predictor variables must be included in the DATA= data set for the forecast horizon that is defined by the BACK= and LEAD= options. For more information about how the forecast horizon is defined, see the FORECAST statement.

ARMA Irregular Component

The state space form for the irregular component that follows an ARMA(p,q)×(P,Q)$_s$ model is described in this section. The notation for ARMA models is explained in the IRREGULAR statement. A number of alternate state space forms are possible in this case; the one given here is based on Jones (1980). With slight abuse of notation, let $p = p + sP$ denote the effective autoregressive order and $q = q + sQ$ denote the effective moving average order of the model. Similarly, let ϕ be the effective autoregressive polynomial and θ be the effective moving average polynomial in the backshift operator with coefficients $\phi_1, \ldots, \phi_p$ and $\theta_1, \ldots, \theta_q$, obtained by multiplying the respective nonseasonal and seasonal factors. Then, a random sequence ϵ_t that follows an ARMA(p,q)×(P,Q)$_s$ model with a white noise sequence a_t has a state space form with state vector of size $m = \max(p, q + 1)$. The system matrices, which are time invariant, are as follows: $Z = [1\ 0\ \ldots\ 0]$. The state transition matrix T, in a blocked form, is given by

$$T = \begin{bmatrix} 0 & I_{m-1} \\ \phi_m \ \ldots & \phi_1 \end{bmatrix}$$

where $\phi_i = 0$ if $i > p$ and I_{m-1} is an $(m-1)$ dimensional identity matrix. The covariance of the state disturbance matrix $Q = \sigma^2 \psi \psi'$ where σ^2 is the variance of the white noise sequence a_t and the vector $\psi = [\psi_0 \ldots \psi_{m-1}]'$ contains the first m values of the impulse response function—that is, the first m coefficients in the expansion of the ratio θ/ϕ. Since ϵ_t is a stationary sequence, the initial state is nondiffuse and $P_\infty = 0$. The description of P_*, the covariance matrix of the initial state, is a little involved; the details are given in Jones (1980).

Models with Dependent Lags

The state space form of a UCM consisting of the lags of the dependent variable is quite different from the state space forms considered so far. Let us consider an example to illustrate this situation. Consider a model that has random walk trend, two simple time-invariant regressors, and that also includes a few—say, k—lags of the dependent variable. That is,

$$\begin{aligned} y_t &= \sum_{i=1}^{k} \phi_i y_{t-i} + \mu_t + \beta_1 x_{1t} + \beta_2 x_{2t} + \epsilon_t \\ \mu_t &= \mu_{t-1} + \eta_t \end{aligned}$$

The state space form of this augmented model can be described in terms of the state space form of a model that has random walk trend with two simple time-invariant regressors. A superscript dagger (†) has been added to distinguish the augmented model state space entities from the corresponding entities of the state space form of the random walk with predictors model. With this notation, the state vector of the augmented model $\alpha_t^\dagger = \begin{bmatrix} \alpha_t' \ y_t \ y_{t-1} \ \ldots \ y_{t-k+1} \end{bmatrix}'$ and the new state noise vector $\zeta_t^\dagger = \begin{bmatrix} \zeta_t' \ u_t \ 0 \ldots \ 0 \end{bmatrix}'$, where u_t is the matrix product $Z_t \zeta_t$. Note that the length of the new state vector is $k + \text{length}(\alpha_t) = k + 4$. The new system matrices, in block form, are

$$Z_t^\dagger = [\,0\,0\,0\,0\,1\ \ldots\ 0\,], \quad T_t^\dagger = \begin{bmatrix} T_t & 0 & \ldots & 0 \\ Z_{t+1}T_t & \phi_1 & \ldots & \phi_k \\ 0 & I_{k-1,k-1} & & 0 \end{bmatrix}$$

where $I_{k-1,k-1}$ is the $k-1$ dimensional identity matrix and

$$Q_t^\dagger = \begin{bmatrix} Q_t & Q_t Z_t' & 0 \\ Z_t Q_t & Z_t Q_t Z_t' & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Note that the T and Q matrices of the random walk with predictors model are time invariant, and in the expressions above their time indices are kept because they illustrate the pattern for more general models. The initial state vector is diffuse, with

$$P_*^\dagger = \begin{bmatrix} P_* & 0 \\ 0 & 0 \end{bmatrix}, \quad P_\infty^\dagger = \begin{bmatrix} P_\infty & 0 \\ 0 & I_{k,k} \end{bmatrix}$$

The parameters of this model are the disturbance variances σ_{ϵ}^{2} and σ_{η}^{2}, the lag coefficients $\phi_1, \phi_2, \ldots, \phi_k$, and the regression coefficients β_1 and β_2. As before, the regression coefficients get estimated during the state smoothing, and the other parameters are estimated by maximizing the likelihood.

Outlier Detection

In time series analysis it is often useful to detect changes over time in the characteristics of the response series. In the UCM procedure you can search for two types of changes, additive outliers (AO) and level shifts (LS). An additive outlier is an unusual value in the series, the cause of which might be a data recording error or a temporary shock to the series generation process. A level shift represents a permanent shift, either up or down, in the level of the series. You can control different aspects of the outlier search, such as the significance level of the reported outliers, by choosing different options in the OUTLIER statement. The search for AOs is done by default, whereas the CHECKBREAK option in the LEVEL statement must be used to turn on the search for LSs.

The outlier detection process implemented in the UCM procedure is based on De Jong and Penzer (1998). In this approach the fitted model is taken to be the *null* model, and the series values and level shifts that are not adequately accounted for by the null model are flagged as outliers. The unusualness of a response series value at a particular time point t_0, with respect to the fitted model, can be judged by estimating its value based on the rest of the data (that is, the series obtained by *deleting* the series value at t_0) and comparing the estimated value to the observed value. If the difference between the estimated and observed values is statistically significant, then such value can be regarded as an AO. Note that this difference between the estimated and observed values is also the regression coefficient of a *dummy* regressor that takes the value 1.0 at t_0 and is 0.0 elsewhere, assuming such a regressor is added to the null model. In this way the series value at t_0 is regarded as AO if the regression coefficient of this dummy regressor is significant. Similarly, you can say that a level shift has occurred at a time point t_0 if the regression coefficient of a regressor, which is 0.0 before t_0 and 1.0 at t_0 and thereafter, is statistically significant. De Jong and Penzer (1998) provide an efficient way to compute such AO and LS regression coefficients and their standard errors at all time points in the series. The outlier summary table, which is produced by default, simply lists the most statistically significant candidates among these.

Missing Values

Embedded missing values in the dependent variable usually cause no problems in UCM modeling. However, no missing values are allowed in the predictor variables. Certain patterns of missing values in the dependent variable can lead to failure of the initialization step of the diffuse Kalman filtering for some models. For example, if in a monthly series all values are missing for a certain month—say, May—then a BSM with monthly seasonality leads to such a situation. However, in this case the initialization step can complete successfully for a nonseasonal model such as local linear model.

Parameter Estimation

The parameter vector in a UCM consists of the variances of the disturbance terms of the unobserved components, the damping coefficients and frequencies in the cycles, the damping coefficient in the autoregression, the lag coefficients of the dependent lags, and the regression coefficients in the regression terms. The regression coefficients are always part of the state vector and are estimated by state smoothing. The remaining parameters are estimated by maximizing either the full diffuse likelihood or the nondiffuse likelihood. The decision to use the full diffuse likelihood or the nondiffuse likelihood depends on the presence or absence of the dependent lag coefficients in the parameter vector. If the parameter vector does not contain any dependent lag coefficients, then the full diffuse likelihood is used. If, on the other hand, the parameter vector does contain some dependent lag coefficients, then the parameters are estimated by maximizing the nondiffuse likelihood. The optimization of the full diffuse likelihood is often unstable when the parameter vector contains dependent lag coefficients. In this sense, when the parameter vector contains dependent lag coefficients, the parameter estimates are not true maximum likelihood estimates.

The optimization of the likelihood, either full or nondiffuse, is carried out using one of several nonlinear optimization algorithms. The user can control many aspects of the optimization process by using the NLOPTIONS statement and by providing the starting values of the parameters while specifying the corresponding components. However, in most cases the default settings work quite well. The optimization process is not guaranteed to converge to a maximum likelihood estimate. In most cases the difficulties in parameter estimation are associated with the specification of a model that is not appropriate for the series being modeled.

Parameter Estimation by Profile Likelihood Optimization

If a disturbance variance, such as the disturbance variance of the irregular component, is a part of the UCM and is a free parameter, then it can be profiled out of the likelihood. This means solving analytically for its optimum and plugging this expression back into the likelihood formula, giving rise to the so-called *profile* likelihood. The expression of the profile likelihood and the MLE of the profiled variance are given earlier in the section "The UCMs as State Space Models" on page 2788, where the computation of the likelihood of the state space model is also discussed.

In some situations the optimization of the profile likelihood can be more efficient because the number of parameters to optimize is reduced by one; however, for a variety of reasons such gains might not always be observed. Moreover, in theory the estimates obtained by optimizing the profile likelihood and the usual likelihood should be the same, but in practice this might not hold because of numerical rounding and other conditions.

In the UCM procedure, by default the usual likelihood is optimized if any of the disturbance variance parameters is held fixed to a nonzero value by using the NOEST option in the corresponding component statement. In other cases the decision whether to optimize the profile likelihood or the usual likelihood is based on several factors that are difficult to document. You can choose which likelihood to optimize during parameter estimation by specifying the PROFILE option for the profile likelihood optimization or the NOPROFILE option for the usual likelihood optimization. In the presence of the PROFILE option, the disturbance variance to profile is checked in a specific order, so that if the irregular component disturbance variance is free then it is always chosen. The situation in other cases is more complicated.

Profiling in the Presence of Fixed Variance Parameters

Note that when the parameter estimation is done by optimizing the profile likelihood, the interpretation of the variance parameters that are held fixed to nonzero values changes. In the presence of the PROFILE option, the disturbance variances that are held at a fixed value by using the NOEST option in their respective component statements are interpreted as being restricted to be that fixed multiple of the profiled variance rather than being fixed at that nominal value. That is, implicitly, the parameter estimation is done under the restriction of holding the disturbance variance *ratio* fixed at a given value rather than the disturbance variance itself. See Example 41.5 for an example of this type of restriction to obtain a UC model that is equivalent to the famous Hodrick-Prescott filter.

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. The distributional properties of the maximum likelihood estimates of general unobserved components models have not been explored fully; therefore the probability values that correspond to a *t* distribution should be interpreted carefully, as they can be misleading. This is particularly true if the parameters in question are close to the boundary of the parameter space. The two sources by Harvey (1989, 2001) are good references for information about this topic. For some parameters, such as, the cycle period, the reported *t* values are uninformative because comparison of the estimated parameter with zero is never needed. In such cases the *t* values and the corresponding probability values should be ignored.

Bootstrap Prediction Intervals (Experimental)

By default, the UCM procedure computes the standard errors of the series and component forecasts (both the filtered and smoothed estimates) by assuming that the estimated parameters are in fact the true parameters. Rodriguez and Ruiz (2010) describe a bootstrap-based procedure to compute the standard errors of the series and component forecasts that takes into account the uncertainty of parameter estimation. As an experimental feature in this release, you can request the computation of standard errors based on this bootstrap-based procedure by specifying the BOOTSTRAP option in the FORECAST statement. Subsequently, the confidence intervals for the series and component forecasts are based on these bootstrap standard errors. The algorithm that PROC UCM uses closely follows the first procedure described in Section 3 of Rodriguez and Ruiz (2010). Note that this bootstrap algorithm is computationally expensive. The computational burden increases with the number of bootstrap replications and is comparable to the computational burden of fitting the specified model as many times as the number of replications. Fortunately, these replications can be executed in parallel, and the UCM procedure can use multiple cores and multiple grid nodes (if they are available) to complete these calculations faster. For a single machine with multiple cores, the procedure automatically detects and uses all the cores. If a grid environment with multiple machines is available (with the appropriate SAS license), you must use the PERFORMANCE statement to supply the necessary information to the UCM procedure.

Computational Issues

Convergence Problems

As explained in the section “Parameter Estimation” on page 2797, the model parameters are estimated by nonlinear optimization of the likelihood. This process is not guaranteed to succeed. For some data sets, the optimization algorithm can fail to converge. Nonconvergence can result from a number of causes, including flat or ridged likelihood surfaces and ill-conditioned data. It is also possible for the algorithm to converge to a point that is not the global optimum of the likelihood.

If you experience convergence problems, the following points might be helpful:

- Data that are extremely large or extremely small can adversely affect results because of the internal tolerances used during the filtering steps of the likelihood calculation. Rescaling the data can improve stability.

- Examine your model for redundancies in the included components and regressors. If some of the included components or regressors are nearly collinear to each other, then the optimization process can become unstable.

- Experimenting with different options offered by the NLOPTIONS statement can help.

- Lack of convergence can indicate model misspecification or a violation of the normality assumption.

Computer Resource Requirements

The computing resources required for the UCM procedure depend on several factors. The memory requirement for the procedure is largely dependent on the number of observations to be processed and the size of the state vector underlying the specified model. If n denotes the sample size and m denotes the size of the state vector, the memory requirement for the smoothing stage of the Kalman filter is of the order of $6 \times 8 \times n \times m^2$ bytes, ignoring the lower-order terms. If the smoothed component estimates are not needed then the memory requirement is of the order of $6 \times 8 \times (m^2 + n)$ bytes. Besides m and n, the computing time for the parameter estimation depends on the type of components included in the model. For example, the parameter estimation is usually faster if the model parameter vector consists only of disturbance variances, because in this case there is an efficient way to compute the likelihood gradient.

Displayed Output

The default printed output produced by the UCM procedure is described in the following list:

- brief information about the input data set, including the data set name and label, and the name of the ID variable specified in the ID statement

- summary statistics for the data in the estimation and forecast spans, including the names of the variables in the model, their categorization as dependent or predictor, the index of the beginning and ending observations in the spans, the total number of observations and the number of missing observations, the smallest and largest measurements, and the mean and standard deviation

- information about the model parameters at the start of the model-fitting stage, including the fixed parameters in the model and the initial estimates of the free parameters in the model

- convergence status of the likelihood optimization process if any parameter estimation is done

- estimates of the free parameters at the end of the model fitting-stage, including the parameter estimates, their approximate standard errors, t statistics, and the approximate p-value

- the likelihood-based goodness-of-fit statistics, including the full likelihood, the portion of the likelihood corresponding to the diffuse initialization, the sum of squares of residuals normalized by their standard errors, and the information criteria: AIC, AICC, HQIC, BIC, and CAIC

- the fit statistics that are based on the raw residuals (observed minus predicted), including the mean squared error (MSE), the root mean squared error (RMSE), the mean absolute percentage error (MAPE), the maximum percentage error (MAXPE), the R square, the adjusted R square, the random walk R square, and Amemiya's R square

- the significance analysis of the components included in the model that is based on the estimation span

- brief information about the components included in the model

- additive outliers in the series, if any are detected

- the multistep series forecasts

- post-sample-prediction analysis table that compares the multistep forecasts with the observed series values, if the BACK= option is used in the FORECAST statement

Statistical Graphics

Statistical procedures use ODS Graphics to create graphs as part of their output. ODS Graphics is described in detail in Chapter 21, "Statistical Graphics Using ODS" (*SAS/STAT User's Guide*).

Before you create graphs, ODS Graphics must be enabled (for example, with the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" in that chapter.

The overall appearance of graphs is controlled by ODS styles. Styles and other aspects of using ODS Graphics are discussed in the section "A Primer on ODS Statistical Graphics" in that chapter.

This section provides information about the basic ODS statistical graphics produced by the UCM procedure.

You can obtain most plots relevant to the specified model by using the global PLOTS= option in the PROC UCM statement. The plot of series forecasts in the forecast horizon is produced by default. You can further control the production of individual plots by using the PLOT= options in the different statements.

The main types of plots available are as follows:

- Time series plots of the component estimates, either filtered or smoothed, can be requested by using the PLOT= option in the respective component statements. For example, the use of PLOT=SMOOTH option in a CYCLE statement produces a plot of smoothed estimate of that cycle.

- Residual plots for model diagnostics can be obtained by using the PLOT= option in the ESTIMATE statement.

- Plots of series forecasts and model decompositions can be obtained by using the PLOT= option in the FORECAST statement.

The following example is a simple illustration of the available plot options.

Analysis of Sunspot Data: Illustration of ODS Graphics

In this example a well-known series, Wolfer's sunspot data (Anderson 1971), is considered. The data consist of yearly sunspot numbers recorded from 1749 to 1924. These sunspot numbers are known to have a cyclical pattern with a period of about eleven years. The following DATA step creates the input data set:

```
data sunspot;
   input year wolfer @@;
   year = mdy(1,1, year);
   format year year4.;
datalines;
1749  809 1750  834 1751  477 1752  478 1753  307 1754  122 1755   96
1756  102 1757  324 1758  476 1759  540 1760  629 1761  859 1762  612
1763  451 1764  364 1765  209 1766  114 1767  378 1768  698 1769 1061

   ... more lines ...
```

The following statements specify a UCM that includes a cycle component and a random walk trend component:

```
proc ucm data=sunspot;
   id year interval=year;
   model wolfer;
   irregular;
   level ;
   cycle plot=(filter smooth);
   estimate back=24 plot=(loess panel cusum wn);
   forecast back=24 lead=24 plot=(forecasts decomp);
run;
```

The following subsections explain the graphics produced by the above statements.

Component Plots

The plots in Figure 41.8 and Figure 41.9, produced by specifying PLOT=(FILTER SMOOTH) in the CYCLE statement, show the filtered and smoothed estimates, respectively, of the cycle component in the model. Note that the smoothed estimate appears smoother than the filtered estimate. This is always true because the filtered estimate of a component at time t is based on the observations prior to time t—that is, it uses measurements from the first observation up to the $(t - 1)$th observation. On the other hand, the corresponding smoothed estimate uses all the available observations—that is, all the measurements from the first observation to the last. This makes the smoothed estimate of the component more precise than the filtered estimate for the time points within historical period. In the forecast horizon, both filtered and smoothed estimates are identical, being based on the same set of observations.

Figure 41.8 Sunspots Series: Filtered Cycle

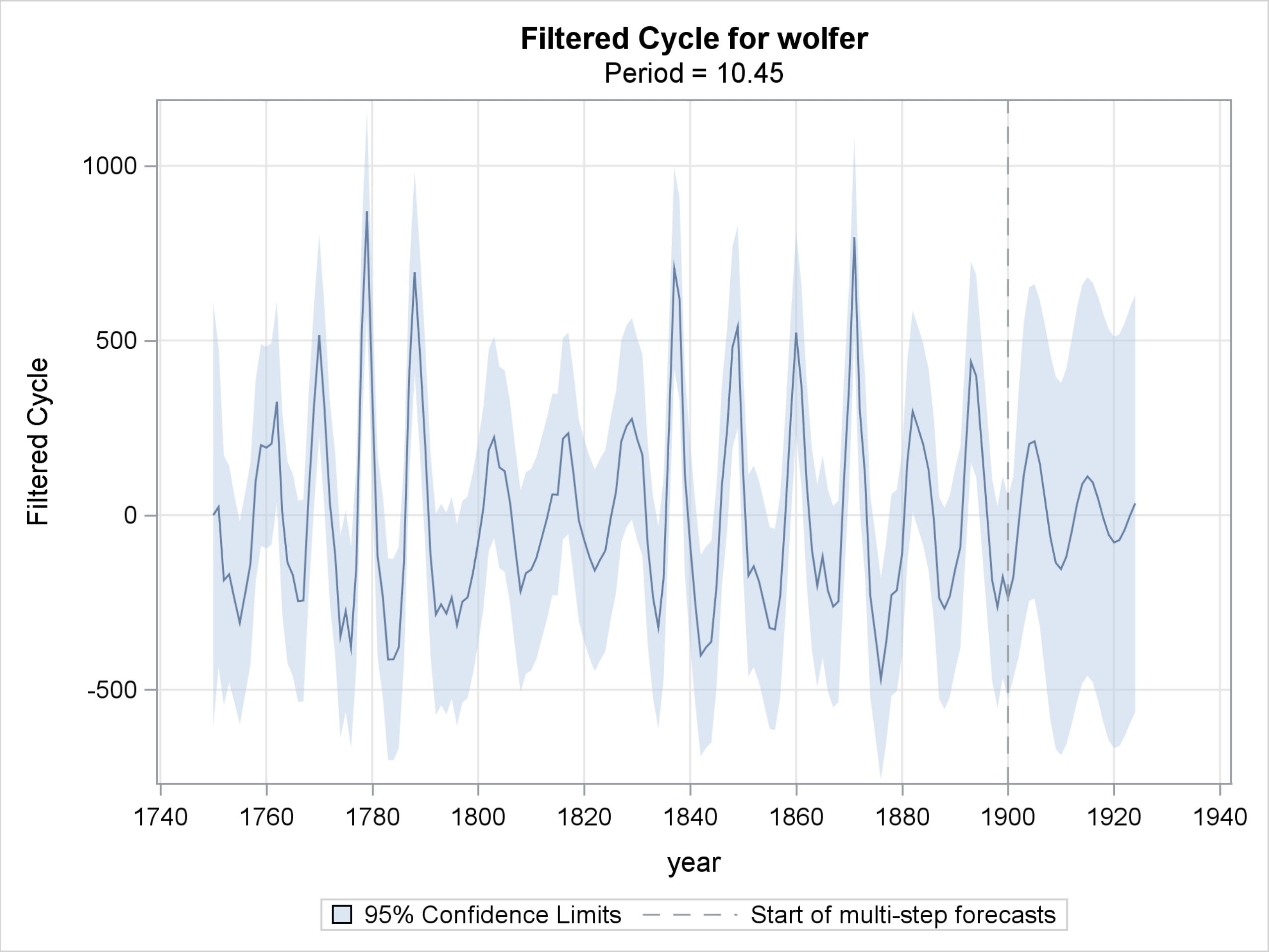
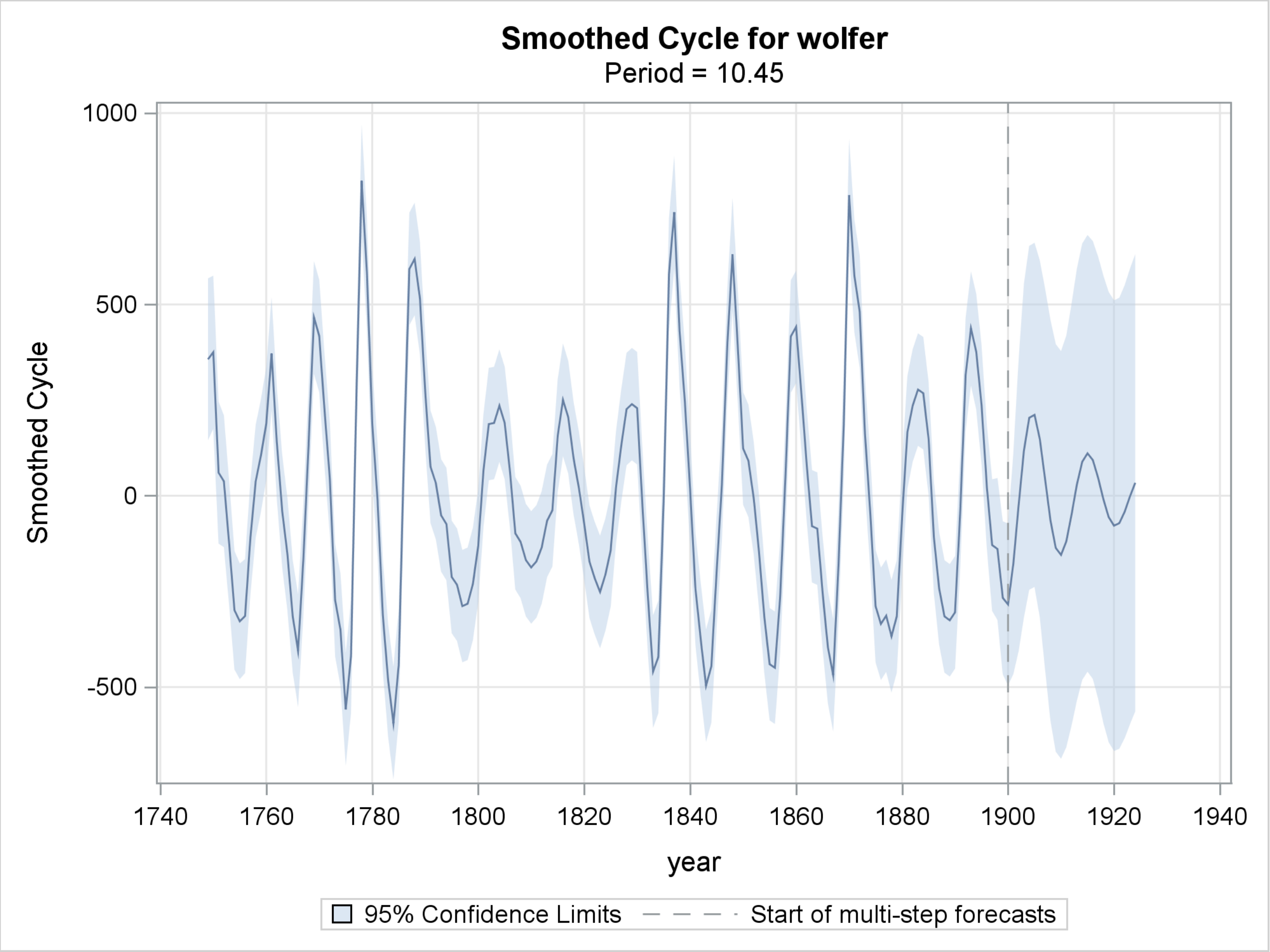


