INDIVIDUAL SPECS (with starting page number)

ARIMA

DESCRIPTION

Specifies the ARIMA part of the regARIMA model. This defines a pure ARIMA model if the **regression** spec is absent. The ARIMA part of the model may include multiplicative seasonal factors and operators with missing lags. Using the **ar** and **ma** arguments, initial values for the individual AR and MA parameters can be specified for the iterative estimation. Also, individual parameters can be held fixed at these initial values while the rest of the parameters are estimated.

USAGE

arima {**model** = `([2 3] 1 1)(0 1 1)12`
title = `"ARIMA Model"`
ar = `(0.3f, -0.14)`
ma = `(-0.7 0.85f)` }

ARGUMENTS

ar Specifies initial values for nonseasonal and seasonal autoregressive parameters in the order that they appear in the **model** argument. If present, the **ar** argument must assign initial values to *all* AR parameters in the model. Initial values are assigned to parameters either by specifying the value in the argument list or by explicitly indicating that it is missing. Missing values take on their default value of 0.1. For example, for a model with two AR parameters, `ar=(0.7, )` is equivalent to `ar=(0.7,0.1)`, but `ar=(0.7)` is not allowed. For a model with three AR parameters, `ar=(0.8,,-0.4)` is equivalent to `ar=(0.8,0.1,-0.4)`. To hold a parameter fixed during estimation at its initial value, immediately follow the value in the **ar** list with an '`f`', e.g., `ar=(0.7f, 0.1)`.

ma Specifies initial values for all moving average parameters in the same way **ar** does so for autoregressive parameters.

model Specifies the ARIMA part of the model. The format follows standard Box-Jenkins (1976) notation. In this notation a nonseasonal ARIMA model is specified as $(p\ d\ q)$, where p is the nonseasonal AR order, d is the number of nonseasonal differences, and q is the nonseasonal MA order. A multiplicative seasonal ARIMA model is specified as $(p\ d\ q)(P\ D\ Q)$, where p, d, and q are as before, P is the seasonal AR order, D is the number of seasonal differences, and Q is the seasonal MA order. Here, the first ARIMA factor, $(p\ d\ q)$, is assumed to be nonseasonal (i.e., its period is one) and the second ARIMA factor, $(P\ D\ Q)$, is assumed to be seasonal with the seasonal period set in the **series** spec. More than two

ARIMA factors can be specified, and ARIMA factors can explicitly be given seasonal periods that differ from the default choices. See DETAILS for more information.

The operator orders ($p\ d\ q$) in the ARIMA factors may be separated by spaces or commas, e.g., `(0 1 1)` is the same as `(0,1,1)`. Operators with missing lags are specified by enclosing those lags present in brackets. For example, `model = ([2 3] 0 0)` specifies the model $(1 - \phi_2 B^2 - \phi_3 B^3) z_t = a_t$.

print and **save** No output tables are available for this spec.

title Specifies a title for the ARIMA model, in quotes. It must be less than 80 characters. The title appears above the ARIMA model description and the table of estimates. The default is to print `ARIMA Model`.

DETAILS

The model argument may include as many ARIMA factors as desired. However, there is a limit of 108 total AR, MA, and differencing coefficients in the model. Also, the maximum lag of any AR or MA parameter is 36, and the maximum number of differences in any ARIMA factor (nonseasonal or seasonal) is 3. (The latter two limits can be changed—see Section 2.8.)

In general, ARIMA factors are specified in the standard $(p\ d\ q)_s$ format, where s is the seasonal period of the operator. Thus, putting `(0 1 1)6` in the model argument includes differencing by $1 - B^6$ and a $1 - \theta_6 B^6$ MA term in the model. However, if the seasonal period s is not specified after an ARIMA factor, it is determined according to the following default rules. The first ARIMA factor without a specified seasonal period is assumed to be nonseasonal, i.e., its seasonal period is assumed to be one. The second ARIMA factor without a specified seasonal period is assumed to be a seasonal factor with the seasonal period set in the **series** spec. For example, if `period = 12` is specified in the **series** spec (or if the period is set to 12 because the start date there is given as *year.month*), then `model = (0 1 1)(0 1 1)` and `model = (0 1 1)1(0 1 1)12` are equivalent. If additional ARIMA factors are specified, these are assumed to be nonseasonal unless they are explicitly given a seasonal period. See Example 6 for an illustration of a model with three ARIMA factors. Note that if the seasonal period is one, then any ARIMA factors without a specified seasonal period have period one.

Users should not specify initial values for MA parameters that yield an MA polynomial with roots inside the unit circle. (See Section 4.4.) Doing so will cause the program to stop and print an error message asking the user to respecify the initial parameters and rerun the program. Initial parameters that yield an MA polynomial with roots *on* the unit circle are allowed only if this noninvertible polynomial is not being estimated.

That is, this is allowed if no estimation is being done, or if the parameters in this polynomial are specified as fixed during estimation. For example, if a model has a first order seasonal MA parameter as the only MA parameter, then `ma=(1.0f)` is always allowed, `ma=(1.0)` is allowed only if no estimation is done, and `ma=(1.1)` is never allowed.

If the likelihood function that is exact for AR polynomials is used (`exact=arma`, which is the default—see the **estimate** spec), users should not specify initial values for AR parameters that yield a nonstationary AR polynomial (one with roots on or inside the unit circle). Doing so will cause the program to stop and print an error message asking the user to respecify the initial parameters and rerun the program.

The use of fixed coefficients in the ARIMA model can invalidate AIC and the other model selection statistics as well as some goodness-of-fit diagnostics, see the Details sections of **estimate** and **check**.

EXAMPLES

The following examples show complete spec files.

Example 1 Specify and estimate a nonseasonal ARIMA model with a first difference and an MA parameter at lag 1, i.e., $(1-B)y_t = (1-\theta B)a_t$.

```
series { title = "Quarterly Grape Harvest" start = 1950.1
        period = 4
          data = (8997 9401 ... 11346) }
arima  { model = (0 1 1)}
estimate { }
```

Example 2 Specify and estimate the following seasonal ARIMA model for y_t, the logarithm of an original time series: $(1-\phi_1 B-\phi_2 B^2)(1-B)(1-B^{12})y_t = (1-\Theta_{12}B^{12})a_t$. Note that the start date in the **series** spec specifies a month, which sets the seasonal period to 12.

```
series { title = "Monthly sales" start = 1976.jan
          data = (138 128 ... 297) }
transform {function = log}
arima  { model = (2 1 0)(0 1 1)}
estimate { }
```

Example 3 Specify and estimate a regARIMA model with fixed seasonal effects, a trend constant, and the ARIMA (0 1 1) model for the regression errors. The model is then $(1-B)(y_t - \sum \beta_i M_{it} - c \cdot t) = (1-\theta B)a_t$, where the M_{it} are the fixed seasonal effect regression variables.

```
Series { Title = "Monthly Sales"  Start = 1976.jan
          Data = (138 128 ... 297) }
Transform { Function = log }
Regression { Variables= (seasonal const)}
Arima { Model = (0 1 1)}
Estimate { }
```

Example 4 Specify and estimate a model with one difference and an AR(2) operator with lag one missing; i.e., the model is $(1 - \phi_2 B^2)(1 - B)y_t = a_t$.

```
series{title = "Annual Olive Harvest" start = 1950
        data = (251 271 ... 240) }
arima{model  = ([2] 1 0)}
estimate{ }
```

