Chapter 30  
Forecasting Process Details

Chapter Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORECASTING PROCESS SUMMARY</td>
<td>1477</td>
</tr>
<tr>
<td>Forecast Combination Models</td>
<td>1479</td>
</tr>
<tr>
<td>External or User Supplied Forecasts</td>
<td>1480</td>
</tr>
<tr>
<td>Adjustments</td>
<td>1480</td>
</tr>
<tr>
<td>Series Transformations</td>
<td>1481</td>
</tr>
<tr>
<td>SMOOTHING MODELS</td>
<td>1482</td>
</tr>
<tr>
<td>Smoothing Model Calculations</td>
<td>1483</td>
</tr>
<tr>
<td>Missing Values</td>
<td>1484</td>
</tr>
<tr>
<td>Predictions and Prediction Errors</td>
<td>1484</td>
</tr>
<tr>
<td>Smoothing Weights</td>
<td>1485</td>
</tr>
<tr>
<td>Equations for the Smoothing Models</td>
<td>1486</td>
</tr>
<tr>
<td>ARIMA MODELS</td>
<td>1494</td>
</tr>
<tr>
<td>Notation for ARIMA Models</td>
<td>1494</td>
</tr>
<tr>
<td>PREDICTOR SERIES</td>
<td>1498</td>
</tr>
<tr>
<td>Time Trend Curves</td>
<td>1498</td>
</tr>
<tr>
<td>Intervention Effects</td>
<td>1500</td>
</tr>
<tr>
<td>Seasonal Dummy Inputs</td>
<td>1501</td>
</tr>
<tr>
<td>SERIES DIAGNOSTIC TESTS</td>
<td>1502</td>
</tr>
<tr>
<td>STATISTICS OF FIT</td>
<td>1502</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>1505</td>
</tr>
</tbody>
</table>
Part 3. General Information
Chapter 30
Forecasting Process Details

This chapter provides computational details on several aspects of the Time Series Forecasting system.

Forecasting Process Summary

This section summarizes the forecasting process.

Parameter Estimation

The parameter estimation process for ARIMA and smoothing models is described graphically in Figure 30.1.

Figure 30.1. Model Fitting Flow Diagram

The specification of smoothing and ARIMA models is described in Chapter 25, “Specifying Forecasting Models,”. Computational details for these kinds of models are provided in the following sections "Smoothing Models" and "ARIMA Models." The results of the parameter estimation process are displayed in the Parameter Estimates table of the Model Viewer window along with the estimate of the model variance and the final smoothing state.

Model Evaluation

The model evaluation process is described graphically in Figure 30.2.
Model evaluation is based on the one-step-ahead prediction errors for observations within the period of evaluation. The one-step-ahead predictions are generated from the model specification and parameter estimates. The predictions are inverse transformed (median or mean) and adjustments are removed. The prediction errors (the difference of the dependent series and the predictions) are used to compute the statistics of fit, which are described in the following section "Diagnostic Tests and Statistics of Fit." The results generated by the evaluation process are displayed in the Statistics of Fit table of the Model Viewer window.

**Forecasting**

The forecasting generation process is described graphically in Figure 30.3.
The forecasting process is similar to the model evaluation process described in the preceding section, except that $k$-step-ahead predictions are made from the end of the data through the specified forecast horizon, and prediction standard errors and confidence limits are calculated. The forecasts and confidence limits are displayed in the Forecast plot or table of the Model Viewer window.

**Forecast Combination Models**

This section discusses the computation of predicted values and confidence limits for forecast combination models. See Chapter 25, “Specifying Forecasting Models,” for information on how to specify forecast combination models and their combining weights.

Given the response time series $\{y_t : 1 \leq t \leq n\}$ with previously generated forecasts for the $m$ component models, a combined forecast is created from the component forecasts as follows:

Predictions: \[ \hat{y}_t = \sum_{i=1}^{m} w_i \hat{y}_{i,t} \]

Prediction Errors: \[ \hat{e}_t = y_t - \hat{y}_t \]
where \( \hat{y}_{i,t} \) are the forecasts of the component models and \( w_i \) are the combining weights.

The estimate of the root mean square prediction error and forecast confidence limits for the combined forecast are computed by assuming independence of the prediction errors of the component forecasts, as follows:

\[
\hat{\sigma}_t = \sqrt{\sum_{i=1}^{m} w_i^2 \hat{\sigma}_{i,t}^2}
\]

\[
\text{Confidence Limits:} \quad \pm \hat{\sigma}_t Z_{\alpha/2}
\]

where \( \hat{\sigma}_{i,t} \) are the estimated root mean square prediction errors for the component models, \( \alpha \) is the confidence limit width, \( 1 - \alpha \) is the confidence level, and \( Z_{\alpha/2} \) is the \( \frac{\alpha}{2} \) quantile of the standard normal distribution.

Since, in practice, there may be positive correlation between the prediction errors of the component forecasts, these confidence limits may be too narrow.

**External or User Supplied Forecasts**

This section discusses the computation of predicted values and confidence limits for external forecast models.

Given a response time series \( y_t \) and external forecast series \( \hat{y}_t \), the prediction errors are computed as \( \hat{e}_t = y_t - \hat{y}_t \) for those \( t \) for which both \( y_t \) and \( \hat{y}_t \) are nonmissing. The mean square error (MSE) is computed from the prediction errors.

The variance of the \( k \)-step-ahead prediction errors is set to \( k \) times the MSE. From these variances, the standard errors and confidence limits are computed in the usual way. If the supplied predictions contain so many missing values within the time range of the response series that the MSE estimate cannot be computed, the confidence limits, standard errors, and statistics of fit are set to missing.

**Adjustments**

Adjustment predictors are subtracted from the response time series prior to model parameter estimation, evaluation, and forecasting. After the predictions of the adjusted response time series are obtained from the forecasting model, the adjustments are added back to produce the forecasts.

If \( y_t \) is the response time series and \( X_{i,t}, 1 \leq i \leq m \) are \( m \) adjustment predictor series, then the adjusted response series \( w_t \) is

\[
w_t = y_t - \sum_{i=1}^{m} X_{i,t}
\]

Parameter estimation for the model is performed using the adjusted response time
series \( w_t \). The forecasts \( \hat{w}_t \) of \( w_t \) are adjusted to obtain the forecasts \( \hat{y}_t \) of \( y_t \).

\[
\hat{y}_t = \hat{w}_t + \sum_{i=1}^{m} X_{i,t}
\]

Missing values in an adjustment series are ignored in these computations.

### Series Transformations

For pure ARIMA models, transforming the response time series may aid in obtaining stationary noise series. For general ARIMA models with inputs, transforming the response time series or one or more of the input time series may provide a better model fit. Similarly, the fit of smoothing models may improve when the response series is transformed.

There are four transformations available, for strictly positive series only. Let \( y_t > 0 \) be the original time series, and let \( w_t \) be the transformed series. The transformations are defined as follows:

- **Log** is the logarithmic transformation. Let \( \ln(y_t) \) be the original time series, and let \( w_t = \ln(y_t) \) be the transformed series.