Figure 41.9 Sunspots Series: Smoothed Cycle

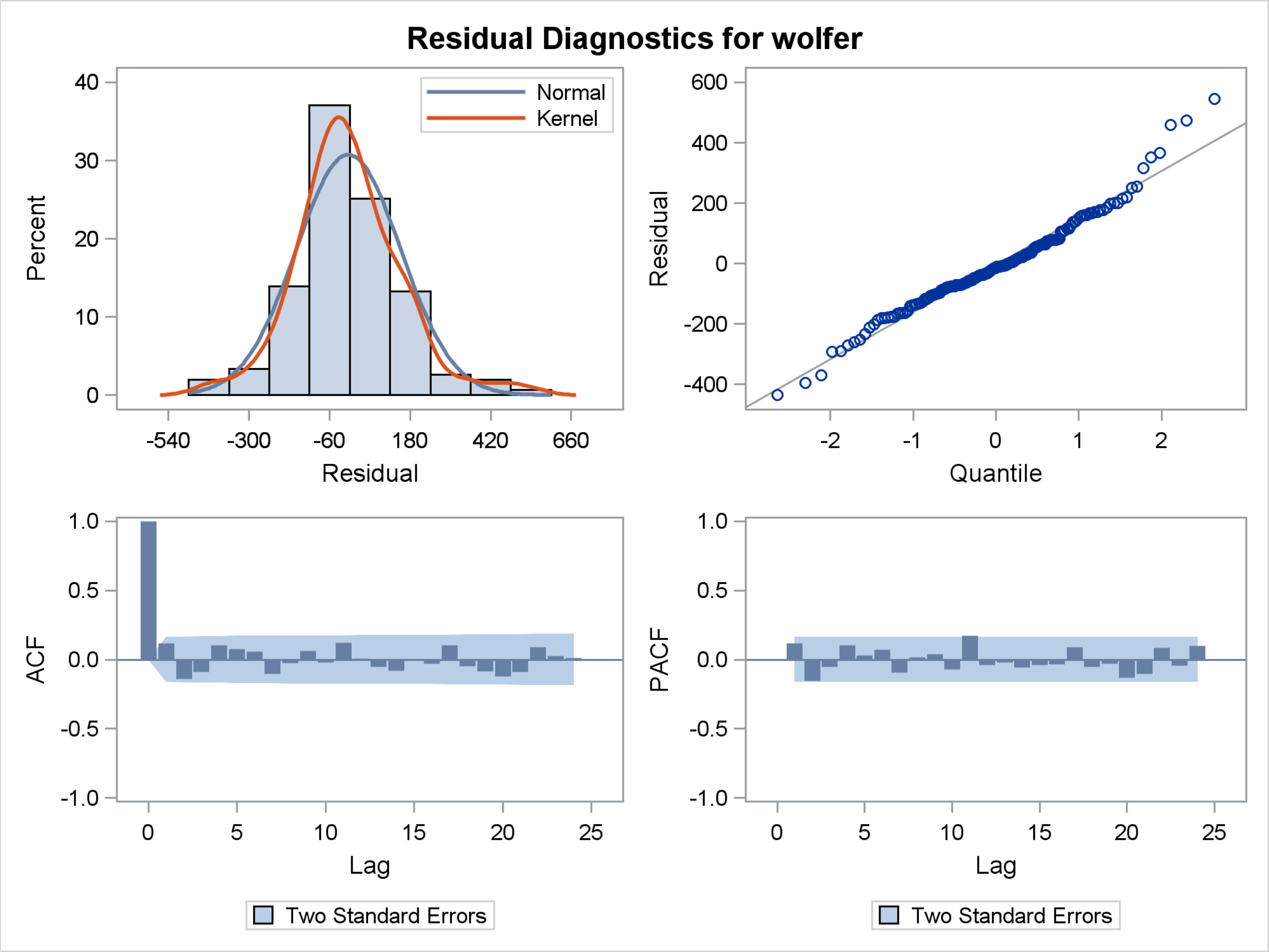


Residual Diagnostics

If the fitted model is appropriate for the given data, then the corresponding one-step-ahead residuals should be approximately *white*—that is, uncorrelated—and approximately normal. Moreover, the residuals should not display any discernible pattern. You can detect departures from these conditions graphically. Different residual diagnostic plots can be requested by using the PLOT= option in the ESTIMATE statement.

The normality can be checked by examining the histogram and the normal quantile plot of residuals. The whiteness can be checked by examining the ACF and PACF plots that show the sample autocorrelation and sample partial-autocorrelation at different lags. The diagnostic panel shown in Figure 41.10, produced by specifying PLOT=PANEL, contains these four plots.

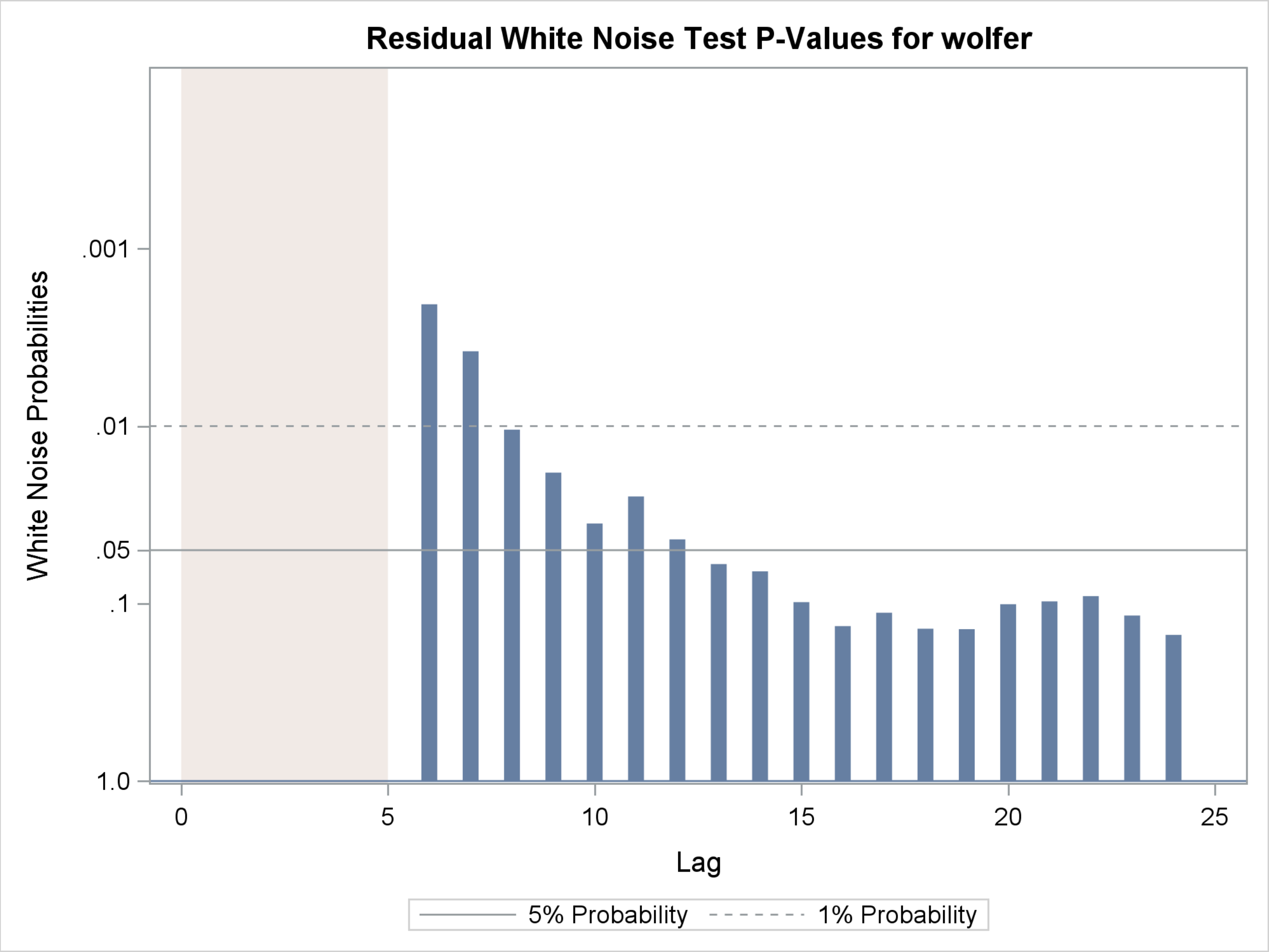
Figure 41.10 Sunspots Series: Residual Diagnostics



The residual histogram and Q-Q plot show no serious violation of normality. The histogram appears reasonably symmetric and follows the overlaid normal density curve reasonably closely. Similarly in the Q-Q plot the residuals follow the reference line fairly closely. The ACF and PACF plots also do not exhibit any violation of the whiteness assumption; the correlations at all nonzero lags seem to be insignificant.

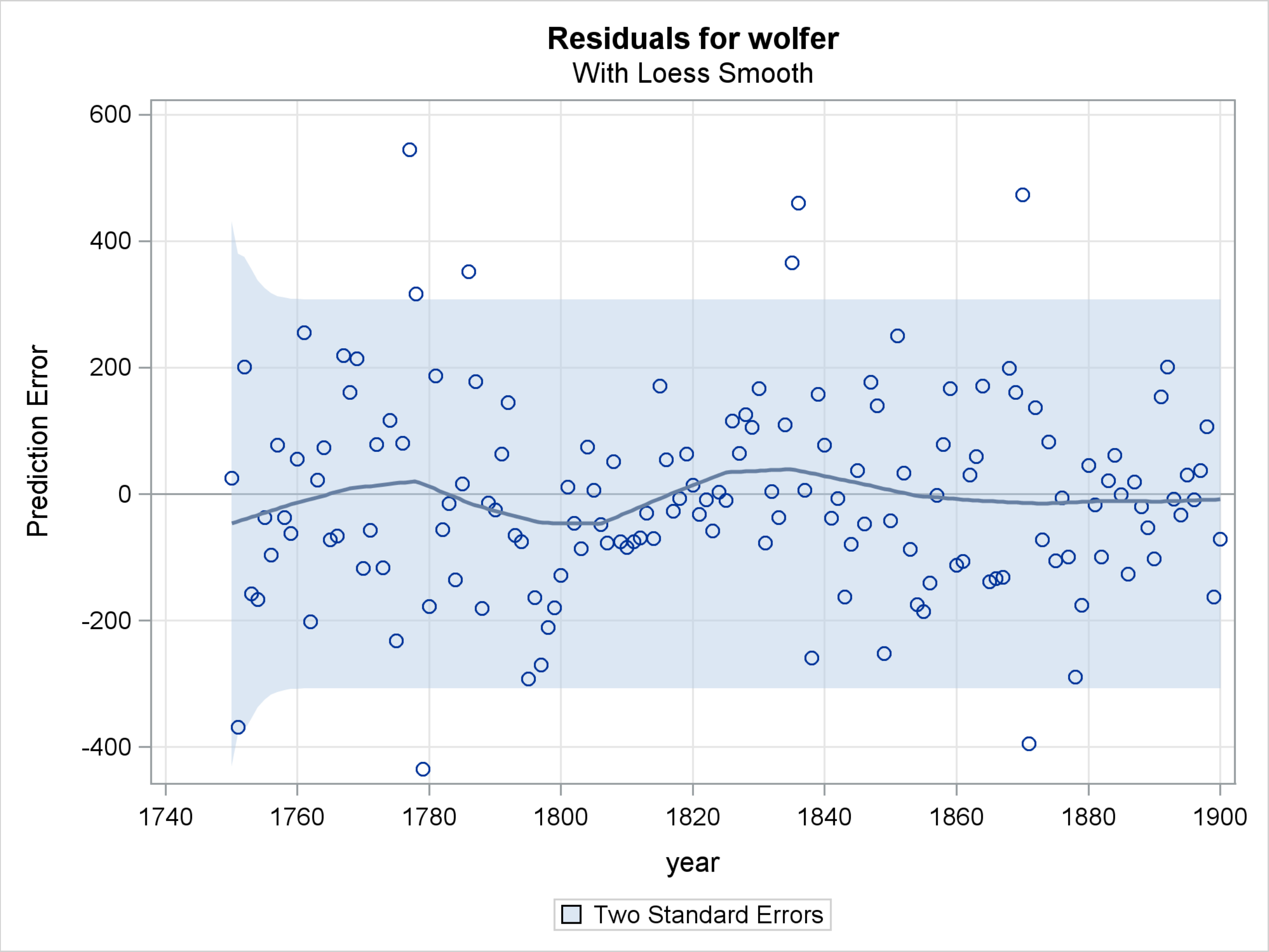
The residual whiteness can also be formally tested by using the Ljung-Box portmanteau test. The plot in Figure 41.11, produced by specifying PLOT=WN, shows the p-values of the Ljung-Box test statistics at different lags. In these plots the p-values for the first few lags, equal to the number of estimated parameters in the model, are not shown because they are always missing. This portion of the plot is shaded blue to indicate this fact. In the case of this model, five parameters are estimated so the p-values for the first five lags are not shown. The p-values are displayed on a log scale in such a way that higher bars imply more extreme test statistics. In this plot some early p-values appear extreme. However, these p-values are based on large sample theory, which suggests that these statistics should be examined for lags larger than the square root of sample size. In this example it means that the p-values for the first $\sqrt{154} \approx 12$ lags can be ignored. With this consideration, the plot shows no violation of whiteness since the p-values after the 12th lag do not appear extreme.

Figure 41.11 Sunspots Series: Ljung-Box Portmanteau Test



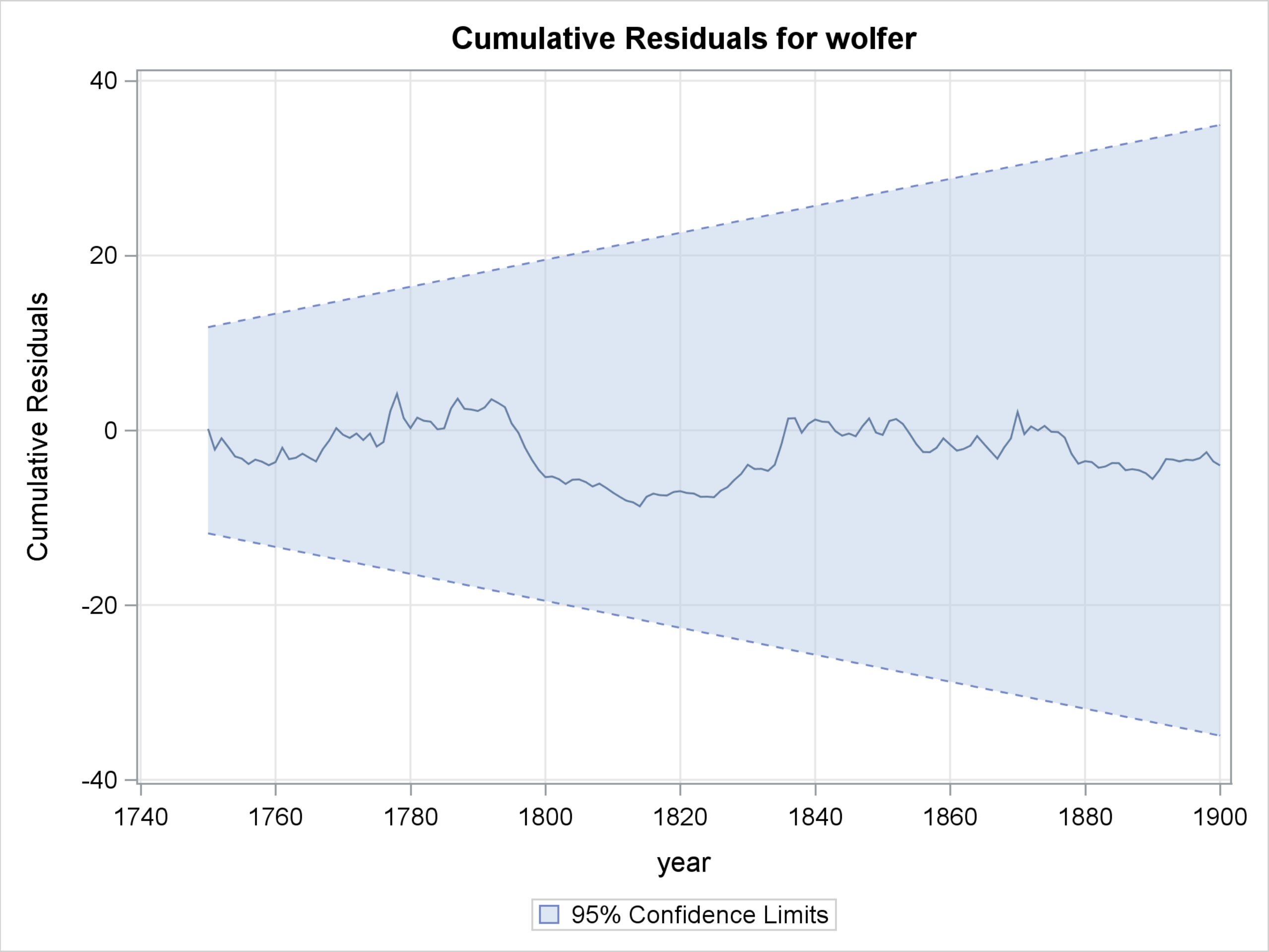
The plot in Figure 41.12, produced by specifying PLOT=LOESS, shows the residuals plotted against time with an overlaid LOESS curve. This plot is useful for checking whether any discernible pattern remains in the residuals. Here again, no significant pattern appears to be present.

Figure 41.12 Sunspots Series: Residual Loess Plot



The plot in Figure 41.13, produced by specifying PLOT=CUSUM, shows the cumulative residuals plotted against time. This plot is useful for checking structural breaks. Here, there appears to be no evidence of structural break since the cumulative residuals remain within the confidence band throughout the sample period. Similarly you can request a plot of the squared cumulative residuals by specifying PLOT=CUSUMSQ.

Figure 41.13 Sunspots Series: CUSUM Plot



Brockwell and Davis (1991) can be consulted for additional information on diagnosing residuals. For more information about CUSUM and CUSUMSQ plots, you can consult Harvey (1989).

Forecast and Series Decomposition Plots

You can use the PLOT= option in the FORECAST statement to obtain the series forecast plot and the series decomposition plots. The series decomposition plots show the result of successively adding different components in the model starting with the trend component. The IRREGULAR component is left out of this process. The following two plots, produced by specifying PLOT=DECOMP, show the results of successive component addition for this example. The first plot, shown in Figure 41.14, shows the smoothed trend component and the second plot, shown in Figure 41.15, shows the sum of smoothed trend and cycle.