Example 5 Specify and estimate a model with a trend constant and with regression errors z_t following an ARIMA model with one seasonal difference and a first order seasonal moving average, but no nonseasonal factor, i.e., $(1 - B^{12})z_t = (1 - \Theta B^{12})a_t$. Note that the seasonal period of the ARIMA factor must be given explicitly in the **model** argument, because, as there is only one ARIMA factor, it would otherwise be assumed to be nonseasonal.

```
series { title = "Monthly sales"  start = 1976.jan
          data = (138 128 ... 297) }
transform { function = log }
regression { variables = const }
arima { model  = (0 1 1)12}
estimate { }
```

Example 6 Specify and estimate a model including three ARIMA factors. The ARIMA model for the regression errors z_t is $(1 - \phi_1 B)(1 - \phi_3 B^3)(1 - B)z_t = (1 - \Theta B^{12})a_t$. The $1 - \phi_3 B^3$ operator might be used to account for quarterly autocorrelation since each quarter is comprised of three months. Note that only the period of the quarterly factor need be given.

```
series { title = "Monthly sales"  start = 1976.jan
          data = (138 128 ... 297) }
transform { function = log }
regression { variables = (const seasonal)}
arima { model  = (1 1 0)(1 0 0)3(0 0 1)}
estimate { }
```

Example 7 Specify and estimate a model with regression errors z_t following the "airline model", ARIMA $(0\ 1\ 1)(0\ 1\ 1)_{12}$, with the seasonal MA parameter fixed at 1.0. The model used for z_t is $(1 - B)(1 - B^{12})z_t = (1 - \theta B)(1 - 1.0B^{12})a_t$. The initial value of 0.1 used for θ is indicated by a missing value in the **ma** list. This model is actually equivalent to that used in Example 3, since it results from overdifferencing the model specified there by $1 - B^{12}$. (See Section 4.4 for a discussion of overdifferencing.)

```
series { title = "Monthly sales"  start = 1976.jan
          data = (138 128 ...  297) }
transform { function = log }
arima  { model = (0 1 1)(0 1 1)12
            ma = ( ,1.0f)}
estimate { }
```

AUTOMDL

DESCRIPTION

Specifies that the ARIMA part of the regARIMA model will be sought using an automatic model selection procedure similar to the one used by X-11-ARIMA/88 (see Dagum(1988)). The user can specify which types of models are to be fitted to the time series in the procedure and can change the thresholds for the selection criteria.

USAGE

```
automdl {mode = both
         method = best
         file = "my.mdl"
         fcstlim = 25.0
         bcstlim = 25.0
         qlim = 15.0
         overdiff = 0.99
         identify = all
         outofsample = yes
         print = (none autochoice)
         savelog = automodel
        }
```

ARGUMENTS

bcstlim Sets the acceptance threshold for the within-sample backcast error test when backcasts are specified by setting `mode=both`. The absolute average percentage error of the backcasted values is then tested against the threshold. For example, `bcstlim=25` sets this threshold to 25 percent. The value entered for this argument must not be less than zero, or greater than 100. The default for **bcstlim** is 20 percent.

fcstlim Sets the acceptance threshold for the within-sample forecast error test. The absolute average percentage error of the extrapolated values within the last three years of data must be less than this value if a model is to be accepted by the automatic modelling procedure. For example, `fcstlim=20` sets this threshold to 20 percent. The value entered for this argument must not be less than zero, or greater than 100. The default for **fcstlim** is 15 percent.

file Valid path and filename of the file containing the model types used in the automatic modelling procedure. The model types are specified using the same notation as in the **model** argument of the **arima** spec; see DETAILS below. If this argument is not used, the program will use the model types in the file `x12a.mdl`; this file must be in the current directory.

identify Determines how automatic identification of outliers (via the **outlier** spec) and/or automatic trading day regressor identification (via the **aictest** argument of the **regression** spec) are done within the automatic model selection procedure. If `identify = all`, automatic trading day regressor and/or automatic outlier identification (done in that order if both are specified) are done for each model specified in the automatic model file. If `identify = first`, automatic trading day regressor and/or automatic outlier identification are done the first model specified in the automatic model file. The decisions made for the first model specified are then used for the remaining models. The identification procedures are redone for the selected model, if the model selected is not the first. The default is `identify = first`.

method Specifies whether the automatic model selection procedure will select the first model which satistisfies the model selection criteria (`method = first`) or the estimated model with the lowest within-sample forecast error of all the model which satisfies the model selection criteria (`method = best`). The default is `method = first`.

mode Specifies that the program will attempt to find a satisfactory model within the set of candidate model types specified by the user, using the criteria developed by Statistics Canada for the `X-11-ARIMA` program and documented in Dagum(1988); see DETAILS. The fitted model chosen will be used to produce a year of forecasts if `mode = fcst`, or will produce a year of forecasts and backcasts if `mode = both`. The default is `mode = fcst`. The **forecast** spec can be used to override the number of forecasts and backcasts used to extend the series. The model will be chosen from the types read in from a file named in the **file** argument (specified below) or from the file `x12a.mdl` if the **file** argument is not used. Do not use both **arima** and **automdl** in the same specification file.

outofsample Determines which kind of forecast error is used for automatic model evaluation and selection. If `outofsample=yes`, out-of-sample forecasts errors are used; these are obtained by removing the data in the forecast period from the data set used to estimate the model and to produce one year of forecasts (for each of the last three years of data). If `outofsample=no`, within-sample forecasts errors are used. That is, the model parameter estimates for the full series are used to generate forecasts for each of the last three years of data. For conformity with `X-11-ARIMA`, outlier adjustments are made to the forecasted data that have been identified as outliers. The default is `outofsample=no`.

overdiff Sets the threshold for the sum of the MA parameter estimates in the overdifferencing test. The program computes the sum of the seasonal (for models with at least one seasonal difference) or non-seasonal (for models with at least one non-seasonal difference) MA parameter estimates. If the sum of the non-seasonal MA parameter estimates is greater than the limit set here, the automatic modelling procedure will reject the model because of overdifferencing. If the sum of the seasonal MA parameter estimates is greater than the limit set here, the automatic modelling procedure will print out a warning message suggesting the use of fixed seasonal effects in the regression spec, but will not reject the model. The default for this argument is 0.9; values entered for this argument should not be any lower than 0.9, and must not be greater than 1.

print The save option is not available for this spec. The tables available for output are listed in Table 6-1.

Table 6-1: Available Output Tables for Automdl

name	*default*	*brief*	*ext*	*description of table*
autoheader	+	+	·	header for the automatic modelling output
automodels	+	+	·	output for each model used in the automatic model procedure
autochoice	+	+	·	model choice of automatic model procedure

Name gives the name of each table for use with the **print** and **save** arguments.
Default indicates which tables are printed (+) or not printed (·) by default.
Brief indicates which tables are printed (+) or not printed (·) when the **brief** print level is specified. See section 5.1 for more information on print levels.
Ext gives the file extension used if the table is saved. A dot indicates that the table cannot be saved. The file extensions given can also be used as short names for the tables in the **print** and **save** arguments.

qlim Sets the acceptance threshold for the p-value of the Ljung-Box Q-statistic for model adequacy. The p-value associated with the fitted model's Q must be greater than this value for a model to be accepted by the automatic modelling procedure. For example, `qlim = 10` sets this threshold to 10 percent. The value entered for this argument must not be less than zero, or greater than 100. The default for **qlim** is 5 percent.

savelog Setting `savelog=automodel` or `savelog=amd` causes the result of the model selection procedure to be output to the session log file (see section 2.6 for more information on the log file).