- **Logistic** is the logistic transformation. Let \( \ln(cy_t/(1-cy_t)) \) be the original time series, and let \( w_t = \ln(cy_t/(1-cy_t)) \) be the transformed series. The scaling factor \( c \) is

\[
c = (1 - e^{-6})10^-\text{ceil}[\log_{10} \text{max}(y_t)]
\]

and \( \text{ceil}(x) \) is the smallest integer greater than or equal to \( x \).

- **Square Root** is the square root transformation. Let \( \sqrt{y_t} \) be the original time series, and let \( w_t = \sqrt{y_t} \) be the transformed series.

- **Box Cox** is the Box-Cox transformation. Let \( \left\{ \begin{array}{ll}
\frac{y_t^{\lambda}-1}{\lambda}, & \lambda \neq 0 \\
\ln(y_t), & \lambda = 0
\end{array} \right. \) be the original time series, and let \( w_t = \left\{ \begin{array}{ll}
\frac{y_t^{\lambda}-1}{\lambda}, & \lambda \neq 0 \\
\ln(y_t), & \lambda = 0
\end{array} \right. \) be the transformed series.

Parameter estimation is performed using the transformed series. The transformed model predictions and confidence limits are then obtained from the transformed time-series and these parameter estimates.

The transformed model predictions \( \hat{w}_t \) are used to obtain either the minimum mean absolute error (MMMAE) or minimum mean squared error (MMSE) predictions \( \hat{y}_t \), depending on the setting of the forecast options. The model is then evaluated based on the residuals of the original time series and these predictions. The transformed model confidence limits are inverse-transformed to obtain the forecast confidence limits.
Part 3. General Information

Predictions for Transformed Models

Since the transformations described in the previous section are monotonic, applying the inverse-transformation to the transformed model predictions results in the median of the conditional probability density function at each point in time. This is the minimum mean absolute error (MMAE) prediction.

If \( w_t = F(y_t) \) is the transform with inverse-transform \( y_t = F^{-1}(w_t) \), then

\[
\text{median}(\hat{y}_t) = F^{-1}(E[w_t]) = F^{-1}(\hat{w}_t)
\]

The minimum mean squared error (MMSE) predictions are the mean of the conditional probability density function at each point in time. Assuming that the prediction errors are normally distributed with variance \( \sigma_t^2 \), the MMSE predictions for each of the transformations are as follows:

Logistic

\[
\hat{y}_t = E[e^{w_t}] = \exp(\hat{w}_t + \sigma_t^2/2)
\]

Logistic is the conditional expectation of inverse-logistic transformation.

\[
\hat{y}_t = E\left[\frac{1}{c(1 + \exp(-w_t))}\right]
\]

where the scaling factor \( c = (1 - e^{-6})10^{-\text{ceil} \left[ \log_{10}(\text{max}(y_t)) \right]} \).

Square Root

\[
\hat{y}_t = E[w_t^2] = \hat{w}_t^2 + \sigma_t^2
\]

Square Root is the conditional expectation of the inverse-square root transformation.

Box Cox

\[
\hat{y}_t = \begin{cases} 
E[(\lambda w_t - \lambda) + 1)^{1/\lambda}], & \lambda \neq 0 \\
E[e^{w_t}] = \exp(\hat{w}_t + \frac{1}{2}\sigma_t^2), & \lambda = 0
\end{cases}
\]

The expectations of the inverse logistic and Box-Cox (\( \lambda \neq 0 \)) transformations do not generally have explicit solutions and are computed using numerical integration.

Smoothing Models

This section details the computations performed for the exponential smoothing and Winters method forecasting models.
Smoothing Model Calculations

The descriptions and properties of various smoothing methods can be found in Gardner (1985), Chatfield (1978), and Bowerman and O’Connell (1979). The following section summarizes the smoothing model computations.

Given a time series \{Y_t : 1 \leq t \leq n\}, the underlying model assumed by the smoothing models has the following (additive seasonal) form:

\[ Y_t = \mu_t + \beta_t t + s_p(t) + \epsilon_t \]

where

\[ \mu_t \] represents the time-varying mean term
\[ \beta_t \] represents the time-varying slope
\[ s_p(t) \] represents the time-varying seasonal contribution for one of the \( p \) seasons
\[ \epsilon_t \] are disturbances

For smoothing models without trend terms, \( \beta_t = 0 \); and for smoothing models without seasonal terms, \( s_p(t) = 0 \). Each smoothing model is described in the following sections.

At each time \( t \), the smoothing models estimate the time-varying components described above with the smoothing state. After initialization, the smoothing state is updated for each observation using the smoothing equations. The smoothing state at the last non-missing observation is used for predictions.

**Smoothing State and Smoothing Equations**

Depending on the smoothing model, the smoothing state at time \( t \) will consist of the following:

- \( L_t \) is a smoothed level that estimates \( \mu_t \).
- \( T_t \) is a smoothed trend that estimates \( \beta_t \).
- \( S_{t-j}, j = 0, \ldots, p - 1, \) are seasonal factors that estimate \( s_p(t) \).

The smoothing process starts with an initial estimate of the smoothing state, which is subsequently updated for each observation using the smoothing equations.

The smoothing equations determine how the smoothing state changes as time progresses. Knowledge of the smoothing state at time \( t - 1 \) and that of the time-series value at time \( t \) uniquely determine the smoothing state at time \( t \). The smoothing weights determine the contribution of the previous smoothing state to the current smoothing state. The smoothing equations for each smoothing model are listed in the following sections.
Part 3. General Information

**Smoothing State Initialization**

Given a time series \( \{Y_t : 1 \leq t \leq n\} \), the smoothing process first computes the smoothing state for time \( t = 1 \). However, this computation requires an initial estimate of the smoothing state at time \( t = 0 \) even though no data exists at or before time \( t = 0 \).

An appropriate choice for the initial smoothing state is made by backcasting from time \( t = n \) to \( t = 1 \) to obtain a prediction at \( t = 0 \). The initialization for the backcast is obtained by regression with constant and linear terms and seasonal dummies (additive or multiplicative) as appropriate for the smoothing model. For models with linear or seasonal terms, the estimates obtained by the regression are used for initial smoothed trend and seasonal factors; however, the initial smoothed level for backcasting is always set to the last observation, \( Y_n \).

The smoothing state at time \( t = 0 \) obtained from the backcast is used to initialize the smoothing process from time \( t = 1 \) to \( t = n \) (refer to Chatfield and Yar 1988).

For models with seasonal terms, the smoothing state is normalized so that the seasonal factors \( S_{t-j} \) for \( j = 0, \ldots, p - 1 \) sum to zero for models that assume additive seasonality and average to one for models (such as Winters method) that assume multiplicative seasonality.

**Missing Values**

When a missing value is encountered at time \( t \), the smoothed values are updated using the error-correction form of the smoothing equations with the one-step-ahead prediction error, \( e_t \), set to zero. The missing value is estimated using the one-step-ahead prediction at time \( t - 1 \), that is, \( \hat{Y}_{t-1}(1) \) (refer to Aldrin 1989). The error-correction forms of each of the smoothing models are listed in the following sections.