Figure 41.14 Sunspots Series: Smoothed Trend

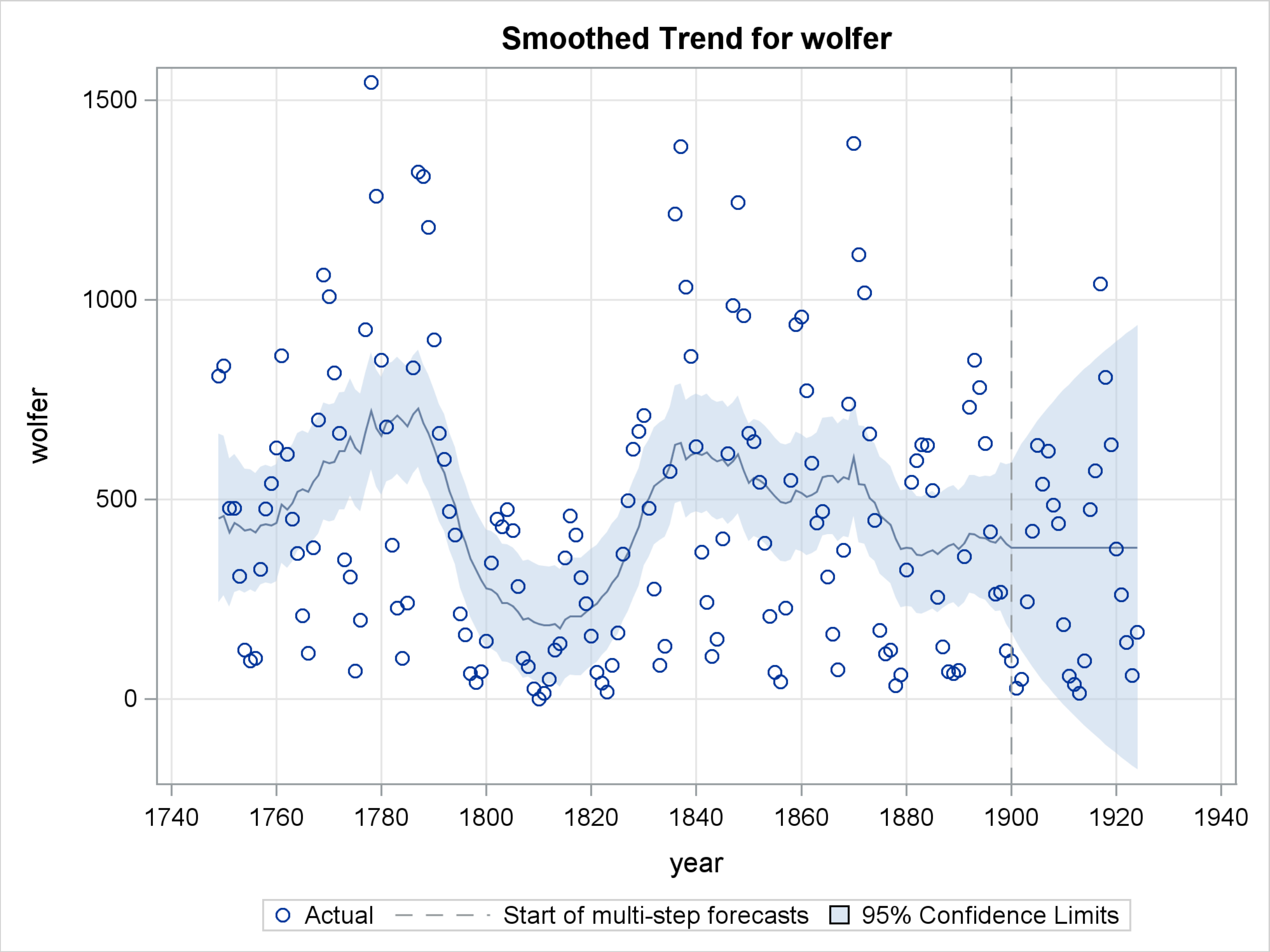
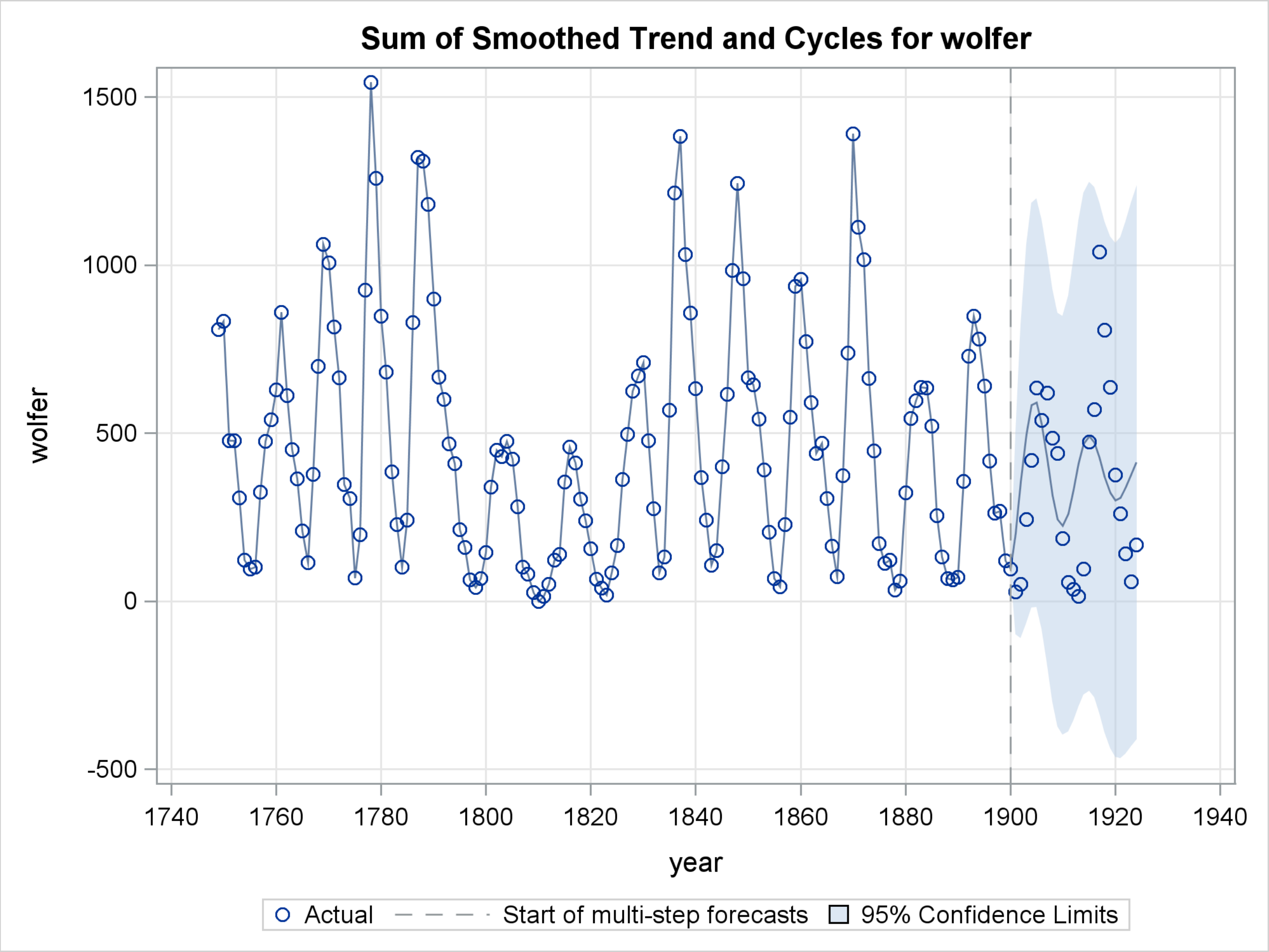
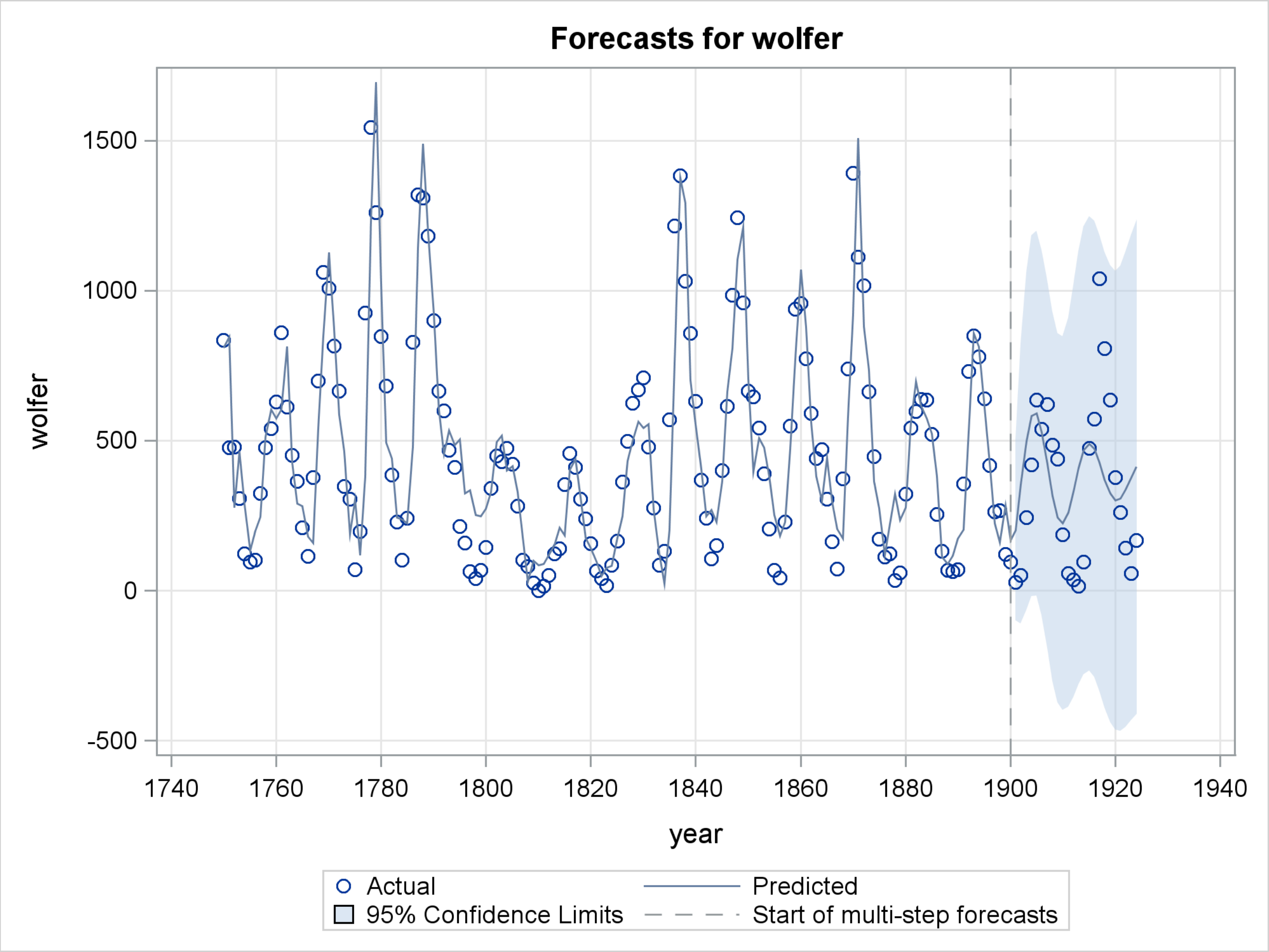


Figure 41.15 Sunspots Series: Smoothed Trend plus Cycle



Finally, Figure 41.16 shows the forecast plot.

Figure 41.16 Sunspots Series: Series Forecasts



ODS Table Names

The UCM procedure assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 41.2.

Table 41.2 ODS Tables Produced by PROC UCM

ODS Table Name	Description	Statement	Option
Tables Summarizing the Estimation and Forecast Spans			
EstimationSpan	Estimation span summary information		default
ForecastSpan	Forecast span summary information		default

Table 41.2 *continued*

ODS Table Name	Description	Statement	Option
Tables Related to Model Parameters			
ConvergenceStatus	Convergence status of the estimation process		default
FixedParameters	Fixed parameters in the model		default
InitialParameters	Initial estimates of the free parameters		default
ParameterEstimates	Final estimates of the free parameters		default
Tables Related to Model Information and Diagnostics			
BlockSeasonDescription	Information about the block seasonals in the model		default
ComponentSignificance	Significance analysis of the components in the model		default
CycleDescription	Information about the cycles in the model		default
FitStatistics	Fit statistics based on the one-step-ahead predictions		default
FitSummary	Likelihood-based fit statistics		default
OutlierSummary	Summary table of the detected outliers		default
SeasonDescription	Information about the seasonals in the model		default
SeasonHarmonics	Summary of harmonics in a trigonometric seasonal component	SEASON	PRINT=HARMONICS
SplineSeasonDescription	Information about the spline-seasonals in the model		default
TrendInformation	Summary information of the level and slope components		default
Tables Related to Filtered Component Estimates			
FilteredAutoReg	Filtered estimate of an autoreg component	AUTOREG	PRINT=FILTER
FilteredBlockSeason	Filtered estimate of a block seasonal component	BLOCKSEASON	PRINT=FILTER
FilteredCycle	Filtered estimate of a cycle component	CYCLE	PRINT=FILTER
FilteredIrregular	Filtered estimate of the irregular component	IRREGULAR	PRINT=FILTER
FilteredLevel	Filtered estimate of the level component	LEVEL	PRINT=FILTER

Table 41.2 *continued*

ODS Table Name	Description	Statement	Option
FilteredRandomReg	Filtered estimate of the time-varying random-regression coefficient	RANDOMREG	PRINT=FILTER
FilteredSeason	Filtered estimate of a seasonal component	SEASON	PRINT=FILTER
FilteredSlope	Filtered estimate of the slope component	SLOPE	PRINT=FILTER
FilteredSplineReg	Filtered estimate of the time-varying spline-regression coefficient	SPLINEREG	PRINT=FILTER
FilteredSplineSeason	Filtered estimate of a spline-seasonal component	SPLINESEASON	PRINT=FILTER
Tables Related to Smoothed Component Estimates			
SmoothedAutoReg	Smoothed estimate of an autoreg component	AUTOREG	PRINT=SMOOTH
SmoothedBlockSeason	Smoothed estimate of a block seasonal component	BLOCKSEASON	PRINT=SMOOTH
SmoothedCycle	Smoothed estimate of the cycle component	CYCLE	PRINT=SMOOTH
SmoothedIrregular	Smoothed estimate of the irregular component	IRREGULAR	PRINT=SMOOTH
SmoothedLevel	Smoothed estimate of the level component	LEVEL	PRINT=SMOOTH
SmoothedRandomReg	Smoothed estimate of the time-varying random-regression coefficient	RANDOMREG	PRINT=SMOOTH
SmoothedSeason	Smoothed estimate of a seasonal component	SEASON	PRINT=SMOOTH
SmoothedSlope	Smoothed estimate of the slope component	SLOPE	PRINT=SMOOTH
SmoothedSplineReg	Smoothed estimate of the time-varying spline-regression coefficient	SPLINEREG	PRINT=SMOOTH
SmoothedSplineSeason	Smoothed estimate of a spline-seasonal component	SPLINESEASON	PRINT=SMOOTH
Tables Related to Series Decomposition and Forecasting			
FilteredAllExceptIrreg	Filtered estimate of sum of all components except the irregular component	FORECAST	PRINT=FDECOMP
FilteredTrend	Filtered estimate of trend	FORECAST	PRINT= FDECOMP

Table 41.2 *continued*

ODS Table Name	Description	Statement	Option
FilteredTrendReg	Filtered estimate of trend plus regression	FORECAST	PRINT=FDECOMP
FilteredTrendRegCyc	Filtered estimate of trend plus regression plus cycles and autoreg	FORECAST	PRINT=FDECOMP
Forecasts	Dependent series forecasts		default
PostSamplePrediction	Forecasting performance in the holdout period	FORECAST	BACK=
SmoothedAllExceptIrreg	Smoothed estimate of sum of all components except the irregular component	FORECAST	PRINT=DECOMP
SmoothedTrend	Smoothed estimate of trend	FORECAST	PRINT= DECOMP
SmoothedTrendReg	Smoothed estimate of trend plus regression	FORECAST	PRINT=DECOMP
SmoothedTrendRegCyc	Smoothed estimate of trend plus regression plus cycles and autoreg	FORECAST	PRINT=DECOMP

NOTE: The tables are related to a single series within a BY group. In the case of models that contain multiple cycles, seasonal components, or block seasonal components, the corresponding component estimate tables are sequentially numbered. For example, if a model contains two cycles and a seasonal component and the PRINT=SMOOTH option is used for each of them, the ODS tables containing the smoothed estimates will be named SmoothedCycle1, SmoothedCycle2, and SmoothedSeason. Note that the seasonal table is not numbered because there is only one seasonal component. There are some exceptions to this numbering rule: the tables, FilteredRandomReg, SmoothedRandomReg, FilteredSplineReg, and SmoothedSplineReg, are always numbered starting with zero.

ODS Graph Names

Statistical procedures use ODS Graphics to create graphs as part of their output. ODS Graphics is described in detail in Chapter 21, "Statistical Graphics Using ODS" (*SAS/STAT User's Guide*).

Before you create graphs, ODS Graphics must be enabled (for example, with the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" in that chapter.

The overall appearance of graphs is controlled by ODS styles. Styles and other aspects of using ODS Graphics are discussed in the section "A Primer on ODS Statistical Graphics" in that chapter.

You can reference every graph produced through ODS Graphics with a name. The names of the graphs that PROC UCM generates are listed in Table 41.3, along with the required statements and options.

Table 41.3 ODS Graphics Produced by PROC UCM

ODS Graph Name	Description	Statement	Option
Plots Related to Residual Analysis			
ErrorACFPlot	Prediction error autocorrelation plot	ESTIMATE	PLOT=ACF
ErrorPACFPlot	Prediction error partial-autocorrelation plot	ESTIMATE	PLOT=PACF
ErrorHistogram	Prediction error histogram	ESTIMATE	PLOT=NORMAL
ErrorQQPlot	Prediction error normal quantile plot	ESTIMATE	PLOT=QQ
ErrorPlot	Plot of prediction errors	ESTIMATE	PLOT=RESIDUAL
ErrorWhiteNoiseLogProbPlot	Plot of *p*-values at different lags for the Ljung-Box portmanteau white noise test statistics	ESTIMATE	PLOT=WN
CUSUMPlot	Plot of cumulative residuals	ESTIMATE	PLOT=CUSUM
CUSUMSQPlot	Plot of cumulative squared residuals	ESTIMATE	PLOT=CUSUMSQ
ModelPlot	Plot of one-step-ahead forecasts in the estimation span	ESTIMATE	PLOT=MODEL
PanelResidualPlot	Panel of residual diagnostic plots	ESTIMATE	PLOT=PANEL
ResidualLoessPlot	Time series plot of residuals with superimposed LOESS smoother	ESTIMATE	PLOT=LOESS
Plots Related to Filtered Component Estimates			
FilteredAutoregPlot	Plot of filtered autoreg component	AUTOREG	PLOT=FILTER
FilteredBlockSeasonPlot	Plot of filtered block season component	BLOCKSEASON	PLOT=FILTER
FilteredCyclePlot	Plot of filtered cycle component	CYCLE	PLOT=FILTER
FilteredIrregularPlot	Plot of filtered irregular component	IRREGULAR	PLOT=FILTER
FilteredLevelPlot	Plot of filtered level component	LEVEL	PLOT=FILTER
FilteredRandomRegPlot	Plot of filtered time-varying regression coefficient	RANDOMREG	PLOT=FILTER
FilteredSeasonPlot	Plot of filtered season component	SEASON	PLOT=FILTER
FilteredSlopePlot	Plot of filtered slope component	SLOPE	PLOT=FILTER
FilteredSplineRegPlot	Plot of filtered time-varying regression coefficient	SPLINEREG	PLOT=FILTER

Table 41.3 *continued*

ODS Graph Name	Description	Statement	Option
FilteredSplineSeasonPlot	Plot of filtered spline-season component	SPLINESEASON	PLOT=FILTER
AnnualSeasonPlot	Plot of annual variation in the filtered season component	SEASON	PLOT=F_ANNUAL
Plots Related to Smoothed Component Estimates			
SmoothedAutoregPlot	Plot of smoothed autoreg component	AUTOREG	PLOT=SMOOTH
SmoothedBlockSeasonPlot	Plot of smoothed block season component	BLOCKSEASON	PLOT=SMOOTH
SmoothedCyclePlot	Plot of smoothed cycle component	CYCLE	PLOT=SMOOTH
SmoothedIrregularPlot	Plot of smoothed irregular component	IRREGULAR	PLOT=SMOOTH
SmoothedLevelPlot	Plot of smoothed level component	LEVEL	PLOT=SMOOTH
SmoothedRandomRegPlot	Plot of smoothed time-varying regression coefficient	RANDOMREG	PLOT=SMOOTH
SmoothedSeasonPlot	Plot of smoothed season component	SEASON	PLOT=SMOOTH
SmoothedSlopePlot	Plot of smoothed slope component	SLOPE	PLOT=SMOOTH
SmoothedSplineRegPlot	Plot of smoothed time-varying regression coefficient	SPLINEREG	PLOT=SMOOTH
SmoothedSplineSeasonPlot	Plot of smoothed spline-season component	SPLINESEASON	PLOT=SMOOTH
AnnualSeasonPlot	Plot of annual variation in the smoothed season component	SEASON	PLOT=S_ANNUAL
Plots Related to Series Decomposition and Forecasting			
ForecastsOnlyPlot	Series forecasts beyond the historical period	FORECAST	DEFAULT
ForecastsPlot	One-step-ahead as well as multistep-ahead forecasts	FORECAST	PLOT=FORECASTS
FilteredAllExceptIrregPlot	Plot of sum of all filtered components except the irregular component	FORECAST	PLOT= FDECOMP
FilteredTrendPlot	Plot of filtered trend	FORECAST	PLOT= FDECOMP