DETAILS

The default settings for the automatic model selection procedure classify a model as acceptable if (1) the absolute average percentage error of the extrapolated values within the last three years of data is less than 15 percent, (2) the p-value associated with the fitted model's Ljung-Box Q-statistic testing the uncorrelatedness of the model's residuals must be greater than 5 percent, and (3) there are no signs of overdifferencing. There is an indication of overdifferencing if the sum of the non-seasonal MA parameter estimates (for models with at least one non-seasonal difference) is greater than 0.9. No model is selected when none of the models of the types in the model file is acceptable. Any of these criteria can be changed using the **fcstlim**, **qlim**, or **overdiff** arguments.

Note that if there is a **regression** spec in the spec file, the regression terms specified there will be used with all the ARIMA models evaluated by the automatic model selection procedure. The original series is transformed as specified in the **transform** spec.

The `X-11-ARIMA` program developed by Statistics Canada uses the following model types in its automatic modelling procedure: $(0,1,1)(0,1,1)_s$, $(0,1,2)(0,1,1)_s$, $(2,1,0)(0,1,1)_s$, $(0,2,2)(0,1,1)_s$, and $(2,1,2)(0,1,1)_s$, where s denotes the seasonal period (see Dagum(1988)). These are provided in the file `x12a.mdl` file provided with the program, but these model types cannot be used if a fixed seasonal effect is specified in the **regression** spec.

Each model type in the file designated by the **file** argument (or `x12a.mdl` if no **file** argument is given) is listed on a separate line, with `"X"` at the end of each line except the last.

Users can select one of the models to be a "default" model by marking the end of the line with an asterisk (`"*"`) rather than an `"X"`. This will allow the program to use the default regARIMA model to generate preadjustment factors based on the regressors specified by the user in the **regression** spec if a model is not selected by the automatic modeling procedure. No forecasts (or backcasts) are generated if none of the models are selected by the procedure.

An example using the `X-11-ARIMA` default models is given below:

```
(0 1 1)(0 1 1) *
(0 1 2)(0 1 1) X
(2 1 0)(0 1 1) X
(0 2 2)(0 1 1) X
(2 1 2)(0 1 1)
```

EXAMPLES

The following examples show complete spec files.

Example 1 Use the automatic ARIMA modelling procedure to select a model and use it to extend the series with one year of forecasts. Trading day and stable seasonal regression effects are to be included in the models. A default seasonal adjustment is to be performed.

```
series      { title = "Monthly sales"   start = 1976.jan
              data = (138 128  ...  297) }
regression  { variables = (td seasonal) }
automdl     { mode = fcst    file = "nosdiff.mdl"  }
estimate    {   }
x11         {  }
```

The contents of `nosdiff.mdl` are given below:

```
(1 1 0) X
(2 1 0) X
(0 1 1) *
(0 1 2) X
(2 1 2)
```

Example 2 Similar to Example 1, except that the forecast acceptance threshold is changed to 20 percent, the chi-square acceptance threshold is set to 10 percent, and the overdifferencing acceptance threshold is changed to 0.99. Also, the first acceptable model will be selected, and automatic outlier identification will be done for all the models listed in **nosdiff.mdl**.

```
series      { title = "Monthly sales"   start = 1976.jan
              data = (138 128  ...  297) }
regression  { variables = td }
automdl     { mode = fcst    file = "nosdiff.mdl"
              method = first   fcstlim = 20   qlim = 10
              overdiff = 0.99  identify = all }
outlier     {  }
estimate    {   }
x11         {  }
```

Example 3 The same as Example 1, except that out-of-sample forecast errors are used in the model identification and selection process.

```
series      { title = "Monthly sales"   start = 1976.jan
              data = (138 128  ...  297) }
regression  { variables = td }
automdl     { mode = fcst    file = "nosdiff.mdl"
              outofsample=yes  }
estimate    {   }
x11         {  }
```

CHECK

DESCRIPTION

Specification to produce statistics for diagnostic checking of residuals from the estimated model. Statistics available for diagnostic checking include the sample ACF and PACF of the residuals with associated standard errors, Ljung-Box Q-statistics and their p-values, summary statistics of the residuals, normality test statistics for the residuals, a spectral plot of the model residuals, and a histogram of the standardized residuals.

USAGE

```
check { maxlag = 24
        print = (none +histogram +acf)
        save = (acf)
        savelog = normalitytest
        }
```

ARGUMENTS

maxlag The number of lags requested for the residual sample ACF and PACF for both tables and plots. The default is 36 for monthly series, 12 for quarterly series.

print and **save** Table 6-2 gives the available output tables for this spec.

savelog The diagnostics available for output to the log file (see section 2.6) are listed on Table 6-3.

DETAILS

The **check** spec uses residuals from the estimated model. If the **estimate** spec is absent, the **check** spec forces estimation of the model (with default estimation options).

Under the null hypothesis that the model is correct, the Ljung-Box Q-statistics are asymptotically distributed as χ^2 with degrees of freedom equal to the number of lags used in computing them less the number of AR and MA parameters estimated. The degrees of freedom are shown on the output. Ignore the Q-statistics and p-values corresponding to zero degrees of freedom.

`X-12-ARIMA` produces two statistics that test the regARIMA model residuals for deviations from normality; both indicate kurtosis. One is Geary's a statistic, defined as in Geary (1936) and Gastwirth and Owens (1977):

Table 6-2: Available Output Tables for Check

name	*default*	*brief*	*ext*	*description of table*
acf	+	·	acf	autocorrelation function of residuals with standard errors, and Ljung-Box Q-statistics computed through each lag
acfplot	+	·	·	plot of residual autocorrelation function with ± 2 standard error limits
pacf	·	·	pcf	partial autocorrelation function of residuals with standard errors
pacfplot	·	·	·	plot of residual partial autocorrelation function with ± 2 standard error limits
acfsquared	+	·	ac2	autocorrelation function of squared residuals with standard errors, and Ljung-Box Q-statistics computed through each lag
acfsquaredplot	+	·	·	plot of squared residual autocorrelation function with ± 2 standard error limits
normalitytest	+	·	·	Geary's a and Kurtosis statistic tests for the normality of the model residuals
specresidual	·	·	spr	spectral plot of the regARIMA model residuals
histogram	+	·	·	histogram of standardized residuals and the following summary statistics of the residuals: minimum, maximum, median, standard deviation, and robust estimate of residual standard deviation (1.48 × the median absolute deviation)

Name gives the name of each table for use with the **print** and **save** arguments.
Default indicates which tables are printed (+) or not printed (·) by default.
Brief indicates which tables are printed (+) or not printed (·) when the **brief** print level is specified. See section 5.1 for more information on print levels.
Ext gives the file extension used if the table is saved. A dot indicates that the table cannot be saved. The file extensions given can also be used as short names for the tables in the **print** and **save** arguments.

Table 6-3: Available Log File Diagnostics for Check

name	*short*	*description of diagnostic*
normalitytest	nrm	Test results from the normality tests on the regARIMA model residuals (Kurtosis and Geary's a statistics)
ljungboxq	lbq	Significant lags for the Ljung-Box Q statistic

Name gives the name of each diagnostic for use with the **savelog** argument.
Short gives a short names for the diagonstics in the **savelog** arguments.

$$a = \frac{\frac{1}{n}\sum_{i=1}^{n} \mid X_i - \bar{X} \mid}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2}}$$

where $\bar{X}$ is the sample mean. The other is the sample kurtosis:

$$b_2 = \frac{m_4}{m_2^2} = \frac{n\sum_{i=1}^{n}(X_i - \bar{X})^4}{(\sum_{i=1}^{n}(X_i - \bar{X})^2)^2}$$

Properties of both are discussed in Section 5.14 of Snedecor and Cochran (1980).