**Predictions and Prediction Errors**

Predictions are made based on the last known smoothing state. Predictions made at time \( t \) for \( k \) steps ahead are denoted \( \hat{Y}_t(k) \) and the associated prediction errors are denoted \( e_t(k) = Y_{t+k} - \hat{Y}_t(k) \). The prediction equation for each smoothing model is listed in the following sections.

The one-step-ahead predictions refer to predictions made at time \( t - 1 \) for one time unit into the future, that is, \( \hat{Y}_{t-1}(1) \), and the one-step-ahead prediction errors are more simply denoted \( e_t = e_{t-1}(1) = Y_t - \hat{Y}_{t-1}(1) \). The one-step-ahead prediction errors are also the model residuals, and the sum of squares of the one-step-ahead prediction errors is the objective function used in smoothing weight optimization.

The variance of the prediction errors are used to calculate the confidence limits (refer to Sweet 1985, McKenzie 1986, Yar and Chatfield 1990, and Chatfield and Yar (1991)). The equations for the variance of the prediction errors for each smoothing model are listed in the following sections. NOTE: \( \text{var}(e_t) \) is estimated by the mean square of the one-step-ahead prediction errors.
Smoothing Weights

Depending on the smoothing model, the smoothing weights consist of the following:

- $\alpha$ is a level smoothing weight.
- $\gamma$ is a trend smoothing weight.
- $\delta$ is a seasonal smoothing weight.
- $\phi$ is a trend damping weight.

Larger smoothing weights (less damping) permit the more recent data to have a greater influence on the predictions. Smaller smoothing weights (more damping) give less weight to recent data.

Specifying the Smoothing Weights

Typically the smoothing weights are chosen to be from zero to one. (This is intuitive because the weights associated with the past smoothing state and the value of current observation would normally sum to one.) However, each smoothing model (except Winters Method – Multiplicative Version) has an ARIMA equivalent. Weights chosen to be within the ARIMA additive-invertible region will guarantee stable predictions (refer to Archibald 1990 and Gardner 1985). The ARIMA equivalent and the additive-invertible region for each smoothing model are listed in the following sections.

Optimizing the Smoothing Weights

Smoothing weights are determined so as to minimize the sum of squared one-step-ahead prediction errors. The optimization is initialized by choosing from a predetermined grid the initial smoothing weights that result in the smallest sum of squared one-step-ahead prediction errors. The optimization process is highly dependent on this initialization. It is possible that the optimization process will fail due to the inability to obtain stable initial values for the smoothing weights (refer to Greene 1993 and Judge et al 1980), and it is possible for the optimization to result in a local minima.

The optimization process can result in weights to be chosen outside both the zero-to-one range and the ARIMA additive-invertible region. By restricting weight optimization to additive-invertible region, you can obtain a local minimum with stable predictions. Likewise, weight optimization can be restricted to the zero-to-one range or other ranges. It is also possible to fix certain weights to a specific value and optimize the remaining weights.

Standard Errors

The standard errors associated with the smoothing weights are calculated from the Hessian matrix of the sum of squared one-step-ahead prediction errors with respect to the smoothing weights used in the optimization process.

Weights Near Zero or One

Sometimes the optimization process results in weights near zero or one.

For Simple or Double (Brown) Exponential Smoothing, a level weight near zero im-
plies that simple differencing of the time series may be appropriate.

For Linear (Holt) Exponential Smoothing, a level weight near zero implies that the
smoothed trend is constant and that an ARIMA model with deterministic trend may
be a more appropriate model.

For Damped-Trend Linear Exponential Smoothing, a damping weight near one im-
plies that Linear (Holt) Exponential Smoothing may be a more appropriate model.

For Winters Method and Seasonal Exponential Smoothing, a seasonal weight near
one implies that a nonseasonal model may be more appropriate and a seasonal weight
near zero implies that deterministic seasonal factors may be present.

---

**Equations for the Smoothing Models**

**Simple Exponential Smoothing**

The model equation for simple exponential smoothing is

\[ Y_t = \mu_t + \epsilon_t \]

The smoothing equation is

\[ L_t = \alpha Y_t + (1 - \alpha) L_{t-1} \]

The error-correction form of the smoothing equation is

\[ L_t = L_{t-1} + \alpha \epsilon_t \]

(Note: For missing values, \( \epsilon_t = 0 \).)

The \( k \)-step prediction equation is

\[ \hat{Y}_t(k) = L_t \]

The ARIMA model equivalency to simple exponential smoothing is the
ARIMA(0,1,1) model

\[ (1 - B) Y_t = (1 - \theta B) \epsilon_t \]

\[ \theta = 1 - \alpha \]

The moving-average form of the equation is

\[ Y_t = \epsilon_t + \sum_{j=1}^{\infty} \alpha \epsilon_{t-j} \]
For simple exponential smoothing, the additive-invertible region is

\[ \{0 < \alpha < 2\} \]

The variance of the prediction errors is estimated as

\[
var(\epsilon_t(k)) = var(\epsilon_t) \left[ 1 + \sum_{j=1}^{k-1} \alpha^2 \right] = var(\epsilon_t)(1 + (k - 1)\alpha^2)
\]

**Double (Brown) Exponential Smoothing**

The model equation for double exponential smoothing is

\[ Y_t = \mu_t + \beta_t + \epsilon_t \]

The smoothing equations are

\[
L_t = \alpha Y_t + (1 - \alpha)L_{t-1}
\]

\[
T_t = \alpha (L_t - L_{t-1}) + (1 - \alpha)T_{t-1}
\]

This method may be equivalently described in terms of two successive applications of simple exponential smoothing:

\[
S_t^{[1]} = \alpha Y_t + (1 - \alpha)S_{t-1}^{[1]}
\]

\[
S_t^{[2]} = \alpha S_t^{[1]} + (1 - \alpha)S_{t-1}^{[2]}
\]

where \( S_t^{[1]} \) are the smoothed values of \( Y_t \) and \( S_t^{[2]} \) are the smoothed values of \( S_t^{[1]} \). The prediction equation then takes the form:

\[
\hat{Y}_t(k) = (2 + \alpha k/(1 - \alpha))S_t^{[1]} - (1 + \alpha k/(1 - \alpha))S_t^{[2]}
\]

The error-correction form of the smoothing equations is

\[
L_t = L_{t-1} + T_{t-1} + \alpha \epsilon_t
\]

\[
T_t = T_{t-1} + \alpha^2 \epsilon_t
\]

(Note: For missing values, \( \epsilon_t = 0 \).)
The k-step prediction equation is

\[ \hat{Y}_t(k) = L_t + ((k - 1) + 1/\alpha)T_t \]

The ARIMA model equivalency to double exponential smoothing is the ARIMA(0,2,2) model

\[ (1 - B)^2 Y_t = (1 - \theta B)^2 \epsilon_t \]

\[ \theta = 1 - \alpha \]

The moving-average form of the equation is

\[ Y_t = \epsilon_t + \sum_{j=1}^{\infty} (2\alpha + (j - 1)\alpha^2) \epsilon_{t-j} \]