Table 41.3 *continued*

ODS Graph Name	Description	Statement	Option
FilteredTrendRegCycPlot	Plot of sum of filtered trend, cycles, and regression effects	FORECAST	PLOT= FDECOMP
FilteredTrendRegPlot	Plot of filtered trend plus regression effects	FORECAST	PLOT= FDECOMP
SmoothedAllExceptIrregPlot	Plot of sum of all smoothed components except the irregular component	FORECAST	PLOT= DECOMP
SmoothedTrendPlot	Plot of smoothed trend	FORECAST	PLOT= TREND
SmoothedTrendRegPlot	Plot of smoothed trend plus regression effects	FORECAST	PLOT= DECOMP
SmoothedTrendRegCycPlot	Plot of sum of smoothed trend, cycles, and regression effects	FORECAST	PLOT= DECOMP
FilteredAllExceptIrregVarPlot	Plot of standard error of sum of all filtered components except the irregular	FORECAST	PLOT= FDECOMPVAR
FilteredTrendVarPlot	Plot of standard error of filtered trend	FORECAST	PLOT= FDECOMPVAR
FilteredTrendRegVarPlot	Plot of standard error of filtered trend plus regression effects	FORECAST	PLOT= FDECOMPVAR
FilteredTrendRegCycVarPlot	Plot of standard error of filtered trend, cycles, and regression effects	FORECAST	PLOT= FDECOMPVAR
SmoothedAllExceptIrregVarPlot	Plot of standard error of sum of all smoothed components except the irregular	FORECAST	PLOT= DECOMPVAR
SmoothedTrendVarPlot	Plot of standard error of smoothed trend	FORECAST	PLOT= DECOMPVAR
SmoothedTrendRegVarPlot	Plot of standard error of smoothed trend plus regression effects	FORECAST	PLOT= DECOMPVAR
SmoothedTrendRegCycVarPlot	Plot of standard error of smoothed trend, cycles, and regression effects	FORECAST	PLOT= DECOMPVAR

OUTFOR= Data Set

You can use the OUTFOR= option in the FORECAST statement to store the series and component forecasts produced by the procedure. This data set contains the following columns:

- the BY variables
- the ID variable. If an ID variable is not specified, then a numerical variable, _ID_, is created that contains the observation numbers from the input data set.
- the dependent series and the predictor series
- FORECAST, a numerical variable containing the one-step-ahead predicted values and the multistep forecasts
- RESIDUAL, a numerical variable containing the difference between the actual and forecast values
- STD, a numerical variable containing the standard error of prediction
- LCL and UCL, numerical variables containing the lower and upper forecast confidence limits
- S_SERIES and VS_SERIES, numerical variables containing the smoothed values of the dependent series and their variances
- S_IRREG and VS_IRREG, numerical variables containing the smoothed values of the irregular component and their variances. These variables are present only if the model has an irregular component.
- F_LEVEL, VF_LEVEL, S_LEVEL, and VS_LEVEL, numerical variables containing the filtered and smoothed values of the level component and the respective variances. These variables are present only if the model has a level component.
- F_SLOPE, VF_SLOPE, S_SLOPE, and VS_SLOPE, numerical variables containing the filtered and smoothed values of the slope component and the respective variances. These variables are present only if the model has a slope component.
- F_AUTOREG, VF_AUTOREG, S_AUTOREG, and VS_AUTOREG, numerical variables containing the filtered and smoothed values of the autoreg component and the respective variances. These variables are present only if the model has an autoreg component.
- F_CYCLE, VF_CYCLE, S_CYCLE, and VS_CYCLE, numerical variables containing the filtered and smoothed values of the cycle component and the respective variances. If there are multiple cycles in the model, these variables are sequentially numbered as F_CYCLE1, F_CYCLE2, etc. These variables are present only if the model has at least one cycle component.
- F_SEASON, VF_SEASON, S_SEASON, and VS_SEASON, numerical variables containing the filtered and smoothed values of the season component and the respective variances. If there are multiple seasons in the model, these variables are sequentially numbered as F_SEASON1, F_SEASON2, etc. These variables are present only if the model has at least one season component.
- F_BLKSEAS, VF_BLKSEAS, S_BLKSEAS, and VS_BLKSEAS, numerical variables containing the filtered and smoothed values of the blockseason component and the respective variances. If there are multiple block seasons in the model, these variables are sequentially numbered as F_BLKSEAS1, F_BLKSEAS2, etc.

- F_SPLSEAS, VF_SPLSEAS, S_SPLSEAS, and VS_SPLSEAS, numerical variables containing the filtered and smoothed values of the splineseason component and the respective variances. If there are multiple spline seasons in the model, these variables are sequentially numbered as F_SPLSEAS1, F_SPLSEAS2, etc. These variables are present only if the model has at least one splineseason component.

- Filtered and smoothed estimates, and their variances, of the time-varying regression coefficients of the variables specified in the RANDOMREG and SPLINEREG statements. A variable is not included if its coefficient is time-invariant, that is, if the associated disturbance variance is zero.

- S_TREG and VS_TREG, numerical variables containing the smoothed values of level plus regression component and their variances. These variables are present only if the model has at least one predictor variable or has dependent lags.

- S_TREGCYC and VS_TREGCYC, numerical variables containing the smoothed values of level plus regression plus cycle component and their variances. These variables are present only if the model has at least one cycle or an autoreg component.

- S_NOIRREG and VS_NOIRREG, numerical variables containing the smoothed values of the sum of all components except the irregular component and their variances. These variables are present only if the model has at least one seasonal or block seasonal component.

OUTEST= Data Set

You can use the OUTEST= option in the ESTIMATE statement to store the model parameters and the related estimation details. This data set contains the following columns:

- the BY variables

- COMPONENT, a character variable containing the name of the component corresponding to the parameter being described

- PARAMETER, a character variable containing the parameter name

- TYPE, a character variable indicating whether the parameter value was fixed by the user or estimated

- _STATUS_, a character variable indicating whether the parameter estimation process converged or failed or there was an error of some other kind

- ESTIMATE, a numerical variable containing the parameter estimate

- STD, a numerical variable containing the standard error of the parameter estimate. This has a missing value if the parameter value is fixed.

- TVALUE, a numerical variable containing the *t*-statistic. This has a missing value if the parameter value is fixed.

- PVALUE, a numerical variable containing the *p*-value. This has a missing value if the parameter value is fixed.

Statistics of Fit

This section explains the goodness-of-fit statistics reported to measure how well the specified model fits the data.

First the various statistics of fit that are computed using the prediction errors, $y_t - \hat{y}_t$, are considered. In these formulas, *n* is the number of nonmissing prediction errors and *k* is the number of fitted parameters in the model. Moreover, the sum of squared errors, $SSE = \sum (y_t - \hat{y}_t)^2$, and the total sum of squares for the series corrected for the mean, $SST = \sum (y_t - \overline{y})^2$, where $\overline{y}$ is the series mean, and the sums are over all the nonmissing prediction errors.

Mean Squared Error
The mean squared prediction error, $MSE = \frac{1}{n} SSE$

Root Mean Squared Error
The root mean square error, RMSE = $\sqrt{MSE}$

Mean Absolute Percent Error
The mean absolute percent prediction error, MAPE = $\frac{100}{n} \sum_{t=1}^{n} |(y_t - \hat{y}_t)/y_t|$.
The summation ignores observations where $y_t = 0$.

R-square
The R-square statistic, $R^2 = 1 - SSE/SST$.
If the model fits the series badly, the model error sum of squares, *SSE*, might be larger than *SST* and the R-square statistic will be negative.

Adjusted R-square
The adjusted R-square statistic, $1 - (\frac{n-1}{n-k})(1 - R^2)$

Amemiya's Adjusted R-square
Amemiya's adjusted R-square, $1 - (\frac{n+k}{n-k})(1 - R^2)$

Random Walk R-square
The random walk R-square statistic (Harvey's R-square statistic that uses the random walk model for comparison), $1 - (\frac{n-1}{n})SSE/RWSSE$, where $RWSSE = \sum_{t=2}^{n} (y_t - y_{t-1} - \mu)^2$, and $\mu = \frac{1}{n-1} \sum_{t=2}^{n} (y_t - y_{t-1})$

Maximum Percent Error
The largest percent prediction error, $100 \max((y_t - \hat{y}_t)/y_t)$. In this computation the observations where $y_t = 0$ are ignored.

The likelihood-based fit statistics are reported separately (see the section "The UCMs as State Space Models" on page 2788). They include the full log likelihood (L_∞), the diffuse part of the log likelihood, the normalized residual sum of squares, and several information criteria: AIC, AICC, HQIC, BIC, and CAIC. Let q denote the number of estimated parameters, n be the number of nonmissing measurements in the estimation span, and d be the number of diffuse elements in the initial state vector that are successfully initialized during the Kalman filtering process. Moreover, let $n^* = (n - d)$. The reported information criteria, all in smaller-is-better form, are described in Table 41.4:

Table 41.4 Information Criteria

Criterion	Formula	Reference
AIC	$-2L_\infty + 2q$	Akaike (1974)
AICC	$-2L_\infty + 2qn^*/(n^* - q - 1)$	Hurvich and Tsai (1989) Burnham and Anderson (1998)
HQIC	$-2L_\infty + 2q \log\log(n^*)$	Hannan and Quinn (1979)
BIC	$-2L_\infty + q \log(n^*)$	Schwarz (1978)
CAIC	$-2L_\infty + q(\log(n^*) + 1)$	Bozdogan (1987)

Examples: UCM Procedure

Example 41.1: The Airline Series Revisited

The series in this example, the monthly airline passenger series, has already been discussed earlier; see the section "A Seasonal Series with Linear Trend" on page 2745. Recall that the series consists of monthly numbers of international airline travelers (from January 1949 to December 1960). Here additional output features of the UCM procedure are illustrated, such as how to use the ESTIMATE and FORECAST statements to limit the span of the data used in parameter estimation and forecasting. The following statements fit a BSM to the logarithm of the airline passenger numbers. The disturbance variance for the slope component is held fixed at value 0; that is, the trend is locally linear with constant slope. In order to evaluate the performance of the fitted model on observed data, some of the observed data are withheld during parameter estimation and forecast computations. The observations in the last two years, years 1959 and 1960, are not used in parameter estimation, while the observations in the last year, year 1960, are not used in the forecasting computations. This is done using the BACK= option in the ESTIMATE and FORECAST statements. In addition, a panel of residual diagnostic plots is obtained using the PLOT=PANEL option in the ESTIMATE statement.

```
data seriesG;
   set sashelp.air;
   logair = log(air);
run;

proc ucm data = seriesG;
   id date interval = month;
   model logair;
   irregular;
   level;
   slope var = 0 noest;
   season length = 12 type=trig;
   estimate back=24 plot=panel;
   forecast back=12 lead=24 print=forecasts;
run;
```

The following tables display the summary of data used in estimation and forecasting (Output 41.1.1 and Output 41.1.2). These tables provide simple summary statistics for the estimation and forecast spans; they include useful information such as the beginning and ending dates of the span, the number of nonmissing values, etc.

Output 41.1.1 Observation Span Used in Parameter Estimation (partial output)

Variable	Type	First	Last	Nobs	Mean
logair	Dependent	JAN1949	DEC1958	120	5.43035

Output 41.1.2 Observation Span Used in Forecasting (partial output)

Variable	Type	First	Last	Nobs	Mean
logair	Dependent	JAN1949	DEC1959	132	5.48654

The following tables display the fixed parameters in the model, the preliminary estimates of the free parameters, and the final estimates of the free parameters (Output 41.1.3, Output 41.1.4, and Output 41.1.5).

Output 41.1.3 Fixed Parameters in the Model

The UCM Procedure

Fixed Parameters in the Model

Component	Parameter	Value
Slope	Error Variance	0

Output 41.1.4 Starting Values for the Parameters to Be Estimated

Preliminary Estimates of the Free Parameters

Component	Parameter	Estimate
Irregular	Error Variance	6.64120
Level	Error Variance	2.49045
Season	Error Variance	1.26676

Output 41.1.5 Maximum Likelihood Estimates of the Free Parameters

Final Estimates of the Free Parameters

			Approx		Approx
Component	Parameter	Estimate	Std Error	t Value	Pr > |t|
Irregular	Error Variance	0.00018686	0.0001212	1.54	0.1233
Level	Error Variance	0.00040314	0.0001566	2.57	0.0100
Season	Error Variance	0.00000350	1.66319E-6	2.10	0.0354

Two types of goodness-of-fit statistics are reported after a model is fit to the series (see Output 41.1.6 and Output 41.1.7). The first type is the likelihood-based goodness-of-fit statistics, which include the full likelihood of the data, the diffuse portion of the likelihood (see the section "Details: UCM Procedure" on page 2782), and the information criteria. The second type of statistics is based on the raw residuals, residual = observed – predicted. If the model is nonstationary, then one-step-ahead predictions are not available for some initial observations, and the number of values used in computing these fit statistics will be different from those used in computing the likelihood-based test statistics.

Output 41.1.6 Likelihood-Based Fit Statistics for the Airline Data

Likelihood Based Fit Statistics

Statistic	Value
Full Log Likelihood	180.63
Diffuse Part of Log Likelihood	-13.93
Non-Missing Observations Used	120
Estimated Parameters	3
Initialized Diffuse State Elements	13
Normalized Residual Sum of Squares	107
AIC (smaller is better)	-355.3
BIC (smaller is better)	-347.2
AICC (smaller is better)	-355
HQIC (smaller is better)	-352
CAIC (smaller is better)	-344.2

Output 41.1.7 Residuals-Based Fit Statistics for the Airline Data

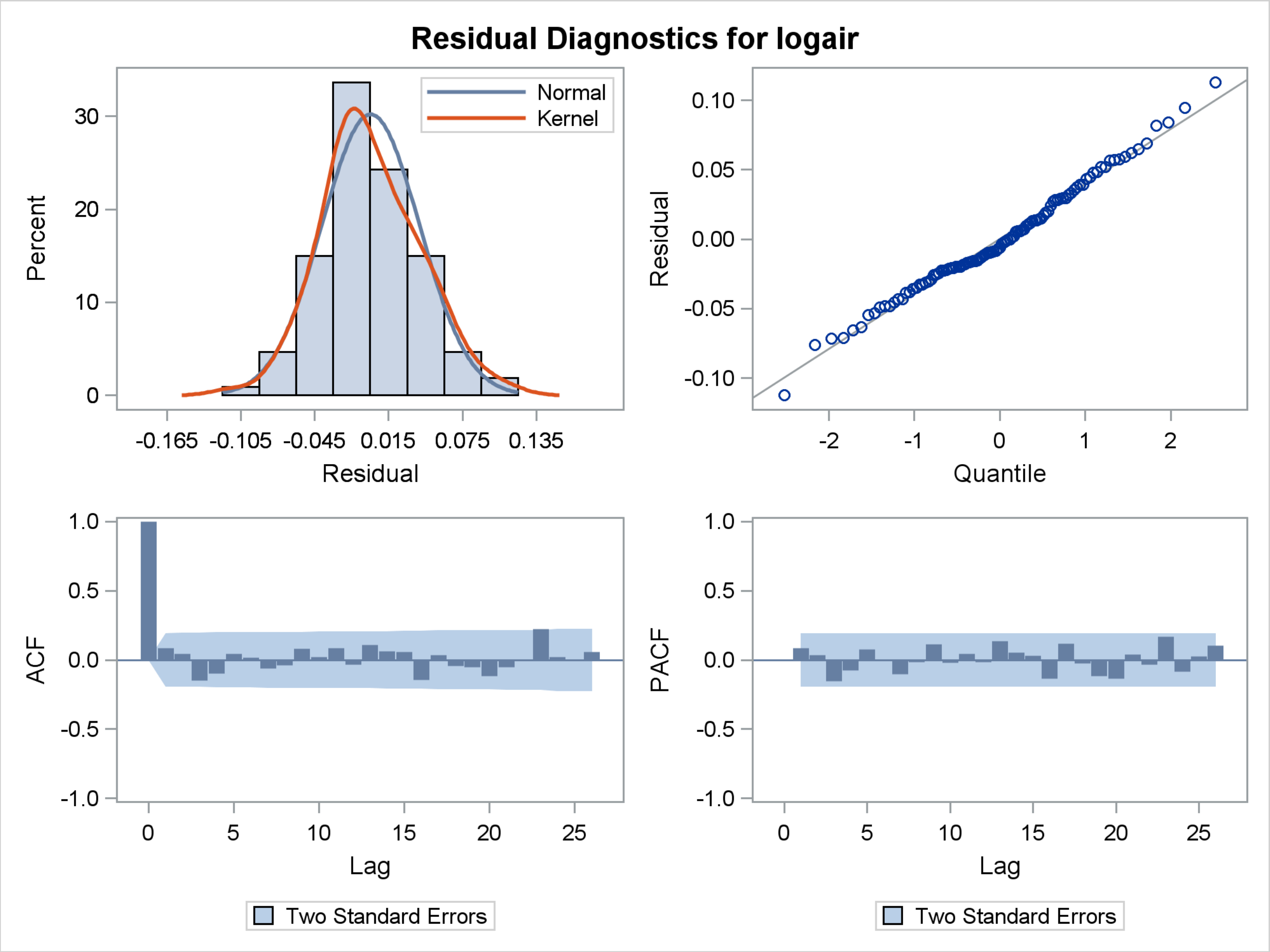
Fit Statistics Based on Residuals

Mean Squared Error	0.00156
Root Mean Squared Error	0.03944
Mean Absolute Percentage Error	0.57677
Maximum Percent Error	2.19396
R-Square	0.98705
Adjusted R-Square	0.98680
Random Walk R-Square	0.86370
Amemiya's Adjusted R-Square	0.98630

Number of non-missing residuals used for computing the fit statistics = 107

The diagnostic plots based on the one-step-ahead residuals are shown in Output 41.1.8. The residual histogram and the Q-Q plot show no reasons to question the approximate normality of the residual distribution. The remaining plots check for the *whiteness* of the residuals. The sample correlation plots, the autocorrelation function (ACF) and the partial autocorrelation function (PACF), also do not show any significant violations of the whiteness of the residuals. Therefore, on the whole, the model seems to fit the data well.

Output 41.1.8 Residual Diagnostics for the Airline Series Using a BSM



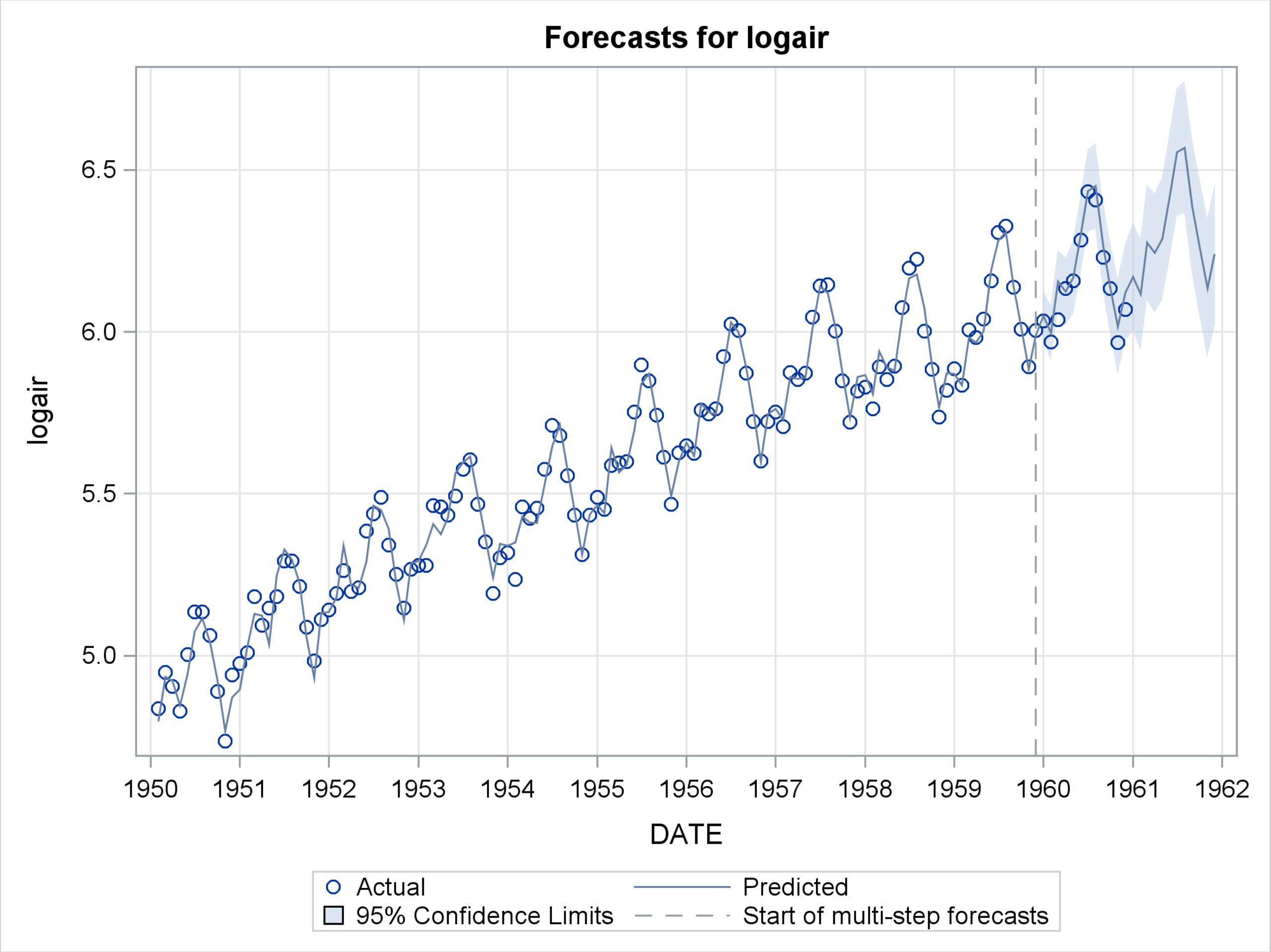
The forecasts are given in Output 41.1.9. In order to save the space, the upper and lower confidence limit columns are dropped from the output, and only the rows corresponding to the year 1960 are shown. Recall that the actual measurements in the years 1959 and 1960 were withheld during the parameter estimation, and the ones in 1960 were not used in the forecast computations.