A significant value of one of these statistics indicates that the standardized residuals do not follow a standard normal distribution. `X-12-ARIMA` tests for significance at the one percent level, from values given in tables from Pearson(1938) and Pearson and Hartley (1954). If the regARIMA model fits the data well, such lack of normality ordinarily causes no problems.

However, a significant value can occur because certain data effects are not captured well by the model. Sometimes these effects can be captured by additional or different regressors (e.g. trading day, holiday or outlier regressors). Thus, significant values can be used as a stimulus to reconsider what regressors to use.

There are other important effects that can cause a significant value, such as random variation of the coefficients or time-varying conditional variances, which cannot be represented by regARIMA models. These other effects cause the test statistics and forecast coverage intervals of X-12-ARIMA to have reduced reliability. Their presence is often indicated by significant values of the Ljung-Box Q-statistics of the squared residuals.

The number of lags for the ACF of the squared residual is set to be equal to seasonal period of the series (12 for monthly series, 4 for quarterly series). This value cannot be changed by the **maxlag** argument.

The use of fixed coefficients in the ARIMA model can invalidate the *DF* (degrees of freedom) values and therefore also the associated chi-square P-values in the Box Ljung Q-statistic output of **check**. This happens when the fixed values are actually estimated values from a previous model fitting. The P-values will have the expected (approximate) validity when a statistically insignificant coefficient has been fixed at the value zero.

EXAMPLES

The following examples show complete spec files.

Example 1 Print all available diagnostic checks of the residuals from the specified model. The sample autocorrelation and partial autocorrelation function of the residuals is computed through lag 36 (the default for monthly time series). The **check** spec forces model estimation to be performed (with default options) even though the **estimate** spec is not present.

```
series { title = "Monthly Retail Sales"  start = 1964.jan
         file = "sales1.dat" }
regression { variables = (td ao1967.jun
                          ls1971.jun easter[14]) }
arima { model = (0 1 1)(0 1 1) }
check { print = (all) }
```

Example 2 For the same series and model as in Example 1, produce all diagnostic checking statistics except the except the printed table and plot of the residual PACF. The residual ACF is computed through lag 24.

```
Series { Title = "Monthly Retail Sales"    Start = 1964.jan
         File = "SALES1.DAT" }
Regression { Variables = (TD AO1967.jun
                         LS1971.jun Easter[14]) }
Arima { Model = (0 1 1)(0 1 1) }
Check { Print = (All -PACF -PACFplot)  Maxlag = 24 }
```

COMPOSITE

DESCRIPTION

This spec is used as part of the procedure for obtaining both indirect and direct adjustments of a composite series. For obtaining composite adjustments, it is one of the required spec files referenced in a metafile. Previous spec files in the metafile must define the component series and how they are combined to form the composite (see the **comptype** and **compwt** arguments of the **series** spec). This spec is used in place of the **series** spec.

The user can specify a title for the composite adjustment, a name for the composite series, which tables are to be printed or stored, and which line-printer plots are to be produced from the indirect adjustment.

USAGE

```
composite {title = "Total one family housing starts"
           name = "hs1ft"
           decimals = 2
           modelspan = (1985.Jan,)
           spectrumstart = 1985.Jan
           print = (brief +indtest)
           save = (indseasonal)
           savelog = (indtest)
          }
```

ARGUMENTS

decimals Specifies the number of decimals that will appear in the seasonal adjustment tables of the main output file. This value must be an integer between 0 and 5, inclusive (for example, `decimals=3`). The default number of decimals is zero.

modelspan Specifies the span (data interval) of the composite time series that is to be used to determine all regARIMA model coefficients. This argument can be utilized when, for example, the user does not want data early in the series to affect the forecasts, or, alternatively, data late in the series to affect regression estimates used for preadjustment before seasonal adjustment. The **modelspan** argument has two values, the start and end date of the desired span. A missing value defaults to the corresponding start or end date of the composite series being analyzed. For example, for monthly data, the statement `modelspan=(1968.1, )` causes whatever regARIMA model is specified in other specs to be estimated from the time series data starting in January, 1968 and ending at the end date of the analysis span. A comma is necessary if either the start or end date is missing. The start and end dates of the model span must both lie within the time span of the composite series, and the start date must precede the end date.

Another end date specification, with the form *0.per*, is available to set the ending date of **modelspan** always to be the most recent occurrence of a specific calendar month (quarter for quarterly data) in the span of data analyzed, where *per* denotes the calendar month (quarter). If the span of data considered ends in a month other than December, `modelspan=(,0.dec)` will cause the model parameters to stay fixed at the values obtained from data ending in the next-to-final calendar year of the span.

name The name of the composite time series. The name must be enclosed in quotes and may contain up to 8 characters. It will be printed as a label on every page of printed output.

print and **save** The optional output tables available for the direct and indirect seasonal adjustments generated by this spec are given in Table 6-4 on the next few pages.

savelog The diagnostics available for output to the log file (see section 2.6) are listed in Table 6-5.

spectrumstart The starting date of the span of data to be used to estimate the spectrum of the composite time series, and the spectra of the direct and indirect seasonally adjusted series and modified irregular series. This date must be in the format `spectrumstart=`*year.seasonal period*. This can be used to determine if there are residual trading day or seasonal effects in the adjusted data from, say, the last ten years. Residual effects can occur when seasonal or trading day patterns are evolving. The default starting date for the spectral plots of monthly series is set to be eight years from the end of the composite series, unless the composite series is less than eight years long, in which case, it is set to the starting date of the composite series. For quarterly series, the default starting date is set to the starting date of the composite series. Example: `spectrumstart=1987.Jan`.

title A title describing the composite time series. The title must be enclosed in quotes and may contain up to 79 characters. It will be printed above the data in the output.

RARELY USED ARGUMENTS

diffspectrum If `diffspectrum=no`, the spectrum of the (transformed) original series or seasonally adjusted series is calculated. The default (`diffspectrum=yes`) produces the spectrum of the month-to-month (quarter-to-quarter) differences of these series.