For double exponential smoothing, the additive-invertible region is

\[ \{0 < \alpha < 2\} \]

The variance of the prediction errors is estimated as

\[ \text{var}(\epsilon_t(k)) = \text{var}(\epsilon_t) \left[ 1 + \sum_{j=1}^{k-1} (2\alpha + (j - 1)\alpha^2)^2 \right] \]

**Linear (Holt) Exponential Smoothing**

The model equation for linear exponential smoothing is

\[ Y_t = \mu_t + \beta_t t + \epsilon_t \]

The smoothing equations are

\[ L_t = \alpha Y_t + (1 - \alpha)(L_{t-1} + T_{t-1}) \]

\[ T_t = \gamma(L_t - L_{t-1}) + (1 - \gamma)T_{t-1} \]

The error-correction form of the smoothing equations is

\[ L_t = L_{t-1} + T_{t-1} + \alpha \epsilon_t \]

\[ T_t = T_{t-1} + \alpha \gamma \epsilon_t \]
(Note: For missing values, $e_t = 0$.)
The $k$-step prediction equation is
\[
\hat{Y}_t(k) = L_t + kT_t
\]
The ARIMA model equivalency to linear exponential smoothing is the ARIMA(0,2,2) model
\[
(1 - B)^2 Y_t = (1 - \theta_1 B - \theta_2 B^2) e_t
\]
\[
\theta_1 = 2 - \alpha - \alpha \gamma
\]
\[
\theta_2 = \alpha - 1
\]
The moving-average form of the equation is
\[
Y_t = \epsilon_t + \sum_{j=1}^{\infty} (\alpha + j\alpha \gamma) \epsilon_{t-j}
\]
For linear exponential smoothing, the additive-invertible region is
\[
\{0 < \alpha < 2\}
\]
\[
\{0 < \gamma < 4/\alpha - 2\}
\]
The variance of the prediction errors is estimated as
\[
var(\epsilon_t(k)) = var(\epsilon_t) \left[ 1 + \sum_{j=1}^{k-1} (\alpha + j\alpha \gamma)^2 \right]
\]
**Damped-Trend Linear Exponential Smoothing**
The model equation for damped-trend linear exponential smoothing is
\[
Y_t = \mu_t + \beta_t t + \epsilon_t
\]
The smoothing equations are
\[
L_t = \alpha Y_t + (1 - \alpha)(L_{t-1} + \phi T_{t-1})
\]
\[
T_t = \gamma (L_t - L_{t-1}) + (1 - \gamma) \phi T_{t-1}
\]
The error-correction form of the smoothing equations is

\[ L_t = L_{t-1} + \phi T_{t-1} + \alpha e_t \]

\[ T_t = \phi T_{t-1} + \alpha \gamma e_t \]

(Note: For missing values, \( e_t = 0 \).)

The \( k \)-step prediction equation is

\[ \hat{Y}_t(k) = L_t + \sum_{i=1}^{k} \phi^i T_i \]

The ARIMA model equivalency to damped-trend linear exponential smoothing is the ARIMA(1,1,2) model

\[ (1 - \phi B)(1 - B)Y_t = (1 - \theta_1 B - \theta_2 B^2)e_t \]

\[ \theta_1 = 1 + \phi - \alpha - \alpha \gamma \phi \]

\[ \theta_2 = (\alpha - 1)\phi \]

The moving-average form of the equation (assuming \(|\phi| < 1\)) is

\[ Y_t = \epsilon_t + \sum_{j=1}^{\infty} \left( \alpha + \alpha \gamma \phi (\phi^j - 1)/(\phi - 1) \right) \epsilon_{t-j} \]

For damped-trend linear exponential smoothing, the additive-invertible region is

\[ \{0 < \alpha < 2\} \]

\[ \{0 < \phi \gamma < 4/\alpha - 2\} \]

The variance of the prediction errors is estimated as

\[ var(e_t(k)) = var(e_t) \left[ 1 + \sum_{j=1}^{k-1} \left( \alpha + \alpha \gamma \phi (\phi^j - 1)/(\phi - 1) \right)^2 \right] \]
Seasonal Exponential Smoothing

The model equation for seasonal exponential smoothing is

\[ Y_t = \mu_t + s_p(t) + \epsilon_t \]

The smoothing equations are

\[ L_t = \alpha(Y_t - S_{t-p}) + (1 - \alpha)L_{t-1} \]
\[ S_t = \delta(Y_t - L_t) + (1 - \delta)S_{t-p} \]

The error-correction form of the smoothing equations is

\[ L_t = L_{t-1} + \alpha \epsilon_t \]
\[ S_t = S_{t-p} + \delta(1 - \alpha) \epsilon_t \]

(Note: For missing values, \( \epsilon_t = 0 \).)

The \( k \)-step prediction equation is

\[ \hat{Y}_t(k) = L_t + S_{t-p+k} \]

The ARIMA model equivalency to seasonal exponential smoothing is the ARIMA(0,1,p+1)(0,1,0)\( _p \) model

\[ (1 - B)(1 - B^p)Y_t = (1 - \theta_1 B - \theta_2 B^p - \theta_3 B^{p+1})\epsilon_t \]

\[ \theta_1 = 1 - \alpha \]
\[ \theta_2 = 1 - \delta(1 - \alpha) \]
\[ \theta_3 = (1 - \alpha)(\delta - 1) \]

The moving-average form of the equation is

\[ Y_t = \epsilon_t + \sum_{j=1}^{\infty} \psi_j \epsilon_{t-j} \]

\[ \psi_j = \begin{cases} \alpha & \text{for } j \mod p \neq 0 \\ \alpha + \delta(1 - \alpha) & \text{for } j \mod p = 0 \end{cases} \]
Part 3. General Information

For seasonal exponential smoothing, the additive-invertible region is

\[
\{ \max(-p\alpha, 0) < \delta(1 - \alpha) < (2 - \alpha) \}
\]

The variance of the prediction errors is estimated as

\[
\text{var}(\epsilon_t(k)) = \text{var}(\epsilon_t) \left[ 1 + \sum_{j=1}^{k-1} \psi_j^2 \right]
\]

Winters Method – Additive Version

The model equation for the additive version of Winters method is

\[
y_t = \mu_t + \beta_t t + s_p(t) + \epsilon_t
\]

The smoothing equations are

\[
L_t = \alpha(y_t - s_{t-p}) + (1 - \alpha)(L_{t-1} + T_{t-1})
\]

\[
T_t = \gamma(L_t - L_{t-1}) + (1 - \gamma)T_{t-1}
\]

\[
S_t = \delta(y_t - L_t) + (1 - \delta)S_{t-p}
\]

The error-correction form of the smoothing equations is

\[
L_t = L_{t-1} + T_{t-1} + \alpha \epsilon_t
\]

\[
T_t = T_{t-1} + \alpha \gamma \epsilon_t
\]

\[
S_t = S_{t-p} + \delta(1 - \alpha) \epsilon_t
\]

(Note: For missing values, \(\epsilon_t = 0\).)