Output 41.1.9 Forecasts for the Airline Data

Obs	date	Forecast	StdErr	logair	Residual
133	JAN60	6.050	0.038	6.033	-0.017
134	FEB60	5.996	0.044	5.969	-0.027
135	MAR60	6.156	0.049	6.038	-0.118
136	APR60	6.124	0.053	6.133	0.010
137	MAY60	6.168	0.058	6.157	-0.011
138	JUN60	6.303	0.061	6.282	-0.021
139	JUL60	6.435	0.065	6.433	-0.002
140	AUG60	6.450	0.068	6.407	-0.043
141	SEP60	6.265	0.071	6.230	-0.035
142	OCT60	6.138	0.073	6.133	-0.005
143	NOV60	6.015	0.075	5.966	-0.049
144	DEC60	6.121	0.077	6.068	-0.053

The figure Output 41.1.10 shows the forecast plot. The forecasts in the year 1960 show that the model predictions were quite good.

Output 41.1.10 Forecast Plot of the Airline Series Using a BSM



Example 41.2: Variable Star Data

The series in this example is studied in detail in Bloomfield (2000). This series consists of brightness measurements (magnitude) of a variable star taken at midnight for 600 consecutive days. The data can be downloaded from a time series archive maintained by the University of York, England (http://www.york.ac.uk/depts/maths/data/ts/welcome.htm (series number 26)). The following DATA step statements read the data in a SAS data set.

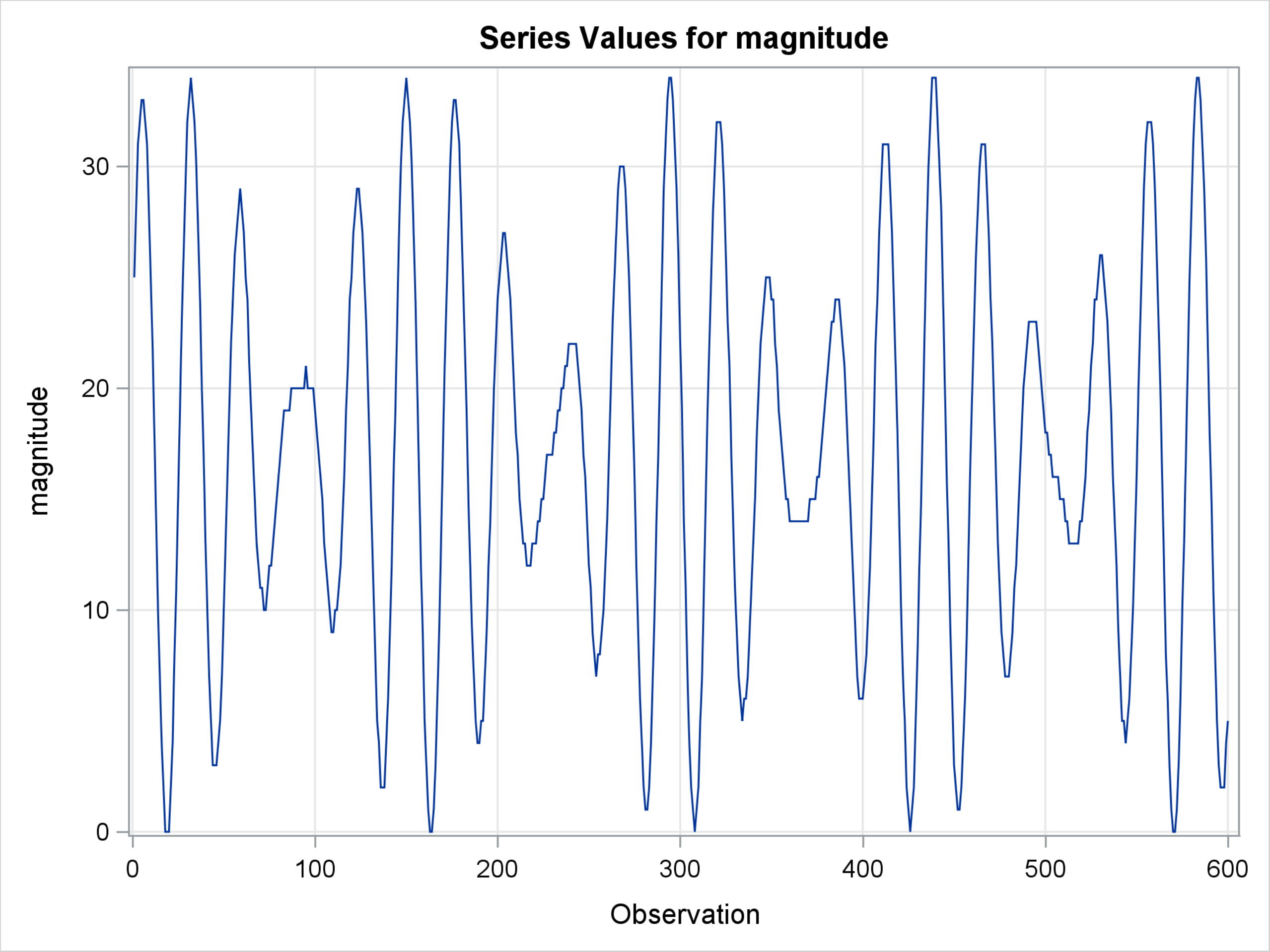
```
data star;
   input magnitude @@;
   day = _n_;
datalines;
  25  28  31  32  33  33  32  31  28   25   22   18
  14  10   7   4   2   0   0   0   2    4    8   11
  15  19  23  26  29  32  33  34  33   32   30   27
  24  20  17  13  10   7   5   3   3    3    4    5
   7  10  13  16  19  22  24  26  27   28   29   28
  27  25  24  21  19  17  15  13  12   11   11   10
  10  11  12  12  13  14  15  16  17   18   19   19

   ... more lines ...
```

The following statements use the TIMESERIES procedure to get a timeseries plot of the series (see Output 41.2.1).

```
proc timeseries data=star plot=series;
   var magnitude;
run;
```

Output 41.2.1 Plot of Star Brightness on Successive Days



The plot clearly shows the cyclic nature of the series. Bloomfield shows that the series is very well explained by a model that includes two deterministic cycles that have periods 29.0003 and 24.0001 days, a constant term, and a simple error term. He also mentions the difficulty involved in estimating the periods from the data (Bloomfield 2000, Chapter 3). In his case the cycle periods are estimated by least squares, and the sum of squares surface has multiple local optima and ridges. The following statements show how to use the UCM procedure to fit this two-cycle model to the series. The constant term in the model is specified by holding the variance parameter of the level component to zero.

```
proc ucm data=star;
   model magnitude;
   irregular;
   level var=0 noest;
   cycle;
   cycle;
   estimate;
run;
```

The final parameter estimates and the goodness-of-fit statistics are shown in Output 41.2.2 and Output 41.2.3, respectively. The model fit appears to be good.

Output 41.2.2 Two-Cycle Model: Parameter Estimates

The UCM Procedure

Final Estimates of the Free Parameters

Component	Parameter	Estimate	Approx Std Error	t Value	Approx Pr > |t|
Irregular	Error Variance	0.09257	0.0053845	17.19	<.0001
Cycle_1	Damping Factor	1.00000	1.81175E-7	5519514	<.0001
Cycle_1	Period	29.00036	0.0022709	12770.4	<.0001
Cycle_1	Error Variance	0.00000882	5.27213E-6	1.67	0.0944
Cycle_2	Damping Factor	1.00000	2.11939E-7	4718334	<.0001
Cycle_2	Period	24.00011	0.0019128	12547.2	<.0001
Cycle_2	Error Variance	0.00000535	3.56374E-6	1.50	0.1330

Output 41.2.3 Two-Cycle Model: Goodness of Fit

Fit Statistics Based on Residuals

Mean Squared Error	0.12072
Root Mean Squared Error	0.34745
Mean Absolute Percentage Error	2.65141
Maximum Percent Error	36.38991
R-Square	0.99850
Adjusted R-Square	0.99849
Random Walk R-Square	0.97281
Amemiya's Adjusted R-Square	0.99847

Number of non-missing residuals used for computing the fit statistics = 599

A summary of the cycles in the model is given in Output 41.2.4.

Output 41.2.4 Two-Cycle Model: Summary

Name	Type	period	Rho	ErrorVar
Cycle_1	Stationary	29.00036	1.00000	0.00000882
Cycle_2	Stationary	24.00011	1.00000	0.00000535

Note that the estimated periods are the same as in Bloomfield’s model, the damping factors are nearly equal to 1.0, and the disturbance variances are very close to zero, implying persistent deterministic cycles. In fact, this model is identical to Bloomfield’s model.

Example 41.3: Modeling Long Seasonal Patterns

This example illustrates some of the techniques you can use to model long seasonal patterns in a series. If the seasonal pattern is of moderate length and the underlying dynamics are simple, then it is easily modeled by using the basic settings of the SEASON statement and these additional techniques are not needed. However, if the seasonal pattern has a long season length and/or has a complex stochastic dynamics, then the techniques discussed here can be useful. You can obtain parsimonious models for a long seasonal pattern by using an appropriate subset of trigonometric harmonics, or by using a suitable spline function, or by using a block-season pattern in combination with a seasonal component of much smaller length. You can also vary the disturbance variances of the subcomponents that combine to form the seasonal component.

The time series used in this example consists of number of calls received per shift at a call center. Each shift is six hours long, and the first shift of the day begins at midnight, resulting in four shifts per day. The observations are available from December 15, 1999, to April 30, 2000. This series is seasonal with season length 28, which is moderate, and in fact there is no particular need to use pattern approximation techniques in this case. However, it is adequate for demonstration purposes. The plan of this example is as follows. First an initial model with a full seasonal component is created. This model is used as a baseline for comparing alternate models created by the techniques that are being illustrated. In practice any candidate model is first checked for adequacy by using various diagnostic procedures. In this illustration the main focus is on the different ways a long seasonal pattern can be modeled and no model diagnostics are done for the models being entertained. The alternate models are compared by using the sum of absolute prediction errors in the holdout region.

The following DATA step statements create the input data set used in this example.

```
data callCenter;
   input calls @@;
   label calls= "Number of Calls Received in a 6 Hour Shift";
   start = '15dec99:00:00'dt;
   datetime = INTNX( 'dthour6', start, _n_-1 );
   format datetime datetime10.;
datalines;
  18    122    244    128     19    113    230    119     17    112
 219     93     14     73    139     53     11     32     74     56
  15    137    289    153     20    125    227    106     16    101
 201     92     14     94    187     69     11     59     94     21

   ... more lines ...
```

Initial exploration of the series clearly indicates that the series does not show any significant trend, and time of day and day of the week have a significant influence on the number of calls received. These considerations suggest a simple random walk trend model along with a seasonal component of season length 28, the total number of shifts in a week. The following statements specify this model. Note the PRINT=HARMONICS option in the SEASON statement, which produces a table that lists the full set of harmonics contributing to the seasonal along with the significance of their contribution. This table will be useful later in choosing a subset trigonometric model. The BACK=28 and the LEAD=28 specifications in the FORECAST statement create a holdout region of 28 observations. The sum of absolute prediction errors (SAE) in this holdout region are used to compare the different models.

```
proc ucm data=callCenter;
   id datetime interval=dthour6;
   model calls;
   irregular;
   level;
   season length=28 type=trig
      print=(harmonics);
   estimate back=28;
   forecast back=28 lead=28;
run;
```

The forecasting performance of this model in the holdout region is shown in Output 41.3.1. The sum of absolute prediction errors SAE = 516.22, which appears in the last row of the holdout analysis table.

Output 41.3.1 Predictions in the Holdout Region: Baseline Model

Obs	datetime	Actual	Forecast	Error	SAE
525	24APR00:00	12	-4.004	16.004	16.004
526	24APR00:06	136	110.825	25.175	41.179
527	24APR00:12	295	262.820	32.180	73.360
528	24APR00:18	172	145.127	26.873	100.232
529	25APR00:00	20	2.188	17.812	118.044
530	25APR00:06	127	105.442	21.558	139.602
531	25APR00:12	236	217.043	18.957	158.559
532	25APR00:18	125	114.313	10.687	169.246
533	26APR00:00	16	2.855	13.145	182.391
534	26APR00:06	108	95.202	12.798	195.189
535	26APR00:12	207	194.184	12.816	208.005
536	26APR00:18	112	97.687	14.313	222.317
537	27APR00:00	15	1.270	13.730	236.047
538	27APR00:06	98	85.875	12.125	248.172
539	27APR00:12	200	184.891	15.109	263.281
540	27APR00:18	113	93.113	19.887	283.168
541	28APR00:00	15	-1.120	16.120	299.288
542	28APR00:06	104	84.983	19.017	318.305
543	28APR00:12	205	177.940	27.060	345.365
544	28APR00:18	89	64.292	24.708	370.073
545	29APR00:00	12	-6.020	18.020	388.093
546	29APR00:06	68	46.286	21.714	409.807
547	29APR00:12	116	100.339	15.661	425.468
548	29APR00:18	54	34.700	19.300	444.768
549	30APR00:00	10	-6.209	16.209	460.978
550	30APR00:06	30	12.167	17.833	478.811
551	30APR00:12	66	49.524	16.476	495.287
552	30APR00:18	61	40.071	20.929	516.216

*

Now that a baseline model is created, the exploration for alternate models can begin. The review of the harmonic table in Output 41.3.2 shows that all but the last three harmonics are significant, and deleting any of them to form a subset trigonometric seasonal component will lead to a poorer model. The last three harmonics, 12th, 13th and 14th, with periods of 2.333, 2.15 and 2.0, respectively, do appear to be possible choices for deletion. Note that the disturbance variance of the seasonal component is not very insignificant (see Output 41.3.3); therefore the seasonal component is stochastic and the preceding logic, which is based on the final state estimate, provides only a rough guideline.

Output 41.3.2 Harmonic Analysis of the Season: Initial Model

The UCM Procedure

Harmonic Analysis of Trigonometric Seasons
(Based on the Final State)

Name	Season Length	Harmonic	Period	Chi-Square	DF	Pr > ChiSq
Season	28	1	28.00000	234.19	2	<.0001
Season	28	2	14.00000	264.19	2	<.0001
Season	28	3	9.33333	95.65	2	<.0001
Season	28	4	7.00000	105.64	2	<.0001
Season	28	5	5.60000	146.74	2	<.0001
Season	28	6	4.66667	121.93	2	<.0001
Season	28	7	4.00000	4299.12	2	<.0001
Season	28	8	3.50000	150.79	2	<.0001
Season	28	9	3.11111	89.68	2	<.0001
Season	28	10	2.80000	8.95	2	0.0114
Season	28	11	2.54545	6.14	2	0.0464
Season	28	12	2.33333	2.20	2	0.3325
Season	28	13	2.15385	3.40	2	0.1828
Season	28	14	2.00000	2.33	1	0.1272

Output 41.3.3 Parameter Estimates: Initial Model

Final Estimates of the Free Parameters

Component	Parameter	Estimate	Approx Std Error	t Value	Approx Pr > |t|
Irregular	**Error Variance**	92.14591	13.10986	7.03	<.0001
Level	**Error Variance**	44.83595	10.65465	4.21	<.0001
Season	**Error Variance**	0.01250	0.0065153	1.92	0.0551

The following statements fit a subset trigonometric model formed by dropping the last three harmonics by specifying the DROPH= option in the SEASON statement:

```
proc ucm data=callCenter;
   id datetime interval=dthour6;
   model calls;
   irregular;
   level;
   season length=28 type=trig droph=12 13 14;
   estimate back=28;
   forecast back=28 lead=28;
run;
```

The last row of the holdout region prediction analysis table for the preceding model is shown in Output 41.3.4. It shows that the subset trigonometric model has better prediction performance in the holdout region than the full trigonometric model, its SAE = 471.53 compared to the SAE = 516.22 for the full model.

Output 41.3.4 SAE for the Subset Trigonometric Model

Obs	datetime	Actual	Forecast	Error	SAE
552	30APR00:18	61	40.836	20.164	471.534

The following statements illustrate a spline approximation to this seasonal component. In the spline specification the knot placement is quite important, and usually some experimentation is needed. In the following model the knots are placed at the beginning and the middle of each day. Note that the knots at the beginning and end of the season, 1 and 28 in this case, should not be listed in the knot list because knots are always placed there anyway.

```
proc ucm data=callCenter;
   id datetime interval=dthour6;
   model calls;
   irregular;
   level;
   splineseason length=28
      knots=3 5 7 9 11 13 15 17 19 21 23 25 27
      degree=3;
   estimate back=28;
   forecast back=28 lead=28;
run;
```

The spline season model takes about half the time to fit that the baseline model takes. The last row of the holdout region prediction analysis table for this model is shown in Output 41.3.5, which shows that the spline season model performs even better than the previous two models in the holdout region, its SAE = 313.79 compared to SAE = 471.53 for the previous model.

Output 41.3.5 SAE for the Spline Season Model

Obs	datetime	Actual	Forecast	Error	SAE
552	30APR00:18	61	23.350	37.650	313.792

The following statements illustrate yet another way to approximate a long seasonal component. Here a combination of BLOCKSEASON and SEASON statements results in a seasonal component that is a sum of two seasonal patterns: one seasonal pattern is simply a regular season with season length 4 that captures the *within-day* seasonal pattern, and the other seasonal pattern is a block seasonal pattern that remains constant during the day but varies from day to day within a week. Note the use of NLOPTIONS statement to change the optimization technique during the parameter estimation to DBLDOG, which in this case performs better than the default technique, TRUREG.

```
proc ucm data=callCenter;
   id datetime interval=dthour6;
   model calls;
   irregular;
   level;
   season length=4 type=trig;
   blockseason nblocks=7 blocksize=4
      type=trig;
   estimate back=28;
   forecast back=28 lead=28;
   nloptions tech=dbldog;
run;
```

This model also takes about half the time to fit that the baseline model takes. The last row of the holdout region prediction analysis table for this model is shown in Output 41.3.6, which shows that the block season model does slightly better than the baseline model but not as good as the other two models, its SAE = 508.52 compared to the SAE = 516.22 of the baseline model.

Output 41.3.6 SAE for the Block Season Model

Obs	datetime	Actual	Forecast	Error	SAE
552	30APR00:18	61	39.339	21.661	508.522

This example showed a few different ways to model a long seasonal pattern. It showed that parsimonious models for long seasonal patterns can be useful, and in some cases even more effective than the full model. Moreover, for very long seasonal patterns the high memory requirements and long computing times might make full models impractical.

Example 41.4: Modeling Time-Varying Regression Effects

In April 1979 the Albuquerque Police Department began a special enforcement program aimed at reducing the number of DWI (driving while intoxicated) accidents. The program was administered by a squad of police officers, who used breath alcohol testing (BAT) devices and a van that houses a BAT device (Batmobile). These data were collected by the Division of Governmental Research of the University of New Mexico, under a contract with the National Highway Traffic Safety Administration of the U.S. Department of Transportation, to evaluate the Batmobile program. The first 29 observations are for a control period, and the next 23 observations are for the experimental (Batmobile) period. The data, freely available at http://lib.stat.cmu.edu/DASL/Datafiles/batdat.html, consist of two variables: ACC, which represents injuries and fatalities from Wednesday to Saturday nighttime accidents, and FUEL, which represents fuel consumption (millions of gallons) in Albuquerque. The variables are measured quarterly starting from the first quarter of 1972 up to the last quarter of 1984, covering the span of 13 years. The following DATA step statements create the input data set.

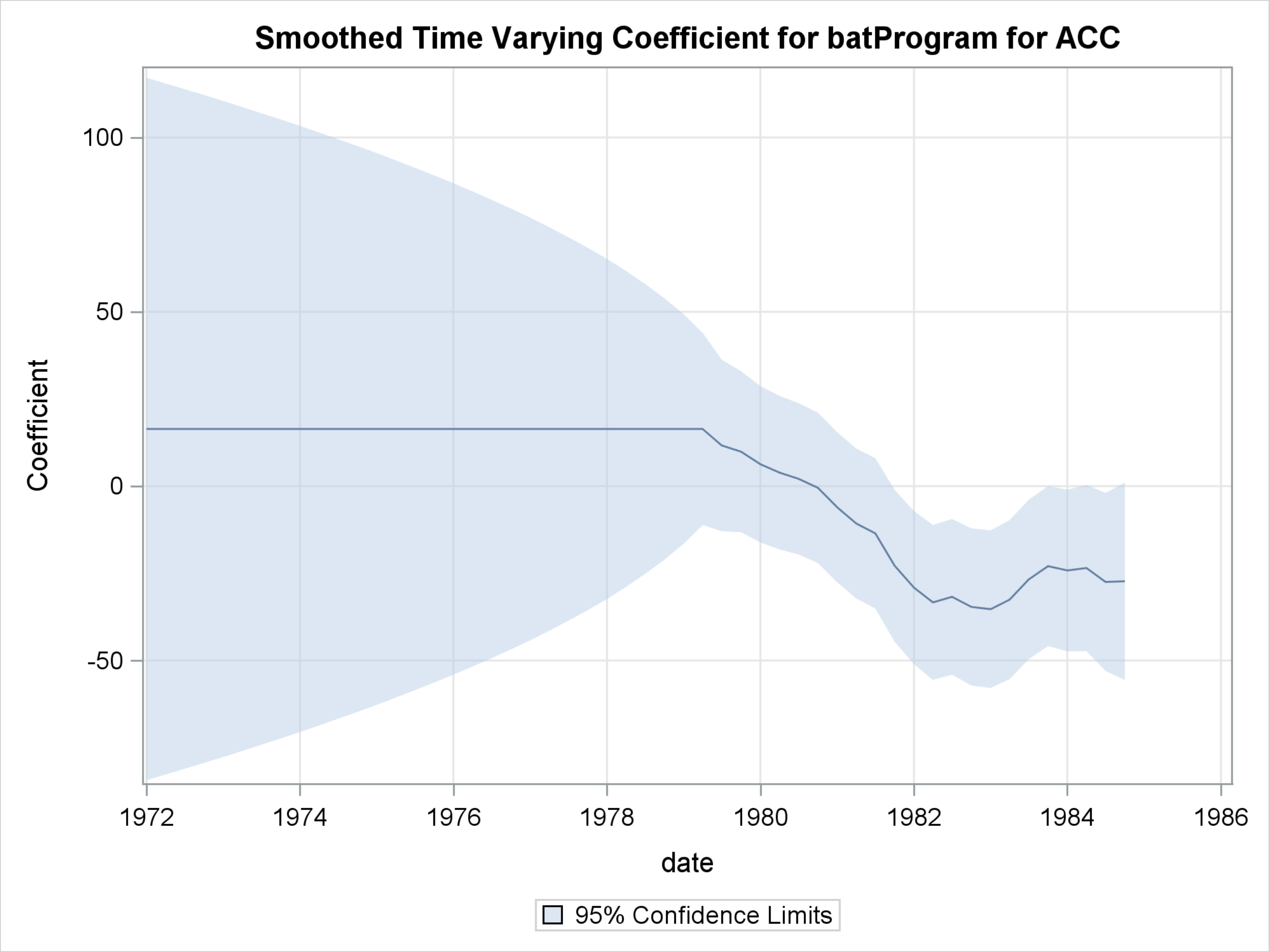
```
data bat;
   input ACC FUEL @@;
   batProgram = 0;
   if _n_ > 29 then batProgram = 1;
   date = INTNX( 'qtr', '1jan1972'd, _n_- 1 );
   format date qtr8.;
datalines;
192    32.592    238    37.250    232    40.032
246    35.852    185    38.226    274    38.711
266    43.139    196    40.434    170    35.898
234    37.111    272    38.944    234    37.717
210    37.861    280    42.524    246    43.965
248    41.976    269    42.918    326    49.789
342    48.454    257    45.056    280    49.385
290    42.524    356    51.224    295    48.562
279    48.167    330    51.362    354    54.646
331    53.398    291    50.584    377    51.320
327    50.810    301    46.272    269    48.664
314    48.122    318    47.483    288    44.732
242    46.143    268    44.129    327    46.258
253    48.230    215    46.459    263    50.686
319    49.681    263    51.029    206    47.236
286    51.717    323    51.824    306    49.380
230    47.961    304    46.039    311    55.683
292    52.263
;
```

There are a number of ways to study these data and the question of the effectiveness of the BAT program. One possibility is to study the *before-after* difference in the injuries and fatalities per million gallons of fuel consumed, by regressing ACC on the FUEL and the dummy variable BATPROGRAM, which is zero before the program began and one while the program is in place. However, it is possible that the effect of the Batmobiles might well be cumulative, because as awareness of the program becomes dispersed, its effectiveness as a deterrent to driving while intoxicated increases. This suggests that the regression coefficient of the BATPROGRAM variable might be *time varying*. The following program fits a model that incorporates these considerations. A seasonal component is included in the model since it is easy to see that the data show strong quarterly seasonality.