Table 6-4: Available Output Tables for Composite

name	*default*	*brief*	*ext*	*description of table*
compositesrs	+	+	cms	aggregated time series data, with associated dates
compositeplot	·	·	·	plot of the aggregate series
outlieradjcomposite	·	·	oac	aggregated time series data, adjusted for outliers.
header	+	+	·	header for indirect seasonal adjustment
indtest	+	+	·	test for adequacy of composite adjustment
indunmodsi	+	·	id8	final unmodified si-ratios (differences) for the indirect adjustment
indftestd8	+	·	·	F-test for stable and moving seasonality for the indirect seasonal adjustment
indreplacsi	+	·	·	final replacement values for extreme si-ratios (differences) for the indirect adjustment
indmovseasrat	+	·	·	moving seasonality ratios for the indirect seasonal adjustment
indseasonal	+	+	isf	final seasonal factors for the indirect seasonal adjustment
indseasonaldiff	+	+	isd	final seasonal difference for the indirect seasonal adjustment (only for pseudo-additive seasonal adjustment)
indseasonalplot	·	·	·	indirect seasonal factor plots, grouped by month or quarter
indseasadj	+	+	isa	final indirect seasonally adjusted series
indadjsatot	+	+	iaa	final indirect seasonally adjusted series, with yearly totals adjusted to match the original series
indsadjround	+	+	irn	rounded indirect final seasonally adjusted series
indresidualseasf	+	·	·	F-test for residual seasonality
indseasadjplot	·	·	·	plot of the final indirect seasonally adjusted series
indtrend	+	·	itn	final trend-cycle for the indirect adjustment
indtrendplot	·	·	·	plot of the final trend-cycle from the indirect seasonal adjustment
indirregular	+	·	iir	final irregular component for the indirect adjustment
indirregularplot	·	·	·	plot of the final irregular component from the indirect seasonal adjustment
orivsindadjplot	·	·	·	plot of the aggregate series with the indirect seasonally adjusted series
indmodoriginal	·	·	ie1	original series modified for extreme values from the indirect seasonal adjustment
indmodsadj	·	·	ie2	seasonally adjusted series modified for extreme values from the indirect seasonal adjustment
indmodirr	·	·	ie3	irregular component modified for extreme values from the indirect seasonal adjustment
indyrtotals	·	·	·	ratio of yearly totals of the original series and the indirect seasonally adjusted series

Name gives the name of each table for use with the **print** and **save** arguments.
Default indicates which tables are printed (+) or not printed (·) by default.
Brief indicates which tables are printed (+) or not printed (·) when the **brief** print level is specified. See section 5.1 for more information on print levels.
Ext gives the file extension used if the table is saved. A dot indicates that the table cannot be saved. The file extensions given can also be used as short names for the tables in the **print** and **save** arguments.

Table 6-4: Available Output Tables for Composite (continued)

name	*default*	*brief*	*ext*	*description of table*
origchanges	+	·	ie5	percent changes (differences) in the original series
indsachange	+	·	ie6	percent changes (differences) in the indirect seasonally adjusted series
indrevsachanges	+	·	i6a	percent changes for indirect seasonally adjusted series with revised yearly totals
indrndsachanges	+	·	i6r	percent changes for rounded indirect seasonally adjusted series
indtrendchange	+	·	ie7	percent changes (differences) in the indirect final trend component
indrobustsa	·	·	iee	final indirect seasonally adjusted series modified for extreme values
indmcdmovavg	·	·	if1	MCD moving average of the final indirect seasonally adjusted series
indx11diag	+	+	·	summary of seasonal adjustment diagnostics for the indirect seasonal adjustment
indqstat	+	+	·	quality control statistics for the indirect seasonal adjustment
ratioplotorig	·	·	·	month-to-month (or quarter-to-quarter) ratio plots of the original series
ratioplotindsa	·	·	·	month-to-month (or quarter-to-quarter) ratio plots of the indirect seasonally adjusted series
speccomposite	+	·	·	spectral plot of first-differenced aggregate series
specindsa	+	+	·	spectral plot of the first-differenced indirect seasonally adjusted series
specindirr	+	+	·	spectral plot of outlier-modified irregular series from the indirect seasonal adjustment
outlieradjcomposite	·	·	oac	aggregated time series data, adjusted for outliers.

Name gives the name of each table for use with the **print** and **save** arguments.
Default indicates which tables are printed (+) or not printed (·) by default.
Brief indicates which tables are printed (+) or not printed (·) when the **brief** print level is specified. See section 5.1 for more information on print levels.
Ext gives the file extension used if the table is saved. A dot indicates that the table cannot be saved. The file extensions given can also be used as short names for the tables in the **print** and **save** arguments.

saveprecision The number of decimals stored when saving a table to a separate file with the **save** argument. The default value of **saveprecision** is 15. Example: `saveprecision=10`.

spectrumtype The type of spectral estimate used in the spectral plots produced by `X-12-ARIMA`. If `spectrumtype = periodogram`, the periodogram of the series is calculated and plotted. The default (`spectrumtype=arspec`) produces an autoregressive model spectrum of the series.

Table 6-5: Available Log File Diagnostics for Composite

name	*short*	*description of diagnostic*
indtest	itt	test for adequacy of composite adjustment
indm1	im1	M1 Quality Control Statistic from indirect adjustment
indm2	im2	M2 Quality Control Statistic from indirect adjustment
indm3	im3	M3 Quality Control Statistic from indirect adjustment
indm4	im4	M4 Quality Control Statistic from indirect adjustment
indm5	im5	M5 Quality Control Statistic from indirect adjustment
indm6	im6	M6 Quality Control Statistic from indirect adjustment
indm7	im7	M7 Quality Control Statistic from indirect adjustment
indm8	im8	M8 Quality Control Statistic from indirect adjustment
indm9	im9	M9 Quality Control Statistic from indirect adjustment
indm10	imt	M10 Quality Control Statistic from indirect adjustment
indm11	ime	M11 Quality Control Statistic from indirect adjustment
indq	iq	Overall index of the acceptablility of the indirect seasonal adjustment
indq2	iq2	Indirect Q statistic computed without the M2 Quality Control statistic
indmovingseasratio	isr	Moving seasonality ratio from indirect adjustment
indicratio	iir	$\bar{I}/\bar{C}$ ratio from indirect adjustment
indfstabled8	id8	F-test for stable seasonality, performed on the final SI-ratios from indirect adjustment
indmovingseasf	isf	F-test for moving seasonality from indirect adjustment
indidseasonal	iid	Identifiable seasonality test result for indirect adjustment
peaks	spk	Visually significant peaks in spectra from indirect adjustment

Name gives the name of each diagnostic for use with the **savelog** argument.
Short gives a short names for the diagonstics in the **savelog** arguments.

yr2000 If `yr2000=yes`, a "century cutoff" for 2-digit years from data stored in "X-11 formats" is set at 1945. Years 00-45 are interpreted as 20xx, and years 46-99 are interpreted as 19xx. This is the default for the program.

If `yr2000=no`, the program assumes all 2-digit years fall in the 20th century and will convert them to 4-digit years accordingly.

Note: this option is set here to affect program behavior when files are read in other specs (such as the **transform** and **x11regression** spec).

DETAILS

An input specifications file with the **composite** spec can only be used in conjunction with spec files for component series which together define a composite series. The names of these other spec files must be listed in a metafile in which the name of this spec file appears last. The `comptype` argument of the **series** spec of each component series controls how the components are combined to form the final aggregate (composite) series. (See Section 2 for examples of how to run metafiles with `X-12-ARIMA`).

A composite adjustment run with this metafile produces an indirect seasonal adjustment of the composite series as well as a direct seasonal adjustment. The indirect adjustment is the combination specified by the **comptype** of the components, each adjusted or not adjusted according to the prescriptions of their spec files. The direct adjustment is done as requested in the spec file of the composite spec. To control the output for the direct seasonal adjustment, use the **print** and **save** arguments of the **x11** spec.

To include an unadjusted series as a component of the indirect seasonal adjustment of the aggregate series, specifiy the summary measures option by setting `type = summary` in the **x11** spec of this component.

As is mentioned in section 2.7, the **-c** flag is used only to restrict a composite seasonal adjustment run done with an input metafile (**-m**). In a composite run, X-12-ARIMA usually seasonally adjusts a set of component series. When **-c** is invoked, the seasonal adjustment and modelling options specified in the input spec files for the component series are ignored; the component series are only used to form the composite series. This option is useful when identifying a regARIMA model for the composite series.