The \(k\)-step prediction equation is

\[
\hat{y}_t(k) = L_t + kT_t + S_{t-p+k}
\]

The ARIMA model equivalency to the additive version of Winters method is the ARIMA(0,1,p+1)(0,1,0)\(p\) model

\[
(1 - B)(1 - B^p)y_t = \left[ 1 - \sum_{i=1}^{p+1} \theta_i B^i \right] \epsilon_t
\]
The moving-average form of the equation is

$$Y_t = \epsilon_t + \sum_{j=1}^{\infty} \psi_j \epsilon_{t-j}$$

$$\psi_j = \begin{cases} 
\alpha + j\alpha\gamma & \text{for } j \mod p \neq 0 \\
\alpha + j\alpha\gamma + \delta(1 - \alpha) & \text{for } j \mod p = 0 
\end{cases}$$

For the additive version of Winters method (see Archibald 1990), the additive-invertible region is

$$\{\max(-p\alpha, 0) < \delta(1 - \alpha) < (2 - \alpha)\}$$

$$\{0 < \alpha\gamma < 2 - \alpha - \delta(1 - \alpha)(1 - \cos(\vartheta))\}$$

where $\vartheta$ is the smallest non-negative solution to the equations listed in Archibald (1990).

The variance of the prediction errors is estimated as

$$\text{var}(\epsilon_t(k)) = \text{var}(\epsilon_t) \left[ 1 + \sum_{j=1}^{k-1} \psi_j^2 \right]$$

**Winters Method – Multiplicative Version**

In order to use the multiplicative version of Winters method, the time series and all predictions must be strictly positive.

The model equation for the multiplicative version of Winters method is

$$Y_t = (\mu_t + \beta_t t) s_p(t) + \epsilon_t$$

The smoothing equations are

$$L_t = \alpha(Y_t/S_{t-p}) + (1 - \alpha)(L_{t-1} + T_{t-1})$$

$$T_t = \gamma(L_t - L_{t-1}) + (1 - \gamma)T_{t-1}$$

$$S_t = \delta(Y_t/L_t) + (1 - \delta)S_{t-p}$$
The error-correction form of the smoothing equations is

\[ L_t = L_{t-1} + T_{t-1} + \alpha e_t / S_{t-p} \]

\[ T_t = T_{t-1} + \alpha \gamma e_t / S_{t-p} \]

\[ S_t = S_{t-p} + \delta (1 - \alpha)e_t / L_t \]

(Note: For missing values, \( e_t = 0 \).)

The \( k \)-step prediction equation is

\[ \hat{Y}_t(k) = (L_t + kT_t)S_{t-p+k} \]

The multiplicative version of Winters method does not have an ARIMA equivalent; however, when the seasonal variation is small, the ARIMA additive-invertible region of the additive version of Winters method described in the preceding section can approximate the stability region of the multiplicative version.

The variance of the prediction errors is estimated as

\[ \text{var}(e_t(k)) = \text{var}(\varepsilon_t) \left[ \sum_{i=0}^{\infty} \sum_{j=0}^{p-1} (\psi_{j+i+p}S_{t+k}/S_{t+k-j})^2 \right] \]

where \( \psi_j \) are as described for the additive version of Winters method, and \( \psi_j = 0 \) for \( j \geq k \).

**ARIMA Models**

AutoRegressive Integrated Moving-Average, or ARIMA, models predict values of a dependent time series with a linear combination of its own past values, past errors (also called shocks or innovations), and current and past values of other time series (predictor time series).

The Time Series Forecasting system uses the ARIMA procedure of SAS/ETS software to fit and forecast ARIMA models. The maximum likelihood method is used for parameter estimation. Refer to Chapter 7, “The ARIMA Procedure,” for details of ARIMA model estimation and forecasting.

This section summarizes the notation used for ARIMA models.

**Notation for ARIMA Models**

A dependent time series that is modeled as a linear combination of its own past values and past values of an error series is known as a (pure) ARIMA model.
Nonseasonal ARIMA Model Notation
The order of an ARIMA model is usually denoted by the notation ARIMA\((p,d,q)\), where

\[ p \] is the order of the autoregressive part
\[ d \] is the order of the differencing (rarely should \( d > 2 \) be needed)
\[ q \] is the order of the moving-average process

Given a dependent time series \( \{Y_t : 1 \leq t \leq n\} \), mathematically the ARIMA model is written as

\[
(1 - B)^dY_t = \mu + \frac{\theta(B)}{\phi(B)} \alpha_t
\]

where

\[ t \] indexes time
\[ \mu \] is the mean term
\[ B \] is the backshift operator; that is, \( BX_t = X_{t-1} \)
\[ \phi(B) \] is the autoregressive operator, represented as a polynomial in the back shift operator: \( \phi(B) = 1 - \phi_1 B - \ldots - \phi_p B^p \)
\[ \theta(B) \] is the moving-average operator, represented as a polynomial in the back shift operator: \( \theta(B) = 1 - \theta_1 B - \ldots - \theta_q B^q \)
\[ \alpha_t \] is the independent disturbance, also called the random error

For example, the mathematical form of the ARIMA(1,1,2) model is

\[
(1 - B)Y_t = \mu + \frac{(1 - \theta_1 B - \theta_2 B^2)}{(1 - \phi_1 B)} \alpha_t
\]

Seasonal ARIMA Model Notation
Seasonal ARIMA models are expressed in factored form by the notation ARIMA\((p,d,q)(P,D,Q)_s\), where

\[ P \] is the order of the seasonal autoregressive part
\[ D \] is the order of the seasonal differencing (rarely should \( D > 1 \) be needed)
\[ Q \] is the order of the seasonal moving-average process
\[ s \] is the length of the seasonal cycle

Given a dependent time series \( \{Y_t : 1 \leq t \leq n\} \), mathematically the ARIMA seasonal model is written as

\[
(1 - B)^d(1 - B^s)^D Y_t = \mu + \frac{\theta(B)\theta_s(B^s)}{\phi(B)\phi_s(B^s)} \alpha_t
\]
where

\[ \phi_s(B^s) \]

is the seasonal autoregressive operator, represented as a polynomial in the back shift operator:

\[ \phi_s(B^s) = 1 - \phi_{s,1}B^s - \cdots - \phi_{s,p}B^{sp} \]

\[ \theta_s(B^s) \]

is the seasonal moving-average operator, represented as a polynomial in the back shift operator:

\[ \theta_s(B^s) = 1 - \theta_{s,1}B^s - \cdots - \theta_{s,q}B^{sq} \]

For example, the mathematical form of the ARIMA(1,0,1)(1,1,2) model is

\[ (1 - B^{12})Y_t = \mu + \frac{(1 - \theta_1B)(1 - \theta_{1,1}B^{12} - \theta_{1,2}B^{24})}{(1 - \phi_1B)(1 - \phi_{1,1}B^{12})}\alpha_t \]

**Abbreviated Notation for ARIMA Models**

If the differencing order, autoregressive order, or moving-average order is zero, the notation is further abbreviated as

- \( I(d) \)
- \( AR(p) \)
- \( IAR(p,d) \)
- \( MA(q) \)
- \( IMA(d,q) \)
- \( ARMA(p,q) \)

**Notation for Transfer Functions**

A transfer function can be used to filter a predictor time series to form a dynamic regression model.