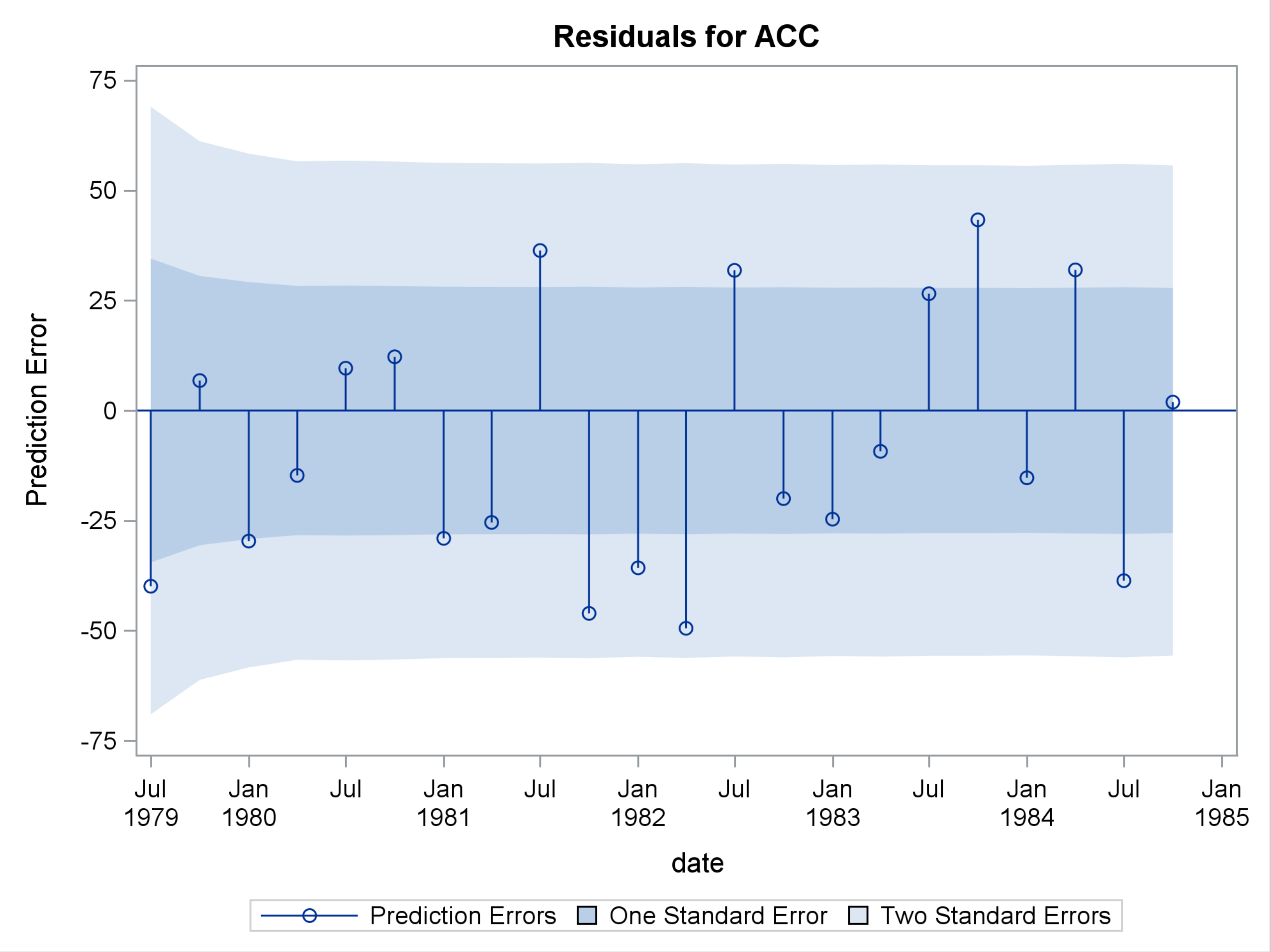
```
proc ucm data=bat;
   model acc = fuel;
   id date interval=qtr;
   irregular;
   level var=0 noest;
   randomreg batProgram / plot=smooth;
   season length=4 var=0 noest plot=smooth;
   estimate plot=(panel residual);
   forecast plot=forecasts lead=0;
run;
```

The model seems to fit the data adequately. No data are withheld for model validation because the series is relatively short. The plot of the time-varying coefficient of BATPROGRAM is shown in Output 41.4.1. As expected, it shows that the effectiveness of the program increases as awareness of the program becomes dispersed. The effectiveness eventually seems to level off. The residual diagnostic plots are shown in Output 41.4.2 and Output 41.4.3, the forecast plot is in Output 41.4.4, the goodness-of-fit statistics are in Output 41.4.5, and the parameter estimates are in Output 41.4.6.

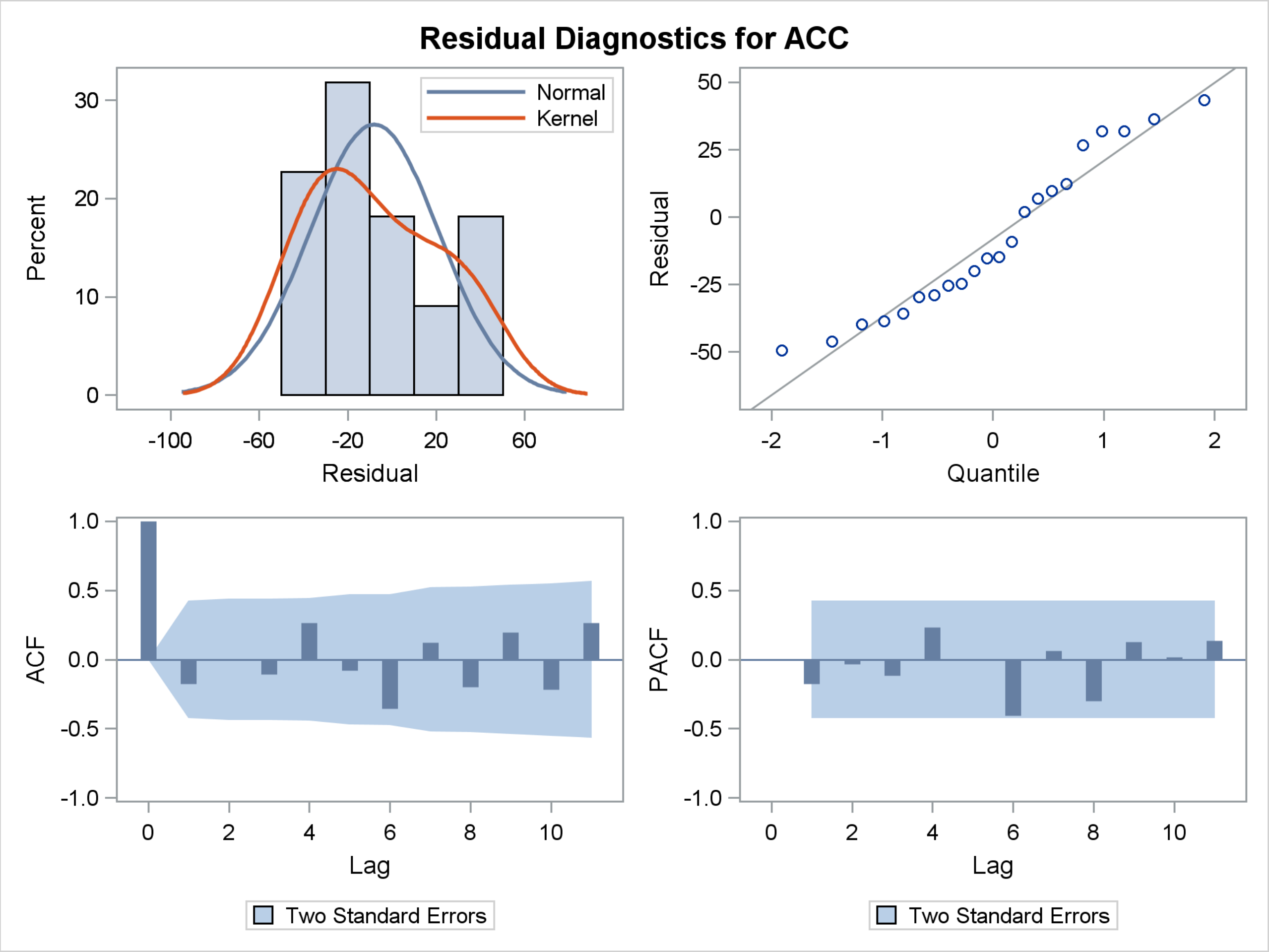
Output 41.4.1 Time-Varying Regression Coefficient of BATPROGRAM



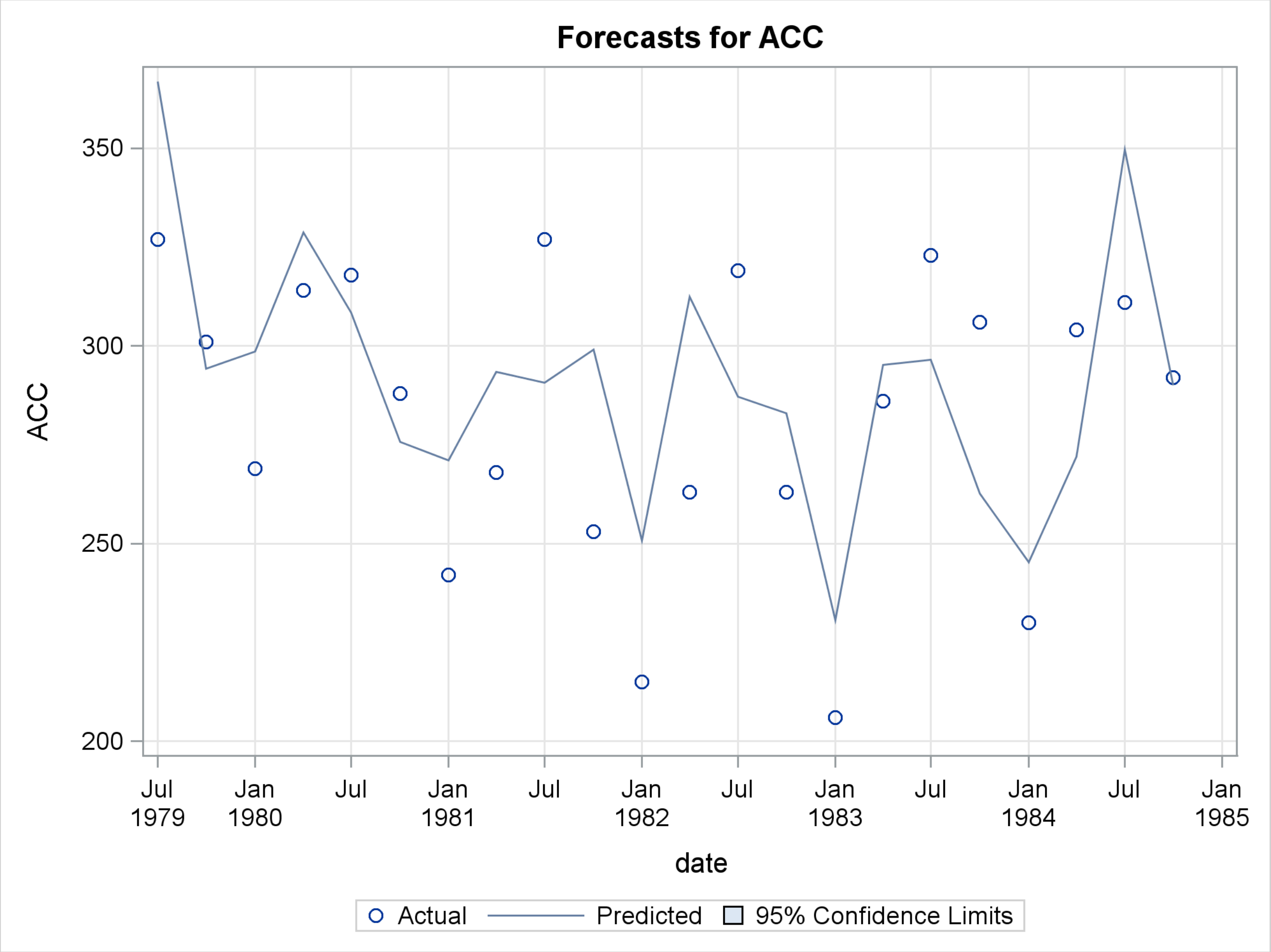
Output 41.4.2 Residuals for the Time-Varying Regression Model



Output 41.4.3 Residual Diagnostics for the Time-Varying Regression Model



Output 41.4.4 One-Step-Ahead Forecasts for the Time-Varying Regression Model



Output 41.4.5 Model Fit for the Time-Varying Regression Model

Fit Statistics Based on Residuals	
Mean Squared Error	866.75562
Root Mean Squared Error	29.44071
Mean Absolute Percentage Error	9.50326
Maximum Percent Error	14.15368
R-Square	0.32646
Adjusted R-Square	0.29278
Random Walk R-Square	0.63010
Amemiya's Adjusted R-Square	0.19175

Number of non-missing residuals used for computing the fit statistics = 22

Output 41.4.6 Parameter Estimates for the Time-Varying Regression Model

Final Estimates of the Free Parameters

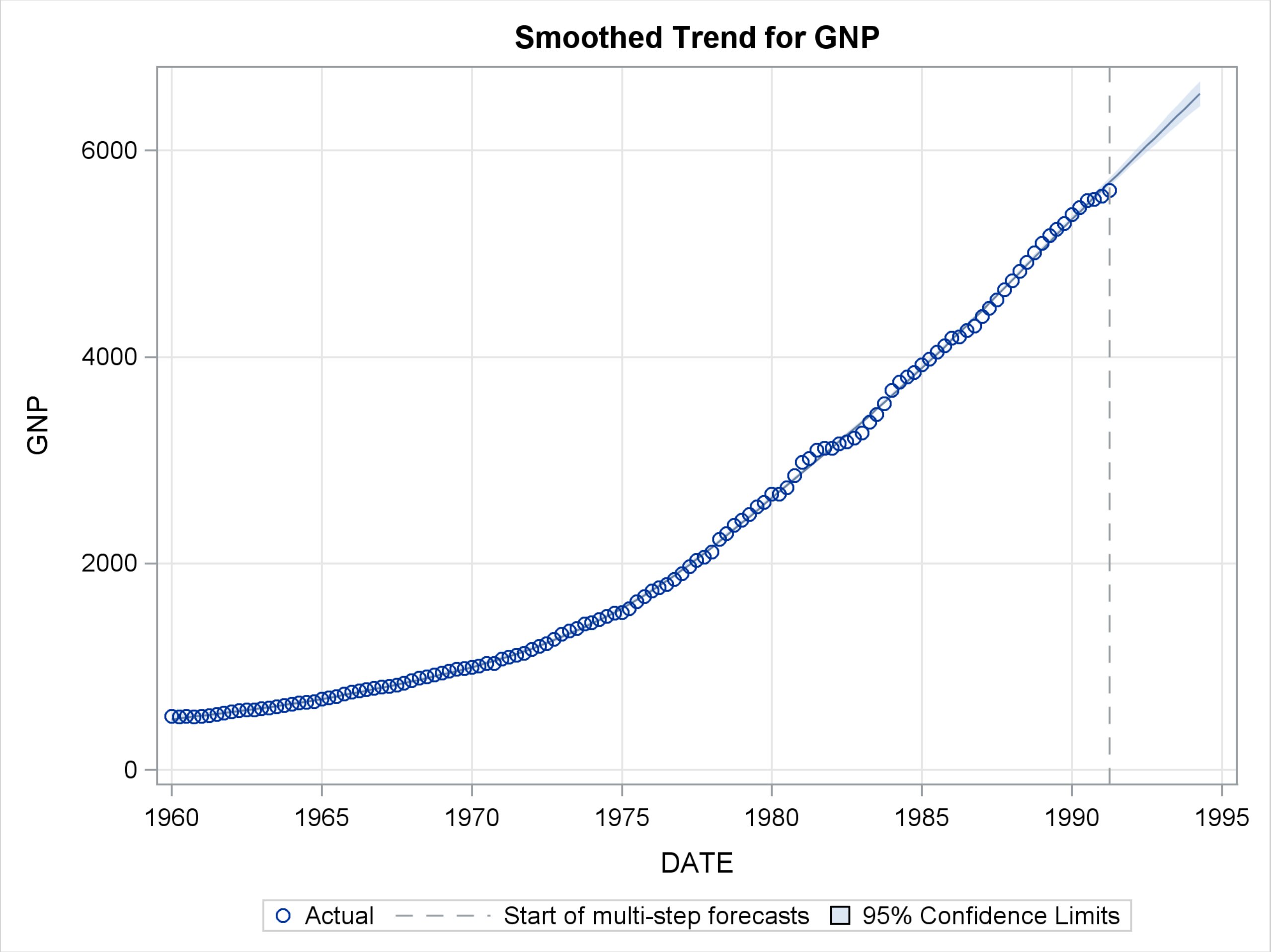
			Approx		Approx
Component	**Parameter**	**Estimate**	**Std Error**	**t Value**	**Pr > |t|**
Irregular	**Error Variance**	480.92258	109.21980	4.40	<.0001
FUEL	**Coefficient**	6.23279	0.67533	9.23	<.0001
batProgram	**Error Variance**	84.22334	79.88166	1.05	0.2917