Although none of the tables of seasonal adjustment diagnostics produced in this spec can be saved to its own file, specifying the seasonal adjustment diagnostic summary option with the **-s** flag at runtime allows the user to store information from the composite analysis into a diagnostic summary file (with the file extension **.xdg**). In addition, the **savelog** argument can write selected diagnostics into the log file for a given run (with the file extension **.log**). For more information, see section 2.

If a sliding spans analysis of the direct and indirect adjustments is desired, the sliding spans analysis option must be specified for each of the component series. If the seasonal filter length is not the same for each component, then the user must use the **length** argument of the **slidingspans** spec to ensure that the spans stored for the component series are of the same length.

When a revisions history analysis of the seasonally adjusted series is specified for a composite seasonal adjustment, the revisions of both the direct and indirect seasonal adjustments of the composite series are produced. The revisions history analysis must be specified for each of the component series.

The spectra of the differenced adjusted series and of the irregulars are automatically searched for peaks at the seasonal and trading day frequencies. A warning message is printed out if peaks are found, and the plot where the peak was found is printed out. When the restricted output (the **-n** flag) option is used, a message is printed suggesting that the user rerun the program to obtain the spectral plot, but the plot is not included in the main output.

EXAMPLES

The following examples illustrating all the steps of a composite adjustment show complete spec files.

Step 1 A spec file must be created for each of the component series. In this example, we process each of the components (Northeast, Midwest, South and West 1-family housing starts), using a simple sum to form the composite. An example of the spec file for the Northeast series (stored in `cne1hs.spc`), which is seasonally adjusted using 3x9 seasonal filters, is given below:

```
series { title="NORTHEAST ONE-FAMILY Housing Starts"
        file="cne1hs.ori"  name="CNE1HS"  format="2R"
        comptype=add  }
x11 {  seasonalma=(s3x9)
     title=(
      "Component for Composite Adjustment"
      "of Total U.S. 1-Family Housing Starts") }
```

The seasonal adjustment of CNE1HS produced by this spec file will be an addend in the calculation of the indirect seasonal adjustment of the composite series.

A spec file for a component series that is not seasonally adjusted is given below:

```
series { title="West ONE-FAMILY Housing Starts"
        file="cwt1hs.ori"  name="CWT1HS"  format="2R"
        comptype=add  }
x11 {  type=summary  }
```

This will cause the unadjusted series stored in cwt1hs.ori to be an addend in the calculation of the indirect seasonal adjustment of the composite series.

Step 2 Create a spec file for the indirect adjustment of total one-family housing starts, the sums of four regional series. The direct seasonal adjustment of the series will be multiplicative and will use a 3x9 seasonal moving average. Both the seasonal factors from the direct adjustment and the implied factors from the indirect adjustment will be saved. The spec file (stored in `c1fths.spc`) appears below:

```
composite { title="TOTAL ONE-FAMILY Housing Starts"
           name="C1FTHS" save=(indseasonal) }
x11 {  seasonalma=(s3x9)
    title="Composite adj. of 1-Family housing starts"
     save=(D10) }
```

Step 3 Create a metafile for the component and composite series. This metafile, stored in `hs1ftot.mta` appears below:

```
cne1hs
cmw1hs
cso1hs
cwt1hs
c1fths
```

Note that the spec file for the composite series is listed last.

Step 4 To run `X-12-ARIMA` for this example, enter the following:

x12a -m hs1ftot

and press the return (enter) key.

ESTIMATE

DESCRIPTION

Estimates the regARIMA model specified by the **regression** and **arima** specs. Allows the setting of various estimation options. Estimation output includes point estimates and standard errors for all estimated AR, MA, and regression parameters; the maximum likelihood estimate of the variance σ^2; t-statistics for individual regression parameters; χ^2-statistics for assessing the joint significance of the parameters associated with certain regression effects (if included in the model); and likelihood based model selection statistics (if the exact likelihood function is used). The regression effects for which χ^2-statistics are produced include stable seasonal effects, trading-day effects, and the set of user-defined regression effects.

USAGE

estimate {**tol** = `1.0e−5`
maxiter = `200`
exact = `arma`
outofsample = `yes`
print = `(none +model +estimates +lkstats)`
save = `(model)`
savelog = `(aic bic)` }

ARGUMENTS

exact Specifies use of exact or conditional likelihood for estimation, likelihood evaluation, and forecasting. The default is `exact = arma`, which uses the likelihood function that is exact for both AR and MA parameters. Other options are: `exact = ma`, use the likelihood function that is exact for MA, but conditional for AR parameters; and `exact = none`, use the likelihood function that is conditional for both AR and MA parameters.

maxiter The maximum number allowed of ARMA iterations (nonlinear iterations for estimating the AR and MA parameters). For models with regression variables, this limit applies to the total number of ARMA iterations over all IGLS iterations. For models without regression variables, this is the maximum number of iterations allowed for the single set of ARMA iterations. The default is `maxiter = 200`.

outofsample Determines the kind of forecast error used in calculating the average magnitude of forecast errors over the last three years, a diagnostic statistic. If `outofsample=yes`, out-of-sample forecasts errors are used; these are obtained by removing the data in the forecast period from the data set used to estimate the model and produce one year of forecasts (for each of the last three years of data). If `outofsample=no`, within-sample forecasts errors are used. That is, the model parameter estimates for the full series are used to generate forecasts for each of the last three years of data. The default is `outofsample=no`.

print and **save** Table 6-6 gives the available output tables for this spec.

savelog The diagnostics available for output to the log file (see section 2.6) are listed in Table 6-7.

tol Convergence tolerance for the nonlinear estimation. Absolute changes in the log-likelihood are compared to **tol** to check convergence of the estimation iterations. For models with regression variables, **tol** is used to check convergence of the IGLS iterations (where the regression parameters are reestimated for each new set of AR and MA parameters), see Otto, Bell and Burman (1987a, 1987b). For models without regression variables there are no IGLS iterations, and **tol** is then used to check convergence of the nonlinear iterations used to estimate the AR and MA parameters. The default value is `tol = 1.0e-5`.

RARELY USED OPTIONS

file Name of the file containing the model settings of a previous `X-12-ARIMA` run. Such a file is produced by setting `save=model` or `save=mdl` in this spec. The filename must be enclosed in quotes. If the file is not in the current directory, the path must also be given. If the **file** argument is used, the **model**, **ma**, and **ar** arguments of the **arima** spec and the **variables**, **user**, and **b** arguments of the **regression** spec cannot be used.

fix Specifies whether certain coefficients found in the model file specified in the **file** argument are to be held fixed instead of being used as initializing values for further estimation. If `fix = all`, both the regression and ARMA parameter estimates will be held fixed at their values in the model file. If `fix = arma`, only ARMA parameter estimates will be held fixed at their model file values. If `fix = none`, none of the parameter estimates will be held fixed. The default is `fix = nochange`, which will preserve coefficient values specified as fixed in the model file and allow restimation of all other coefficients.