Let \( Y_t \) be the dependent series and let \( X_t \) be the predictor series, and let \( \Psi(B) \) be a linear filter or transfer function for the effect of \( X_t \) on \( Y_t \). The ARIMA model is then

\[ (1 - B)^d(1 - B^s)^DY_t = \mu + \Psi(B)X_t + \frac{\theta(B)\theta_s(B^s)}{\phi(B)\phi_s(B^s)}\alpha_t \]

This model is called a dynamic regression of \( Y_t \) on \( X_t \).

**Nonseasonal Transfer Function Notation**

Given the \( i \)th predictor time series \( \{X_{i,t} : 1 \leq t \leq n\} \), the transfer function is written as \([\text{Dif}(d_i)\text{Lag}(k_i)\text{N}(q_i)\text{D}(p_i)]\) where

- \( d_i \) is the simple order of the differencing for the \( i \)th predictor time series, \((1 - B)^{d_i}X_{i,t}\) (rarely should \( d_i > 2 \) be needed)
- \( k_i \) is the pure time delay (lag) for the effect of the \( i \)th predictor time series, \( X_{i,t}B^{k_i} = X_{i,t-k_i} \)
Chapter 30. ARIMA Models

\( p_i \) is the simple order of the denominator for the \( i \)th predictor time series

\( q_i \) is the simple order of the numerator for the \( i \)th predictor time series

The mathematical notation used to describe a transfer function is

\[
\Psi_i(B) = \frac{\omega_i(B)}{\delta_i(B)}(1 - B)^{d_i}B^{k_i}
\]

where

\[ B \] is the backshift operator; that is, \( BX_t = X_{t-1} \)

\[ \delta_i(B) \] is the denominator polynomial of the transfer function for the \( i \)th predictor time series: \( \delta_i(B) = 1 - \delta_{i,1}B - \ldots - \delta_{i,p_i}B^{p_i} \)

\[ \omega_i(B) \] is the numerator polynomial of the transfer function for the \( i \)th predictor time series: \( \omega_i(B) = 1 - \omega_{i,1}B - \ldots - \omega_{i,q_i}B^{q_i} \)

The numerator factors for a transfer function for a predictor series are like the MA part of the ARMA model for the noise series. The denominator factors for a transfer function for a predictor series are like the AR part of the ARMA model for the noise series. Denominator factors introduce exponentially weighted, infinite distributed lags into the transfer function.

For example, the transfer function for the \( i \)th predictor time series with

\[ k_i = 3 \quad \text{time lag is 3} \]

\[ d_i = 1 \quad \text{simple order of differencing is one} \]

\[ p_i = 1 \quad \text{simple order of the denominator is one} \]

\[ q_i = 2 \quad \text{simple order of the numerator is two} \]

would be written as \([\text{Dif}(1)\text{Lag}(3)\text{N}(2)/\text{D}(1)]\). The mathematical notation for the transfer function in this example is

\[
\Psi_i(B) = \frac{(1 - \omega_{i,1}B - \omega_{i,2}B^2)}{(1 - \delta_{i,1}B)}(1 - B)B^3
\]

**Seasonal Transfer Function Notation**

The general transfer function notation for the \( i \)th predictor time series \( X_{i,t} \) with seasonal factors is \([\text{Dif}(d_i)(D_i)_s \text{Lag}(k_i) \text{N}(q_i)(Q_i)_s/\text{D}(p_i)(P_i)_s]\) where

\[ D_i \] is the seasonal order of the differencing for the \( i \)th predictor time series (rarely should \( D_i > 1 \) be needed)

\[ P_i \] is the seasonal order of the denominator for the \( i \)th predictor time series (rarely should \( P_i > 2 \) be needed)
Part 3. General Information

\( Q_i \) is the seasonal order of the numerator for the \( i \)th predictor time series, (rarely should \( Q_i \) > 2 be needed)

\( s \) is the length of the seasonal cycle

The mathematical notation used to describe a seasonal transfer function is

\[
\Psi_i(B) = \frac{\omega_{s,i}(B^s)}{\delta_i(B)\delta_{s,i}(B^s)}(1 - B)^{d_i}(1 - B^s)^{D_i}B^{k_i}
\]

where

\( \delta_{s,i}(B^s) \) is the denominator seasonal polynomial of the transfer function for the \( i \)th predictor time series:

\[
\delta_{s,i}(B) = 1 - \delta_{s,i,1}B - \ldots - \delta_{s,i,p_i}B^{p_i}
\]

\( \omega_{s,i}(B^s) \) is the numerator seasonal polynomial of the transfer function for the \( i \)th predictor time series:

\[
\omega_{s,i}(B) = 1 - \omega_{s,i,1}B - \ldots - \omega_{s,i,q_i}B^{q_i}
\]

For example, the transfer function for the \( i \)th predictor time series \( X_{i,t} \) whose seasonal cycle \( s = 12 \) with

- \( d_i = 2 \) simple order of differencing is two
- \( D_i = 1 \) seasonal order of differencing is one
- \( q_i = 2 \) simple order of the numerator is two
- \( Q_i = 1 \) seasonal order of the numerator is one

would be written as \([\text{Dif}(2)(1)_s \text{ N}(2)(1)_s] \). The mathematical notation for the transfer function in this example is

\[
\Psi_i(B) = (1 - \omega_{i,1}B - \omega_{i,2}B^2)(1 - \omega_{s,i,1}B^{12})(1 - B)^2(1 - B^{12})
\]

Note: In this case, \([\text{Dif}(2)(1)_s \text{ N}(2)(1)_s] = [\text{Dif}(2)(1)_s \text{ Lag}(0)\text{N}(2)(1)_s / \text{D}(0)(0)_s] \).

Predictor Series

This section discusses time trend curves, seasonal dummies, interventions, and adjustments.

Time Trend Curves

When you specify a time trend curve as a predictor in a forecasting model, the system computes a predictor series that is a deterministic function of time. This variable is
then included in the model as a regressor, and the trend curve is fit to the dependent series by linear regression, in addition to other predictor series.

Some kinds of nonlinear trend curves are fit by transforming the dependent series. For example, the exponential trend curve is actually a linear time trend fit to the logarithm of the series. For these trend curve specifications, the series transformation option is set automatically, and you cannot independently control both the time trend curve and transformation option.

The computed time trend variable is included in the output data set in a variable named in accordance with the trend curve type. Let \( t \) represent the observation count from the start of the period of fit for the model, and let \( X_t \) represent the value of the time trend variable at observation \( t \) within the period of fit. The names and definitions of these variables are as follows. (NOTE: These deterministic variables are reserved variable names.)