Example 41.5: Trend Removal Using the Hodrick-Prescott Filter

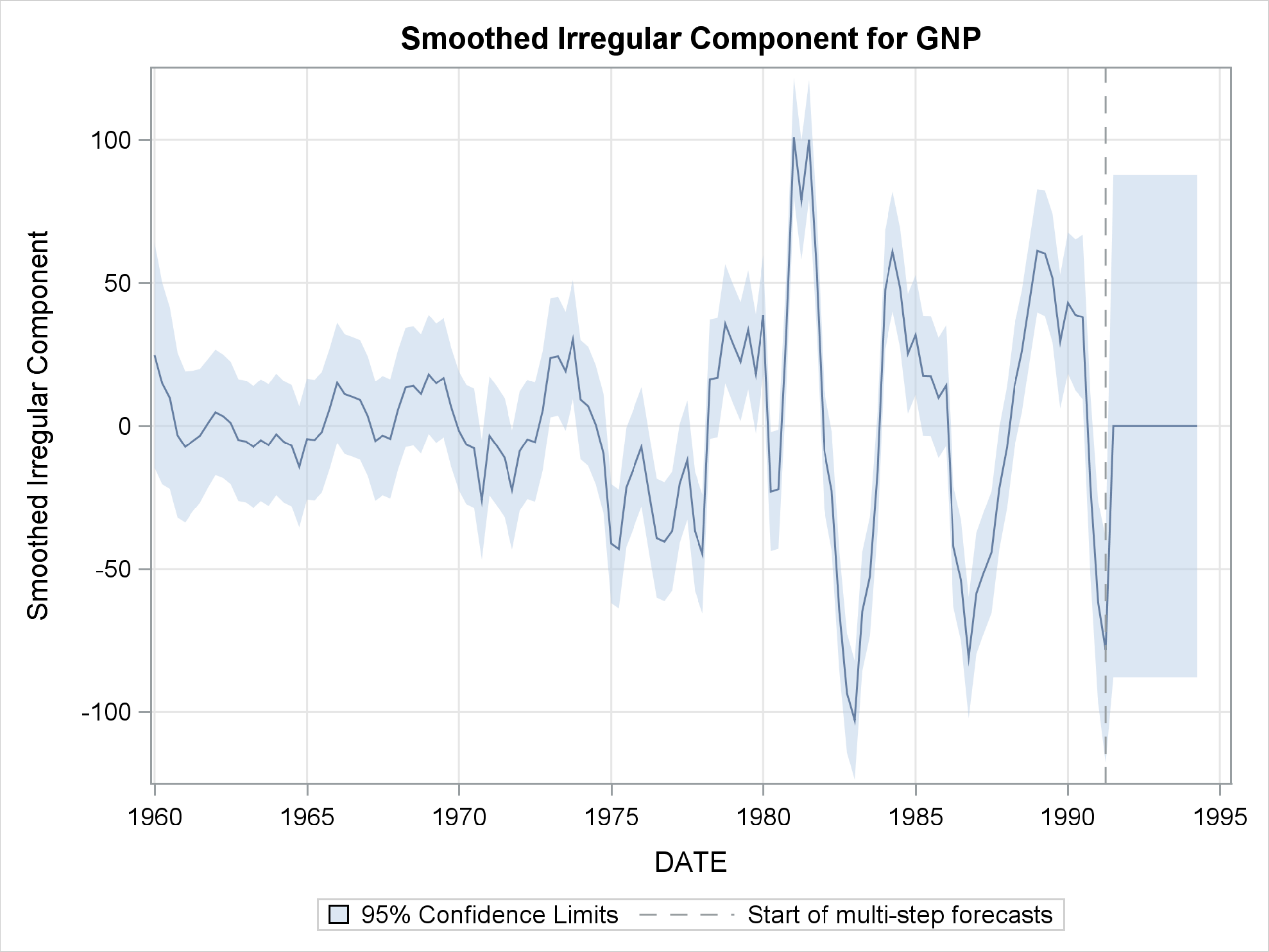
Hodrick-Prescott filter (Hodrick and Prescott 1997) is a popular tool in macroeconomics for fitting smooth trend to time series. It is well known that the trend computation according to this filter is equivalent to fitting the local linear trend plus irregular model with the level disturbance variance restricted to zero and the slope disturbance variance restricted to be a suitable multiple of the irregular component variance. The multiple used depends on the frequency of the series; for example, for quarterly series the commonly recommended multiple is $1/1600 = 0.000625$. For other intervals there is no consensus, but a frequently suggested value for monthly series is $1/14400$ and the value for an annual series can range from $1/400 = 0.0025$ to $1/7 = 0.15$. The data set considered in this example consists of quarterly GNP values for the United States from 1960 to 1991. In the UCM procedure statements that follow, the presence of the PROFILE option in the ESTIMATE statement implies that the restriction that the disturbance variance of the slope component be fixed at 0.000625 is interpreted differently: it implies that the disturbance variance of the slope component be restricted to be 0.000625 *times* the estimated irregular component variance, as needed for the Hodrick-Prescott filter. The plot of the fitted trend is shown in Output 41.5.1, and the plot of the smoothed irregular component, which corresponds to the detrended series, is given in Output 41.5.2. The detrended series can be further analyzed for business cycles.

```
proc ucm data=sashelp.gnp;
   id date interval=qtr;
   model gnp;
   irregular plot=smooth;
   level var=0 noest plot=smooth;
   slope var=0.000625 noest;
   estimate PROFILE;
   forecast plot=(decomp);
run;
```

Output 41.5.1 Smoothed Trend for the GNP Series as per the Hodrick-Prescott Filter



Output 41.5.2 Detrended GNP Series



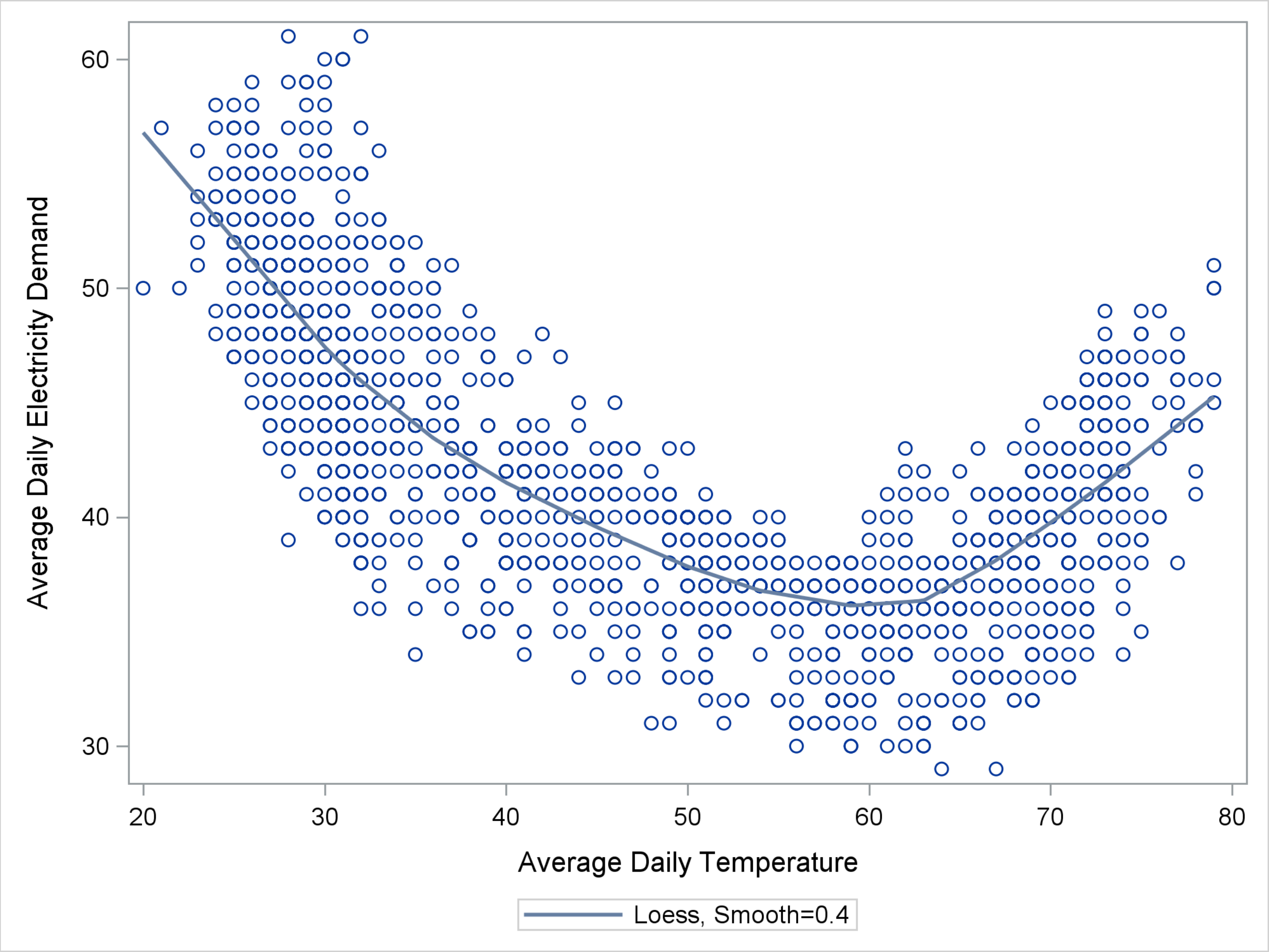
Example 41.6: Using Splines to Incorporate Nonlinear Effects

The data in this example are created to mirror the electricity demand and temperature data recorded at a utility company in the midwest region of the United States. The data set (not shown), utility, has three variables: load, temp, and date. The load column contains the daily electricity demand, the temp column has the average daily temperature readings, and the date column records the observation date.

The following statements produce a plot, shown in Output 41.6.1, of electricity load versus temperature. Clearly the relationship is smooth but nonlinear: the load generally increases when the temperatures are away from the comfortable sixties.

```
proc sgplot data=utility;
    loess x=temp y=load / smooth=0.4;
run;
```

Output 41.6.1 Load versus Temperature Plot



The time series plot of the load (not shown) also shows that, apart from a day-of-the-week seasonal effect, there are no additional easily identifiable patterns in the series. The series has no apparent upward or downward trend. The following statements fit a UCM to the series that takes into account these observations. The particular choice of the model is a result of a little modeling exercise that compared a small number of competing models. The chosen model is adequate but by no means the best possible. The temperature effect is modeled by a deterministic three-degree spline with knots at 30, 40, 50, 60, and 75. The knot locations and the degree were chosen by visual inspection of the plot (Output 41.6.1). An autoreg component is used in place of the simple irregular component, which improved the residual analysis. The last 60 days of data are withheld for out-of-sample forecast evaluation (note the BACK= option in both the ESTIMATE and FORECAST statements). The OUTLIER statement is used to increase the number of outliers reported to 10. Since no CHECKBREAK option is used in the LEVEL statement, only the additive outliers are searched. In this example the use of the EXTRADIFFUSE= option in the ESTIMATE and FORECAST statements is useful for discarding some early one-step-ahead forecasts and residuals with large variance.

```
proc ucm data=utility;
   id date interval=day;
   model load;
   autoreg;
   level plot=smooth;
```

```
      splinereg temp knots=30 40 50 65 75 degree=3
         variance=0 noest;
      season length=7 var=0 noest;
      estimate plot=panel back=60
         extradiffuse=50;
      outlier maxnum=10;
      forecast back=60 lead=60
         extradiffuse=50;
   run;
```

The parameter estimates are given in Output 41.6.2, and the residual goodness-of-fit statistics are shown in Output 41.6.3. The residual diagnostic plots are shown in Output 41.6.4. The ACF and PACF plots appear satisfactory, but the normality plots, particularly the Q-Q plot, show possible violations. It appears that, at least in part, this nonNormal behavior of the residuals might be attributable to the outliers in the series. The outlier summary table, Output 41.6.5, shows the most likely outlying observations. Notice that most of these outliers are holidays, like July 4th, when the electricity load is lower than usual for that day of the week.

Output 41.6.2 Electricity Load: Parameter Estimates

The UCM Procedure

Final Estimates of the Free Parameters

Component	Parameter	Estimate	Approx Std Error	t Value	Approx Pr > |t|
Level	Error Variance	0.21185	0.05025	4.22	<.0001
AutoReg	Damping Factor	0.57522	0.03466	16.60	<.0001
AutoReg	Error Variance	2.21057	0.20478	10.79	<.0001
temp	Spline Coefficient_1	4.72502	1.93997	2.44	0.0149
temp	Spline Coefficient_2	2.19116	1.71243	1.28	0.2007
temp	Spline Coefficient_3	-7.14492	1.56805	-4.56	<.0001
temp	Spline Coefficient_4	-11.39950	1.45098	-7.86	<.0001
temp	Spline Coefficient_5	-16.38055	1.36977	-11.96	<.0001
temp	Spline Coefficient_6	-18.76075	1.28898	-14.55	<.0001
temp	Spline Coefficient_7	-8.04628	1.09017	-7.38	<.0001
temp	Spline Coefficient_8	-2.30525	1.25102	-1.84	0.0654

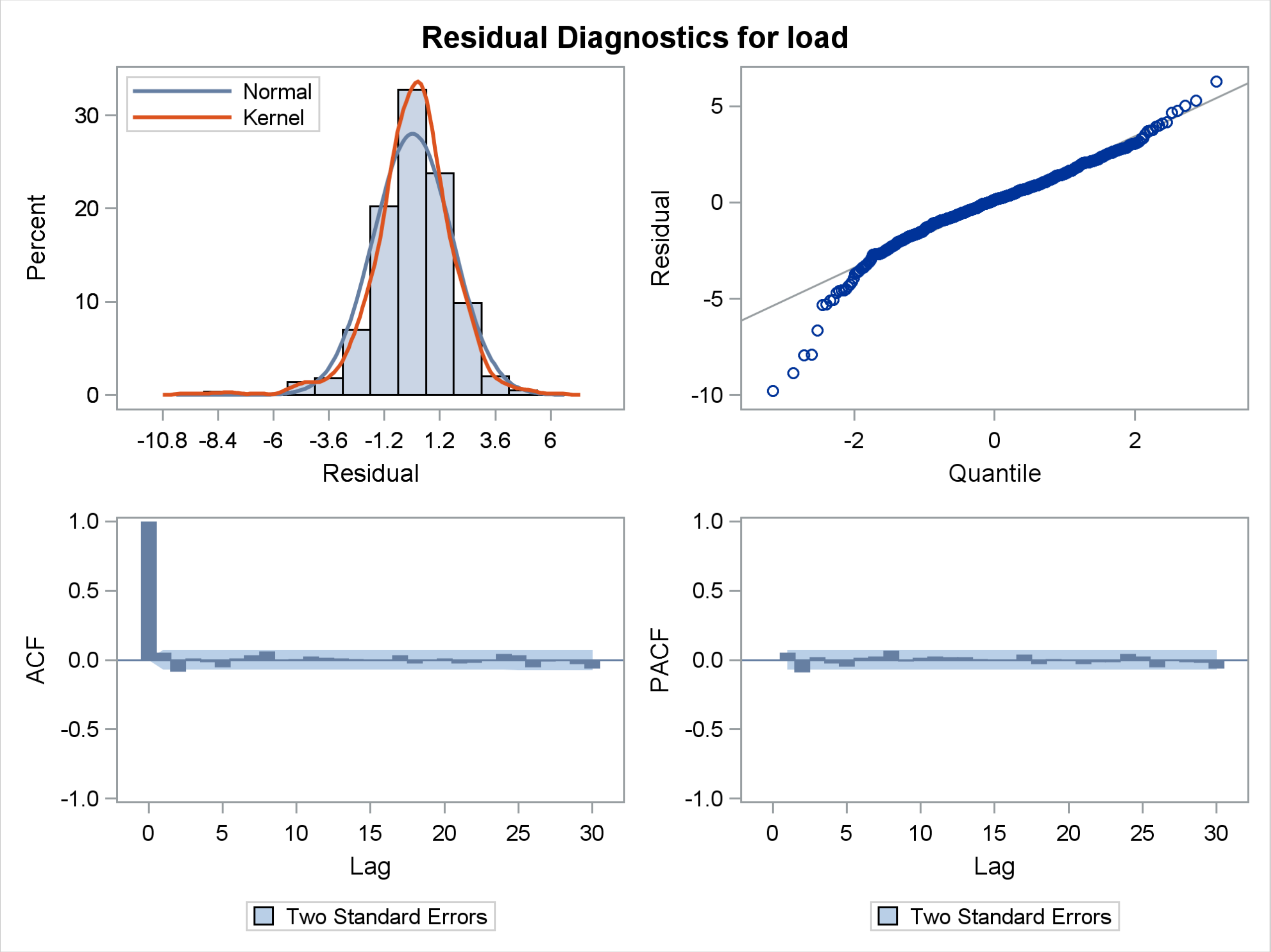
Output 41.6.3 Electricity Load: goodness-of-fit

Fit Statistics Based on Residuals

Mean Squared Error	2.90945
Root Mean Squared Error	1.70571
Mean Absolute Percentage Error	2.92586
Maximum Percent Error	14.96281
R-Square	0.92739
Adjusted R-Square	0.92721
Random Walk R-Square	0.69618
Amemiya's Adjusted R-Square	0.92684

Number of non-missing residuals used for computing the fit statistics = 791

Output 41.6.4 Electricity Load: Residual Diagnostics

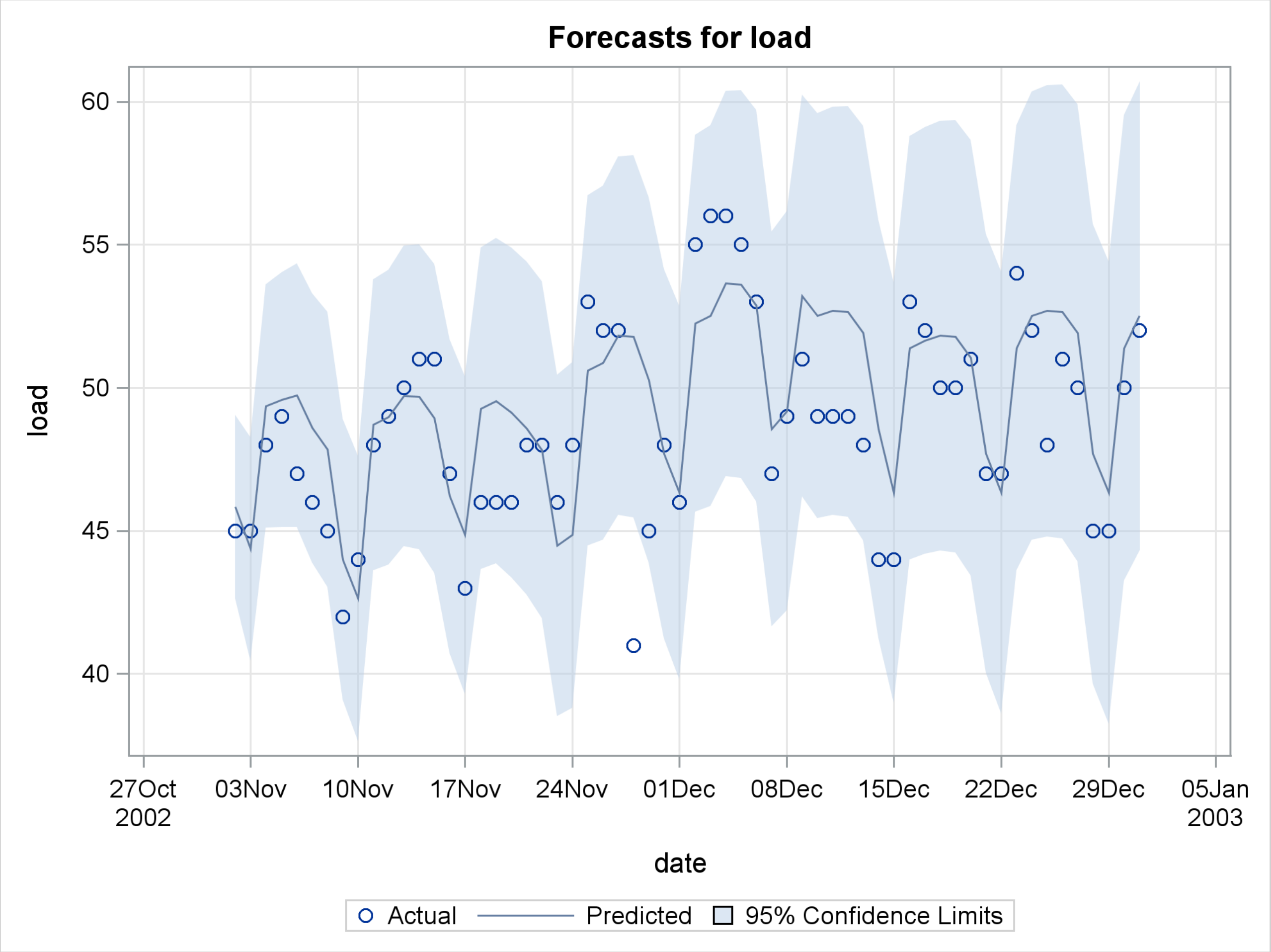


Output 41.6.5 Additive Outliers in the Electricity Load Series

Obs	Time	Estimate	StdErr	ChiSq	DF	ProbChiSq
1281	04JUL2002	-7.99908	1.3417486	35.54	1	<.0001
916	04JUL2001	-6.55778	1.338431	24.01	1	<.0001
329	25NOV1999	-5.85047	1.3379735	19.12	1	<.0001
977	03SEP2001	-5.67254	1.3389138	17.95	1	<.0001
1341	02SEP2002	-5.49631	1.337843	16.88	1	<.0001
693	23NOV2000	-5.27968	1.3374368	15.58	1	<.0001
915	03JUL2001	5.06557	1.3375273	14.34	1	0.0002
1057	22NOV2001	-5.01550	1.3386184	14.04	1	0.0002
551	04JUL2000	-4.89965	1.3381557	13.41	1	0.0003
879	28MAY2001	-4.76135	1.3375349	12.67	1	0.0004

The plot of the load forecasts for the withheld data is shown in Output 41.6.6.