Table 6-6: Available Output Tables for Estimate

name	*default*	*brief*	*ext*	*description of table*
options	+	·	·	header for the estimation options
iterations	·	·	itr	detailed output for estimation iterations, including log-likelihood values and parameters, and counts of function evaluations and iterations
iterationerrors	·	·	·	error messages for estimation iterations, including failure to converge
model	+	+	mdl	if used with the **print** argument, this controls printing of a short description of the model; if used with the **save** argument, this creates a file containing **regression** and **arima** specs corresponding to the model, with the estimation results used to specify initial values for the ARMA parameters
regcmatrix	·	·	rcm	correlation matrix of regression parameter estimates if used with the **print** argument; covariance matrix of same if used with the **save** argument
armacmatrix	·	·	acm	correlation matrix of ARMA parameter estimates if used with the **print** argument; covariance matrix of same if used with the **save** argument
estimates	+	+	est	regression and ARMA parameter estimates, with standard errors
averagefcsterr	+	·	·	average magnitude of forecast errors over each of the last three years of data.
lkstats	+	+	lks	log-likelihood at final parameter estimates and, if `exact = arma` is used (default option), corresponding model selection criteria (AIC, AICC, Hannan-Quinn, BIC)
lformulas	·	·	·	formulas for computing the log-likelihood and model selection criteria
roots	·	·	rts	roots of the autoregressive and moving average operators in the estimated model
regressioneffects	·	·	ref	$\mathbf{X}\hat{\beta}$, matrix of regression variables multiplied by the vector of estimated regression coefficients
residuals	·	·	rsd	model residuals with associated dates or observation numbers

Name gives the name of each table for use with the **print** and **save** arguments.
Default indicates which tables are printed (+) or not printed (·) by default.
Brief indicates which tables are printed (+) or not printed (·) when the **brief** print level is specified. See section 5.1 for more information on print levels.
Ext gives the file extension used if the table is saved. A dot indicates that the table cannot be saved. The file extensions given can also be used as short names for the tables in the **print** and **save** arguments.

DETAILS

The inference results provided by `X-12-ARIMA` are asymptotically valid (approximately correct for sufficiently long time series) under "standard" assumptions—see Section 3.5. The likelihood based model selection statistics are provided only if the exact likelihood function is used. See Section 4.5 for comments on the use of model selection statistics.

Table 6-7: Available Log File Diagnostics for Estimate

name	*short*	*description of diagnostic*
aic	aic	Akaike's Information Criterion (AIC)
aicc	acc	Akaike's Information Criterion (AIC) adjusted for the length of the series
bic	bic	Baysean Information Criterion (BIC)
hannanquinn	hq	Hannan-Quinn Information Criterion
averagefcsterr	afc	Average forecast error over the last three years of data

Name gives the name of each diagnostic for use with the **savelog** argument.
Short gives a short names for the diagonstics in the **savelog** arguments.

If the estimation iterations converge, `X-12-ARIMA` prints a message to this effect, and then displays the estimation results. If the iterations fail to converge, `X-12-ARIMA` prints a message indicating this and then displays the parameter values at the last iteration. These values should not be used as parameter estimates. Instead, the program should be rerun, possibly starting at the parameter values obtained when the iterations terminated. Potential causes of convergence problems and suggested remedies are discussed in Section 4.

Tol should not be set either "too large" or "too small". Setting **tol** too large can result in estimates too far from the true MLEs, while setting **tol** too small can result in an unnecessarily large number of iterations or lead to a false impression of the precision of the results. What is too large or too small a value for **tol** depends on the problem; the default value of **tol**$= 10^{-5}$ is offered as a reasonable compromise. Setting **tol** to a number less than machine precision for a double precision number (approximately 10^{-14} for PCs and Sun4 computers) results in an error, but values for **tol** that even begin to approach machine precision are certainly too small.

For models with regression variables, a second convergence tolerance is needed to determine convergence of the ARMA iterations within each IGLS iteration. This tolerance is set by the program to $100 \times$ **tol** for the first two IGLS iterations, after which it is reset to **tol**. (Since relatively large changes can be made to the regression parameters in the initial IGLS iterations, it is not worth determining the ARMA parameters within **tol** at the start.) Thus, when **tol** takes on its default value of 10^{-5}, the ARMA convergence tolerance is 10^{-3} for the first two IGLS iterations, and thereafter it is 10^{-5} (= **tol**). Also, for models with regression variables, a limit is needed for the maximum number of ARMA iterations allowed within each IGLS iteration. This limit is set to 40.

If the ARMA iterations fail to converge on a particular IGLS iteration, this is generally not a problem. The program will continue with the next IGLS iteration, and its ARMA iterations may very well converge. In fact, all that is necessary for overall convergence is that the ARMA iterations of the *last* IGLS iteration converge, and that the IGLS iterations themselves converge to the tolerance **tol** within **maxiter** total

Table 6-8: Example of ARMA Roots Output

```
Roots of ARIMA Model
 Root                    Real  Imaginary    Modulus  Frequency
 -----------------------------------------------------------
 Nonseasonal AR
  Root   1            -0.6784     0.8817     1.1125     0.3544
  Root   2            -0.6784    -0.8817     1.1125    -0.3544
 Nonseasonal MA
  Root   1            -7.4107     0.0000     7.4107     0.5000
 Seasonal MA
  Root   1             1.5583     0.0000     1.5583     0.0000
```

ARMA iterations.

Setting `print=roots` produces a table of roots of all the AR and MA operators of the estimated model. In addition to the roots, the table provides the modulus (magnitude) and frequency (on $[-.5, .5]$) of each root. Roots with modulus greater than one lie outside the unit circle, corresponding to stationary AR or invertible MA operators. (See Section 4.4.) AR roots on or inside the unit circle (modulus ≤ 1) should occur only when the likelihood function is defined conditionally for AR parameters (`exact = ma` or `exact = none`). MA roots inside the unit circle (modulus < 1) will never occur, since invertibility is enforced in the estimation. MA roots on the unit circle (modulus $= 1$) can be estimated within round-off error, or can occur in an MA operator all of whose parameters are specified as fixed during estimation.

In sample output shown in Table 6-8, the nonseasonal AR(2) polynomial has a pair of complex conjugate roots (zeros), $z = x \pm iy$, with $x = -.6784$ and $y = .8817$, whose modulus (magnitude) is $r = (x^2 + y^2)^{1/2} = 1.1125$. Because this number is close to unity (1.000), it is worthwhile to examine the nonnegative frequency of the root, i.e. the number $\lambda \geq 0$ such that $z = re^{\pm i2\pi\lambda}$ to determine if the series may contain a deterministic periodic component. The reasoning behind this is as follows. Whenever a modeled time series has a periodic component $f(t)$ with period $1/\lambda$, i.e. $f(t + 1/\lambda) = f(t)$, then an estimated AR polynomial of sufficiently high order is likely to have a root near $e^{\pm i2\pi\lambda}$ (unless the differencing operators have $e^{\pm i2\pi\lambda}$ as a root). There are theoretical results that help to explain why this happens, but a heuristic explanation is that for the simplest functions with this period,

$$f(t) = A\cos(2\pi\lambda t + c),$$

the AR(2) polynomial $\phi(B) = \left(1 - e^{-i2\pi\lambda}B\right)\left(1 - e^{i2\pi\lambda}B\right)$, whose roots are $e^{\pm i2\pi\lambda}$, has the property that

$$\phi(B) f(t) = 0.$$

Thus this AR(2) factor can perfectly predict $f(t)$ from $f(t-1)$ and $f(t-2)$. Fitting a model with an AR operator of order 2 or higher will tend to make the AR parameters take on values so that $\phi(B) f(t) = 0$. (An AR(1) polynomial suffices when $e^{i2\pi\lambda}$ is real, i.e. when $\lambda = 0, 1/2$.) . Hence the occurrance of an AR root with modulus $r \doteq 1$ suggests the presence of an approximately periodic component in the time series.