<table>
<thead>
<tr>
<th>Trend Type</th>
<th>Variable Name</th>
<th>Formula</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Trend</td>
<td><em>LINEAR</em></td>
<td>( X_t = t - c )</td>
<td></td>
</tr>
<tr>
<td>Quadratic Trend</td>
<td><em>QUAD</em></td>
<td>( X_t = (t - c)^2 )</td>
<td>Note that a quadratic trend implies a linear trend as a special case and results in two regressors <em>QUAD</em> and <em>LINEAR</em>.</td>
</tr>
<tr>
<td>Cubic Trend</td>
<td><em>CUBE</em></td>
<td>( X_t = (t - c)^3 )</td>
<td>Note that a cubic trend implies a quadratic trend as a special case and results in three regressors <em>CUBE</em>, <em>QUAD</em>, and <em>LINEAR</em>.</td>
</tr>
<tr>
<td>Logistic Trend</td>
<td><em>LOGIT</em></td>
<td>( X_t = t )</td>
<td>The model is a linear time trend applied to the logistic transform of the dependent series. Thus, specifying a logistic trend is equivalent to specifying the Logistic series transformation and a linear time trend. A logistic trend predictor can be used only in conjunction with the logistic transformation, which is set automatically when you specify logistic trend.</td>
</tr>
<tr>
<td>Logarithmic Trend</td>
<td><em>LOG</em></td>
<td>( X_t = \ln(t) )</td>
<td></td>
</tr>
<tr>
<td>Exponential Trend</td>
<td><em>EXP</em></td>
<td>( X_t = t )</td>
<td>The model is a linear time trend applied to the logarithms of the dependent series. Thus, specifying an exponential trend is equivalent to specifying the log series transformation and a linear time trend. An exponential trend predictor can be used only in conjunction with the log transformation, which is set automatically when you specify exponential trend.</td>
</tr>
<tr>
<td>Hyperbolic Trend</td>
<td><em>HYP</em></td>
<td>( X_t = 1/t )</td>
<td></td>
</tr>
<tr>
<td>Power Curve Trend</td>
<td><em>POW</em></td>
<td>( X_t = \ln(t) )</td>
<td>The model is a logarithmic time trend applied to the logarithms of the dependent series. Thus, specifying a power curve is equivalent to specifying the log series transformation and a logarithmic time trend. A power curve predictor can be used only in conjunction with the log transformation, which is set automatically when you specify a power curve trend.</td>
</tr>
</tbody>
</table>
**Part 3. General Information**

EXP(A+B/TIME) Trend  Variable name _ERT_, with \( X_t = 1/t \). The model is a hyperbolic time trend applied to the logarithms of the dependent series. Thus, specifying this trend curve is equivalent to specifying the log series transformation and a hyperbolic time trend. This trend curve can be used only in conjunction with the log transformation, which is set automatically when you specify this trend.

### Intervention Effects

Interventions are used for modeling events that occur at specific times. That is, they are known changes that affect the dependent series or outliers.

The \( i \)th intervention series is included in the output data set with variable name _INTVi_, which is a reserved variable name.

#### Point Interventions

The point intervention is a one-time event. The \( i \)th intervention series \( X_{i,t} \) has a point intervention at time \( t_{int} \) when the series is nonzero only at time \( t_{int} \), that is,

\[
X_{i,t} = \begin{cases} 
1, & t = t_{int} \\
0, & otherwise 
\end{cases}
\]

#### Step Interventions

Step interventions are continuing, and the input time series flags periods after the intervention. For a Step intervention, before time \( t_{int} \), the \( i \)th intervention series \( X_{i,t} \) is zero and then steps to a constant level thereafter, that is,

\[
X_{i,t} = \begin{cases} 
1, & t \geq t_{int} \\
0, & otherwise 
\end{cases}
\]

#### Ramp Interventions

A Ramp intervention is a continuing intervention that increases linearly after the intervention time. For a Ramp intervention, before time \( t_{int} \), the \( i \)th intervention series \( X_{i,t} \) is zero and increases linearly thereafter, that is, proportional to time.

\[
X_{i,t} = \begin{cases} 
t - t_{int}, & t \geq t_{int} \\
0, & otherwise 
\end{cases}
\]

### Intervention Effect

Given the \( i \)th intervention series \( X_{i,t} \), you can define how the intervention takes effect by filters (transfer functions) of the form

\[
\psi_i(B) = \frac{1 - \omega_{i,1}B - \ldots - \omega_{i,q_i}B^{q_i}}{1 - \delta_{i,1}B - \ldots - \delta_{i,p_i}B^{p_i}}
\]

where \( B \) is the backshift operator \( B_{yt} = y_{t-1} \).
Chapter 30. Series Diagnostic Tests

The denominator of the transfer function determines the decay pattern of the intervention effect, whereas the numerator terms determine the size of the intervention effect time window.

For example, the following intervention effects are associated with the respective transfer functions.

<table>
<thead>
<tr>
<th>Intervention Type</th>
<th>Transfer Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Immediately</td>
<td>$\Psi_i(B) = 1$</td>
</tr>
<tr>
<td>Gradually</td>
<td>$\Psi_i(B) = 1/(1 - \delta_i B)$</td>
</tr>
<tr>
<td>1 Lag window</td>
<td>$\Psi_i(B) = 1 - \omega_i,1 B$</td>
</tr>
<tr>
<td>3 Lag window</td>
<td>$\Psi_i(B) = 1 - \omega_i,1 B - \omega_i,2 B^2 - \omega_i,3 B^3$</td>
</tr>
</tbody>
</table>

### Intervention Notation

Interventions The notation used to describe intervention effects has the form \( \text{type}:t_{int}(q_i)/(p_i) \), where \( \text{type} \) is Point, Step, or Ramp; \( t_{int} \) is the time of the intervention (for example, OCT87); \( q_i \) is the transfer function numerator order; and \( p_i \) is the transfer function denominator order. If \( q_i = 0 \), the part "\((q_i)\)" is omitted; if \( p_i = 0 \), the part "/(p_i)" is omitted.

In the Intervention Specification window, the Number of Lags option specifies the transfer function numerator order \( q_i \), and the Effect Decay Pattern option specifies the transfer function denominator order \( p_i \). In the Effect Decay Pattern options, values and resulting \( p_i \) are: \textbf{None}, \( p_i = 0 \); \textbf{Exp}, \( p_i = 1 \); \textbf{Wave}, \( p_i = 2 \).

For example, a step intervention with date 08MAR90 and effect pattern \textbf{Exp} is denoted "Step:08MAR90/(1)" and has a transfer function filter $\Psi_i(B) = 1/(1 - \delta_1 B)$. A ramp intervention immediately applied on 08MAR90 is denoted "Ramp:08MAR90" and has a transfer function filter $\Psi_i(B) = 1$.

### Seasonal Dummy Inputs

For a seasonal cycle of length \( s \), the seasonal dummy regressors include \( \{X_{i,t} : 1 \leq i \leq (s - 1), 1 \leq t \leq n\} \) for models that include an intercept term and \( \{X_{i,t} : 1 \leq i \leq s, 1 \leq t \leq n\} \) for models that exclude an intercept term. Each element of a seasonal dummy regressor is either zero or one, based on the rule:

$$X_{i,t} = \begin{cases} 
1, & \text{when } i = t \mod s \\
0, & \text{otherwise}
\end{cases}$$

Note that if the model includes an intercept term, the number of seasonal dummy regressors is one less than \( s \) to ensure that the linear system is full rank.