Output 41.6.6 Electricity Load: Forecast Evaluation of the Withheld Data



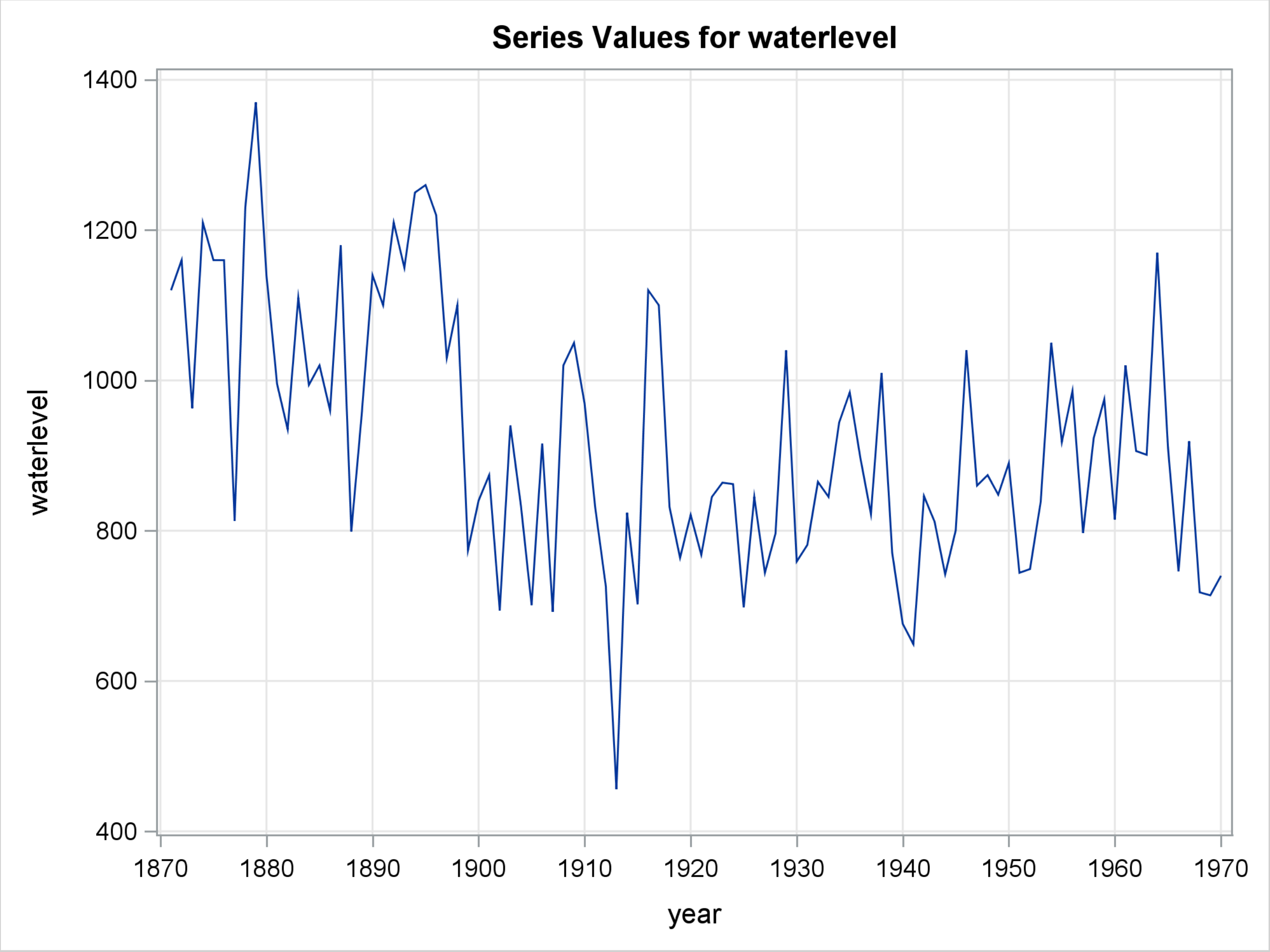
Example 41.7: Detection of Level Shift

The series in this example consists of the yearly water level readings of the Nile River recorded at Aswan, Egypt (Cobb 1978; De Jong and Penzer 1998). The readings are from the years 1871 to 1970. The series does not show any apparent trend or any other distinctive patterns; however, there is a shift in the water level starting at the year 1899. This shift could be attributed to the start of construction of a dam near Aswan in that year. A time series plot of this series is given in Output 41.7.1. The following DATA step statements create the input data set.

```
data nile;
   input waterlevel @@;
   year = intnx( 'year', '1jan1871'd, _n_-1 );
   format year year4.;
datalines;
   1120  1160  963  1210  1160  1160  813  1230   1370  1140
   995   935   1110 994   1020  960   1180 799    958   1140
   1100  1210  1150 1250  1260  1220  1030 1100   774   840
   874   694   940  833   701   916   692  1020   1050  969
   831   726   456  824   702   1120  1100 832    764   821
   768   845   864  862   698   845   744  796    1040  759
   781   865   845  944   984   897   822  1010   771   676
   649   846   812  742   801   1040  860  874    848   890
   744   749   838  1050  918   986   797  923    975   815
   1020  906   901  1170  912   746   919  718    714   740
;

proc timeseries data=nile plot=series;
   id year interval=year;
   var waterlevel;
run;
```

Output 41.7.1 Nile Water Level

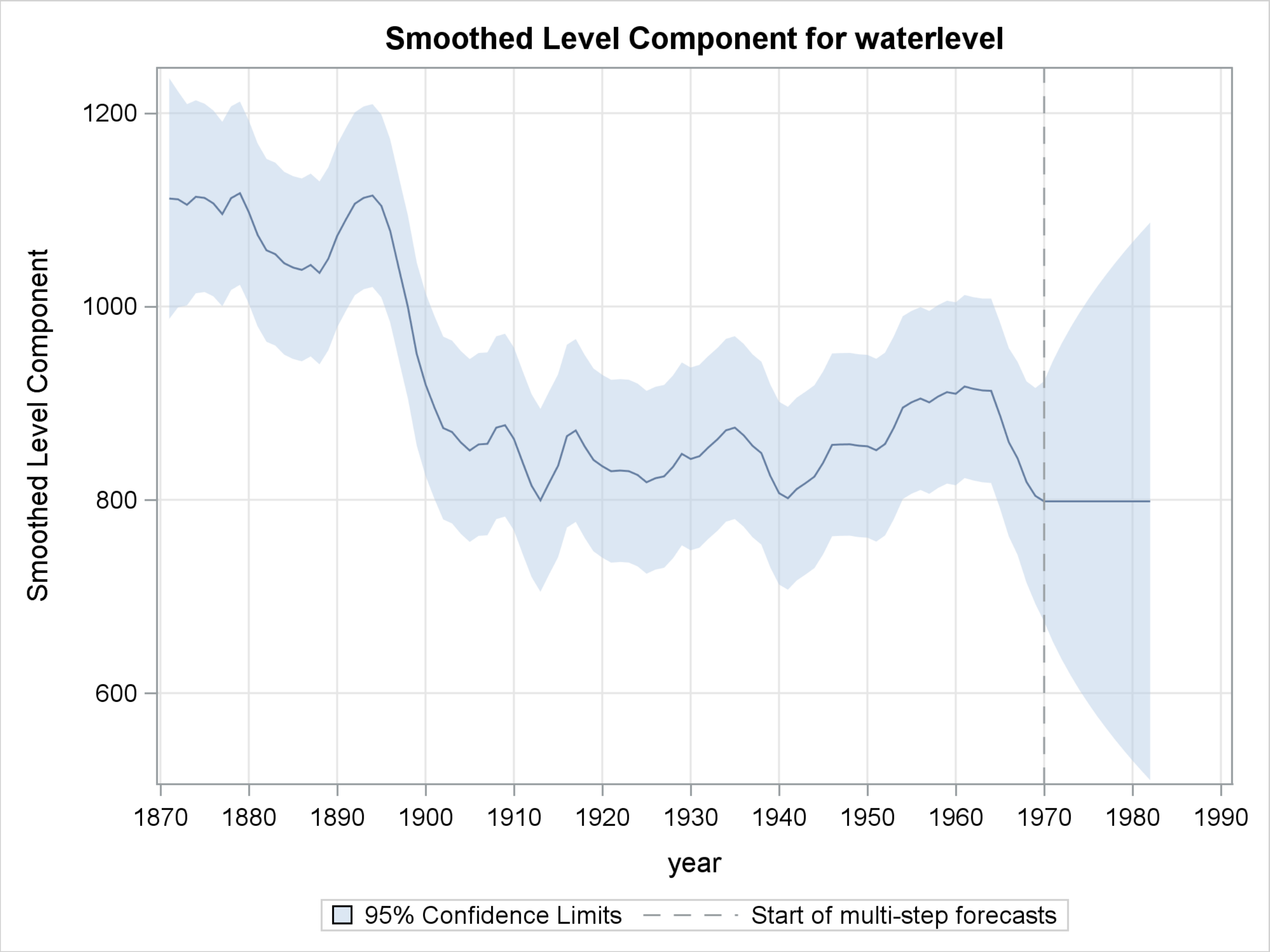


In this situation it is known that a shift in the water level occurred within the span of the series, and its effect can be easily taken into account by including an appropriate indicator variable as a regressor. However, in many situation such prior information is not available, and it is useful to detect such a shift in a data analytic fashion. You can check for breaks in the level by using the CHECKBREAK option in the LEVEL statement. The following statements fit a simple locally constant level plus error model to the series:

```
proc ucm data=nile;
   id year interval=year;
   model waterlevel;
   irregular;
   level plot=smooth checkbreak;
   estimate;
   forecast plot=decomp;
run;
```

The plot in Output 41.7.2 shows a noticeable drop in the smoothed water level around 1899.

Output 41.7.2 Smoothed Trend without the Shift of 1899



The "Outlier Summary" table in Output 41.7.3 shows the most likely types of breaks and their locations within the series span. The shift of 1899 is easily detected.

Output 41.7.3 Detection of Structural Breaks in the Nile River Level

Outlier Summary

Obs	year	Break Type	Estimate	Standard Error	Chi-Square	DF	Pr > ChiSq
29	**1899**	**Level**	-315.73791	97.639753	10.46	1	0.0012

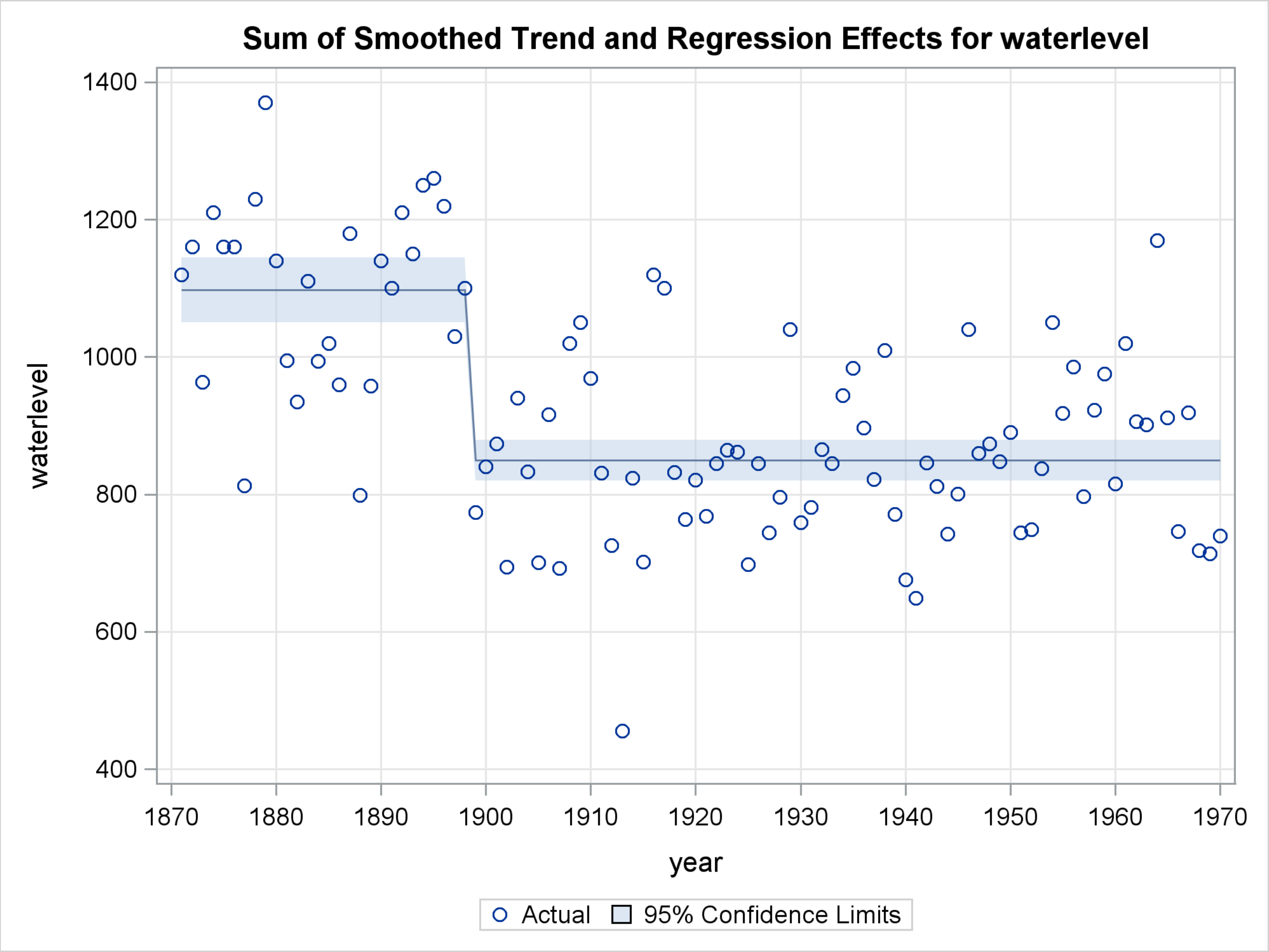
The following statements specify a UCM that models the level of the river as a locally constant series with a shift in the year 1899, represented by a dummy regressor (SHIFT1899):

```
data nile;
   set nile;
   shift1899 = ( year >= '1jan1899'd );
run;

proc ucm data=nile;
   id year interval=year;
   model waterlevel = shift1899;
   irregular;
   level;
   estimate;
   forecast plot=decomp;
run;
```

The plot in Output 41.7.4 shows the smoothed trend, including the correction due to the shift in the year 1899. Notice the simplicity in the shape of the smoothed curve after the incorporation of the shift information.

Output 41.7.4 Smoothed Trend plus Shift of 1899



Example 41.8: ARIMA Modeling

This example shows how you can use the UCM procedure for ARIMA modeling. The parameter estimates and predictions for ARIMA models obtained by using PROC UCM will be close to those obtained by using PROC ARIMA (in the presence of the ML option in its ESTIMATE statement) if the model is stationary or if the model is nonstationary and there are no missing values in the data. For more information about the ARIMA procedure, see Chapter 8, "The ARIMA Procedure." However, if there are missing values in the data and the model is nonstationary, then the UCM and ARIMA procedures can produce significantly different parameter estimates and predictions. An article by Kohn and Ansley (1986) suggests a statistically sound method of estimation, prediction, and interpolation for nonstationary ARIMA models with missing data. This method is based on an algorithm that is equivalent to the Kalman filtering and smoothing algorithm used in the UCM procedure. The results of an illustrative example in their article are reproduced here using the UCM procedure. In this example an ARIMA(0,1,1)×(0,1,1)$_{12}$ model is applied to the logarithm of the air series in the sashelp.air data set. Four different missing value patterns are considered to highlight different aspects of the problem:

- *Data1*. The full data set of 144 observations.
- *Data2*. The set of 78 observations that omit January through November in each of the last 6 years.
- *Data3*. The data set with the 5 observations July 1949, June, July, and August 1957, and July 1960 missing.
- *Data4*. The data set with all July observations missing and June and August 1957 also missing.

The following DATA steps create these data sets:

```
data Data1;
   set sashelp.air;
   logair = log(air);
run;

data Data2;
   set data1;
   if year(date) >= 1955 and month(date) < 12 then logair = .;
run;

data Data3;
   set data1;
   if (year(date) = 1949 and month(date) = 7) then logair = .;
   if ( year(date) = 1957 and
       (month(date) = 6 or month(date) = 7 or month(date) = 8))
       then logair = .;
   if (year(date) = 1960 and month(date) = 7) then logair = .;
run;

data Data4;
   set data1;
   if month(date) = 7 then logair = .;
   if year(date) = 1957 and (month(date) = 6 or month(date) = 8)
      then logair = .;
run;
```

The following statements specify the ARIMA$(0,1,1) \times (0,1,1)_{12}$ model for the logair series in the first data set (Data1):

```
proc ucm data=Data1;
   id date interval=month;
   model logair;
   irregular q=1 sq=1 s=12;
   deplag lags=(1)(12) phi=1 1 noest;
   estimate outest=est1;
   forecast outfor=for1;
run;
```

Note that the moving average part of the model is specified by using the Q=, SQ=, and S= options in the IRREGULAR statement and the differencing operator, $(1-B)(1-B^{12})$, is specified by using the DEPLAG statement. The model does not contain an intercept term; therefore no LEVEL statement is needed. The parameter estimates are saved in a data set EST1 by using the OUTEST= option in the ESTIMATE statement and the forecasts and the component estimates are saved in a data set FOR1 by using the OUTFOR= option in the FORECAST statement. The same analysis is performed on the other three data sets, but is not shown here.

Output 41.8.1 resembles Table 1 in Kohn and Ansley (1986). This table is generated by merging the parameter estimates from the four analyses. Only the moving average parameter estimates and their standard errors are reported. The columns EST1 and STD1 correspond to the estimates for Data1. The parameter estimates and their standard errors for other three data sets are similarly named. Note that the parameter estimates closely match the parameter estimates in the article. However, their standard errors differ slightly. This difference could be the result of different ways of computing the Hessian at the optimum. The white noise error variance estimates are not reported here, but they agree quite closely with those in the article.

Output 41.8.1 Data Sets 1–4: Parameter Estimates and Standard Errors

PARAMETER	est1	std1	est2	std2	est3	std3	est4	std4
MA_1	0.402	0.090	0.457	0.121	0.408	0.092	0.431	0.091
SMA_1	0.557	0.073	0.758	0.236	0.566	0.075	0.573	0.074

Output 41.8.2 resembles Table 2 in Kohn and Ansley (1986). It contains forecasts and their standard errors for the four data sets. The numbers are very close to those in the article.

Output 41.8.2 Data Sets 1–4: Forecasts and Standard Errors

DATE	for1	std1	for2	std2	for3	std3	for4	std4
JAN61	6.110	0.037	6.084	0.052	6.110	0.037	6.111	0.037
FEB61	6.054	0.043	6.091	0.058	6.054	0.043	6.055	0.043
MAR61	6.172	0.048	6.247	0.063	6.173	0.048	6.174	0.048
APR61	6.199	0.053	6.205	0.068	6.199	0.053	6.200	0.052
MAY61	6.233	0.057	6.199	0.072	6.232	0.058	6.233	0.056
JUN61	6.369	0.061	6.308	0.076	6.367	0.062	6.368	0.060
JUL61	6.507	0.065	6.409	0.079	6.497	0.067	.	.
AUG61	6.503	0.069	6.414	0.082	6.503	0.069	6.503	0.067
SEP61	6.325	0.072	6.299	0.085	6.325	0.072	6.326	0.071
OCT61	6.209	0.075	6.174	0.087	6.209	0.076	6.209	0.074
NOV61	6.063	0.079	6.043	0.089	6.064	0.079	6.064	0.077
DEC61	6.168	0.082	6.174	0.086	6.168	0.082	6.169	0.080

Output 41.8.3 is based on Data2. It resembles Table 3 in Kohn and Ansley (1986). The columns S_SERIES and VS_SERIES in the OUTFOR= data set contain the interpolated values of logair and their variances. The estimate column in Output 41.8.3 reports interpolated values (which are the same as S_SERIES), and the std column reports their standard errors (which are computed as square root of VS_SERIES) for January–November 1957. The actual logair values for these months, which are missing in Data2, are also provided for comparison. The numbers are very close to those in the article.

Output 41.8.3 Data Set 2: Interpolated Values and Standard Errors

DATE	logair	estimate	std
JAN57	5.753	5.733	0.045
FEB57	5.707	5.738	0.049
MAR57	5.875	5.893	0.052
APR57	5.852	5.850	0.054
MAY57	5.872	5.843	0.055
JUN57	6.045	5.951	0.055
JUL57	6.142	6.051	0.055
AUG57	6.146	6.055	0.054
SEP57	6.001	5.938	0.052
OCT57	5.849	5.812	0.049
NOV57	5.720	5.680	0.045

Output 41.8.4 resembles Table 4 in Kohn and Ansley (1986). These numbers are based on Data3, and they also are very close to those in the article.

Output 41.8.4 Data Set 3: Interpolated Values and Standard Errors

DATE	logair	estimate	std
JUL49	4.997	5.013	0.031
JUN57	6.045	6.024	0.030
JUL57	6.142	6.147	0.031
AUG57	6.146	6.148	0.030
JUL60	6.433	6.409	0.031

Output 41.8.5 resembles Table 5 in Kohn and Ansley (1986). As before, the numbers are very close to those in the article.

Output 41.8.5 Data Set 4: Interpolated Values and Standard Errors

DATE	logair	estimate	std
JUN57	6.045	6.023	0.030
AUG57	6.146	6.147	0.030

The similarity between the outputs in this example and the results shown in Kohn and Ansley (1986) demonstrate that PROC UCM can be effectively used for nonstationary ARIMA models with missing data.

References

Akaike, H. (1974). “A New Look at the Statistical Model Identification.” *IEEE Transactions on Automatic Control* AC-19:716–723.

Anderson, T. W. (1971). *The Statistical Analysis of Time Series*. New York: John Wiley & Sons.

Bloomfield, P. (2000). *Fourier Analysis of Time Series*. 2nd ed. New York: John Wiley & Sons.

Box, G. E. P., and Jenkins, G. M. (1976). *Time Series Analysis: Forecasting and Control*. Rev. ed. San Francisco: Holden-Day.

Bozdogan, H. (1987). “Model Selection and Akaike’s Information Criterion (AIC): The General Theory and Its Analytical Extensions.” *Psychometrika* 52:345–370.

Brockwell, P. J., and Davis, R. A. (1991). *Time Series: Theory and Methods*. 2nd ed. New York: Springer-Verlag.

Burnham, K. P., and Anderson, D. R. (1998). *Model Selection and Inference: A Practical Information-Theoretic Approach*. New York: Springer-Verlag.

Cobb, G. W. (1978). “The Problem of the Nile: Conditional Solution to a Change Point Problem.” *Biometrika* 65:243–251.

De Jong, P., and Chu-Chun-Lin, S. (2003). “Smoothing with an Unknown Initial Condition.” *Journal of Time Series Analysis* 24:141–148.

De Jong, P., and Penzer, J. (1998). “Diagnosing Shocks in Time Series.” *Journal of the American Statistical Association* 93:796–806.

Durbin, J., and Koopman, S. J. (2001). *Time Series Analysis by State Space Methods*. Oxford: Oxford University Press.

Hannan, E. J., and Quinn, B. G. (1979). “The Determination of the Order of an Autoregression.” *Journal of the Royal Statistical Society, Series B* 41:190–195.

Harvey, A. C. (1989). *Forecasting, Structural Time Series Models, and the Kalman Filter*. Cambridge: Cambridge University Press.

Harvey, A. C. (2001). “Testing in Unobserved Components Models.” *Journal of Forecasting* 20:1–19.

Hodrick, R. J., and Prescott, E. C. (1997). “Postwar U.S. Business Cycles: An Empirical Investigation.” *Journal of Money, Credit, and Banking* 29:1–16.

Hurvich, C. M., and Tsai, C.-L. (1989). “Regression and Time Series Model Selection in Small Samples.” *Biometrika* 76:297–307.

Jones, R. H. (1980). “Maximum Likelihood Fitting of ARMA Models to Time Series with Missing Observations.” *Technometrics* 22:389–396.

Kohn, R., and Ansley, C. F. (1986). “Estimation, Prediction, and Interpolation for ARIMA Models with Missing Data.” *Journal of the American Statistical Association* 81:751–761.

Rodriguez, A., and Ruiz, E. (2010). "Bootstrap Prediction Mean Squared Errors of Unobserved States Based on the Kalman Filter with the Estimated Parameters." Working Paper 10-03 (01), Statistics and Econometric Series, Departamento de Estadística, Universidad Carlos de Madrid.

Schwarz, G. (1978). "Estimating the Dimension of a Model." *Annals of Statistics* 6:461–464.

West, M., and Harrison, J. (1999). *Bayesian Forecasting and Dynamic Models*. 2nd ed. New York: Springer-Verlag.

Subject Index

Syntax Index