The seasonal dummy variables are included in the output data set with variable names prefixed with "SDUMMY\(i\)" and sequentially numbered. They are reserved variable names.
Part 3. General Information

Series Diagnostic Tests

This section describes the diagnostic tests used to determine the kinds of forecasting models appropriate for a series.

The series diagnostics are a set of heuristics that provide recommendations on whether or not the forecasting model should contain a log transform, trend terms, and seasonal terms. These recommendations are used by the automatic model selection process to restrict the model search to a subset of the model selection list. (You can disable this behavior using the Automatic Model Selection Options window.)

The tests used by the series diagnostics will not always produce the correct classification of the series. They are intended to accelerate the process of searching for a good forecasting model for the series, but you should not rely on them if finding the very best model is important to you.

If you have information about the appropriate kinds of forecasting model (perhaps from studying the plots and autocorrelations shown in the Series Viewer window), you can set the series diagnostic flags in the Series Diagnostics window. Select the "YES", "NO", or "MAYBE" values for the Log Transform, Trend, and Seasonality options in the Series Diagnostics window as you think appropriate.

The series diagnostics tests are intended as a heuristic tool only, and no statistical validity is claimed for them. These tests may be modified and enhanced in future releases of the Time Series Forecasting System. The testing strategy is as follows:

1. **Log transform test.** The log test fits a high order autoregressive model to the series and to the log of the series and compares goodness-of-fit measures for the prediction errors of the two models. If this test finds that log transforming the series is suitable, the Log Transform option is set to "YES," and the subsequent diagnostic tests are performed on the log transformed series.

2. **Trend test.** The resultant series is tested for presence of a trend using an augmented Dickey-Fuller test and a random walk with drift test. If either test finds that the series appears to have a trend, the Trend option is set to "YES," and the subsequent diagnostic tests are performed on the differenced series.

3. **Seasonality test.** The resultant series is tested for seasonality. A seasonal dummy model with AR(1) errors is fit and the joint significance of the seasonal dummy estimates is tested. If the seasonal dummies are significant, the AIC statistic for this model is compared to the AIC for and AR(1) model without seasonal dummies. If the AIC for the seasonal model is lower than that of the non-seasonal model, the Seasonal option is set to "YES".

Statistics of Fit

This section explains the goodness-of-fit statistics reported to measure how well different models fit the data. The statistics of fit for the various forecasting models can
be viewed or stored in a data set using the Model Viewer window.

The various statistics of fit reported are as follows. In these formula, \( n \) is the number of nonmissing observations and \( k \) is the number of fitted parameters in the model.

**Number of Nonmissing Observations.**
The number of nonmissing observations used to fit the model.

**Number of Observations.**
The total number of observations used to fit the model, including both missing and nonmissing observations.

**Number of Missing Actuals.**
The number of missing actual values.

**Number of Missing Predicted Values.**
The number of missing predicted values.

**Number of Model Parameters.**
The number of parameters fit to the data. For combined forecast, this is the number of forecast components.

**Total Sum of Squares (Uncorrected).**
The total sum of squares for the series, SST, uncorrected for the mean: \( \sum_{t=1}^{n} y_t^2 \).

**Total Sum of Squares (Corrected).**
The total sum of squares for the series, SST, corrected for the mean:
\[
\sum_{t=1}^{n} (y_t - \bar{y})^2,
\]
where \( \bar{y} \) is the series mean.

**Sum of Square Errors.**
The sum of the squared prediction errors, SSE. \( SSE = \sum_{t=1}^{n} (y_t - \hat{y}_t)^2 \), where \( \hat{y}_t \) is the one-step predicted value.

**Mean Square Error.**
The mean squared prediction error, MSE, calculated from the one-step-ahead forecasts. \( MSE = \frac{1}{n} SSE \). This formula enables you to evaluate small holdout samples.

**Root Mean Square Error.**
The root mean square error (RMSE), \( \sqrt{MSE} \).

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

**Mean Absolute Error.**
The mean absolute prediction error, \( \frac{1}{n} \sum_{t=1}^{n} |y_t - \hat{y}_t| \).

**R-Square.**
The \( R^2 \) statistic, \( R^2 = 1 - \frac{SSE}{SST} \). If the model fits the series badly, the
model error sum of squares, \( SSE \), may be larger than \( SST \) and the \( R^2 \) statistic will be negative.

**Adjusted R-Square.**
The adjusted \( R^2 \) statistic, 
\[
1 - \left( \frac{n-1}{n-k} \right) (1 - R^2).
\]

**Amemiya’s Adjusted R-Square.**
Amemiya’s adjusted \( R^2 \), 
\[
1 - \left( \frac{n+k}{n-k} \right) (1 - R^2).
\]

**Random Walk R-Square.**
The random walk \( R^2 \) statistic (Harvey’s \( R^2 \) statistic using the random walk model for comparison), 
\[
1 - \left( \frac{n-1}{n} \right) \frac{SSE}{RWSSE},
\]
where \( RWSSE = \sum_{t=2}^{n} (y_t - y_{t-1} - \mu)^2 \), and 
\[
\mu = \frac{1}{n-1} \sum_{t=2}^{n} (y_t - y_{t-1}).
\]

**Akaike’s Information Criterion.**
Akaike’s information criterion (AIC), 
\[
n \ln (MSE) + 2k.
\]

**Schwarz Bayesian Information Criterion.**
Schwarz Bayesian information criterion (SBC or BIC), 
\[
n \ln (MSE) + k \ln (n).
\]

**Amemiya’s Prediction Criterion.**
Amemiya’s prediction criterion, 
\[
\frac{1}{n} SST \left( \frac{n+k}{n-k} \right) (1 - R^2) = \left( \frac{n+k}{n-k} \right) \frac{1}{n} SSE.
\]

**Maximum Error.**
The largest prediction error.

**Minimum Error.**
The smallest prediction error.

**Maximum Percent Error.**
The largest percent prediction error, 
\[
100 \max \left( \frac{y_t - \hat{y}_t}{y_t} \right).
\]
The summation ignores observations where \( y_t = 0 \).

**Minimum Percent Error.**
The smallest percent prediction error, 
\[
100 \min \left( \frac{y_t - \hat{y}_t}{y_t} \right).
\]
The summation ignores observations where \( y_t = 0 \).

**Mean Error.**
The mean prediction error, 
\[
\frac{1}{n} \sum_{t=1}^{n} (y_t - \hat{y}_t).
\]

**Mean Percent Error.**
The mean percent prediction error, 
\[
\frac{100}{n} \sum_{t=1}^{n} \left( \frac{y_t - \hat{y}_t}{y_t} \right).
\]
The summation ignores observations where \( y_t = 0 \).
References


Box, G.E.P. and Tiao, G.C. (1975), "Intervention Analysis with Applications to Economic and Environmental Problems," JASA, 70, 70-79.


Part 3. General Information