For monthly series, the frequencies of seasonal effects are $\lambda = 1/12$, $2/12$, $3/12$, ... , $6/12$ (equivalent to 0.0833, 0.1666, 0.2500, ..., 0.5000, repectively). The frequency $\lambda = 0$ is associated with trend movements, and the frequency $\lambda = 0.3482$ with trading day effects. Note that the frequency 0.3544 of the nonseasonal AR roots in the table above is very close to the trading day frequency. In fact, the time series whose model produced the table has a strong trading day component, and the automatic modelling procedure added the AR(2) factor to account for it, because there were no trading day regressors in the `regression` spec.

In the MA polynomials, near unit roots with seasonal or trend frequencies usually indicate that the MA polynomials have one or more roots in common with the differencing or seasonal differencing polynomials. The presence of such a common factor $\kappa(B)$ indicates that the time series has deterministic trend or seasonal components. More specifically, in the notation of equation (3) of Section 3 (but ignoring regressors), it means that there is a function $f(t)$ satisfying $\kappa(B) f(t) = 0$ such that the time series y_t can be modeled as

$$\phi(B)\Phi(B^s)\left\{\frac{(1-B)^d(1-B^s)}{\kappa(B)^D}\right\} y_t = f(t) + \left\{\frac{\theta(B)\Theta(B^s)}{\kappa(B)}\right\} a_t.$$

In the example table above, the model's seasonal moving average polynomial is $\Theta(B^{12}) = 1 - \Theta B^{12}$ with $\Theta = 0.6417$ so the root is $1/\Theta = 1.5583$ (the root of $\Theta(z) = 1 - \Theta z$). Experience suggests that $1/\Theta$ generally needs to be 1.10 or less before it might be appropriate to replace the model with one having only fixed seasonal effects (i.e. a model with $D = 0$ and with. `variables=seasonals` in the `regression` spec).

If the nonseasonal MA polynomial has a root close to the number 1 (i.e. modulus near 1, frequency near 0), it often means that there is *overdifferencing*. That is, one should consider an alternative model with differencing order d and nonseasonal MA order q both smaller by one, and a trend constant (i.e. $f(t) = C$ above with `variables=const` in the `regression` spec) should be included in the alternative model if it has a significant t-statistic.

The use of fixed coefficients in the ARIMA model or the regression model of the **regression** or **x11regression** specs can invalidate the *AIC, AICC, Hannan Quinn,* and *BIC* model selection statistics in the output. This happens when the fixed values are actually estimated values from a previous model fitting. The *P*-values will have the expected (approximate) validity when a statistically insignificant coefficient has been fixed at the value zero.

EXAMPLES

The following examples show complete spec files.

Example 1 Estimate by generalized least squares the regression coefficients in the model $(1-B)(y_t - \sum_{i=1}^{11} \beta_i M_{it}) = (1-\theta B)a_t$, where the M_{it} are regression variables for monthly fixed seasonal effects. The MA parameter θ is held fixed at the value 0.25. Model residuals are saved in a file in the current directory with the same name as the spec file, but with the extension `.rsd`.

```
series { title = "Monthly Sales" start = 1976.1
         data = (138 128 ... 297) }
regression { variables = seasonal }
arima { model = (0,1,1)   ma = (0.25f) }
estimate { save = residuals }
```

Example 2 Estimate the seasonal model $(1-\phi B)(1-B)(1-B^{12})z_t = (1-\Theta B^{12})a_t$, with **tol** set to 10^{-4}, a looser convergence criterion than the default, and decrease the maximum number of iterations allowed to 100. Since there are no regression parameters in the model, both **tol** and **maxiter** apply to the single set of nonlinear ARMA iterations used to estimate ϕ and Θ. The likelihood function used in parameter estimation is exact for MA and conditional for AR parameters. The **print** argument specifies that the likelihood and parameter values are printed for each iteration and, following the last iteration, the roots of the estimated AR and MA operators are printed. The **save** argument will save the final regARIMA model into a file.

```
series { title = "Monthly Inventory" start = 1978.12
         data = (1209 834 ... 1002) }
transform { function = log }
regression { variables = (td ao1999.01)  }
arima { model = (1,1,0)(0,1,1)}
estimate { tol = 1e-4  maxiter = 100  exact = ma  save = mdl
           print = (iterations roots)  }
```

Example 3 Same as Example 2, except the regARIMA model estimates saved in example 2 are used in this run via the **file** argmument. All parameter estimates are fixed to the values stored in the model file.

```
series { title = "Monthly Inventory" start = 1978.12
         data = (1209 834 ... 1002) }
transform { function = log }
estimate { file = "Inven.mdl"
           fix = all  }
```

Example 4 Same as Example 3, except that three additional data values are available and we wish to have the program determine if any of them are outliers. The ending date of the data span in Examples 2 and 3 is December, 1999. The regARIMA model parameters are to be kept fixed at the values obtained from this data span, which were stored by Example 2.

```
series { title = "Monthly Inventory" start = 1978.12
         data = (1209 834 ... 1002 1425 901 1375) }
transform { function = log }
estimate { file = "Inven.mdl"
           fix = all  }
outlier { span=(2000.01,)  }
```

FORECAST

DESCRIPTION

Specification to forecast and/or backcast the time series given in the **series** spec using the estimated model. The output contains point forecasts and forecast standard errors for the transformed series, and point forecasts and prediction intervals for the original series.

USAGE

```
forecast {    maxlead = 12
              maxback = 12
          probability = .95
              exclude = 10
                print = (none +original +transformed)
                 save = (variances) }
```

ARGUMENTS

exclude Number of observations excluded from the end of the series (or from the end of the span specified by the **span** argument of the **series** spec, if present) before forecasting. The default is to start forecasting from the end of the series (or span), i.e., `exclude = 0`.

maxback Number of backcasts produced. The default is 0 and the maximum is 60. (The limit of 60 can be changed—see Section 2.8.) **Note:** Backcasts are not produced for time series that are more than 15 years long, or if the starting date specified in the **modelspan** argument of the **series** spec is not the same as the starting date of the analysis span specified in the **span** argument of the **series** spec.

maxlead Number of forecasts produced. The default is 12 and the maximum is 60. (The limit of 60 can be changed—see Section 2.8.)

print and **save** The optional output tables are listed on Table 6-9.

probability Coverage probability for prediction intervals, assuming normality. The default is `probability=.95`, in which case prediction intervals on the transformed scale are *point forecast* $\pm$ 1.96 $\times$ *forecast standard error*.

Table 6-9: **Available Output Tables for Forecast**

name	*default*	*brief*	*ext*	*description of table*
transformed	+	·	ftr	point forecasts on the transformed scale, with corresponding forecast standard errors
variances	·	·	fvr	forecast error variances on the transformed scale, showing the contributions of the error assuming the model is completely known (stochastic variance) and the error due to estimating any regression parameters (error in estimating AR and MA parameters is ignored)
forecasts	+	·	fct	point forecasts on the original scale, along with upper and lower prediction interval limits

Name gives the name of each table for use with the **print** and **save** arguments.
Default indicates which tables are printed (+) or not printed (·) by default.
Brief indicates which tables are printed (+) or not printed (·) when the **brief** print level is specified. See section 5.1 for more information on print levels.
Ext gives the file extension used if the table is saved. A dot indicates that the table cannot be saved. The file extensions given can also be used as short names for the tables in the **print** and **save** arguments.

DETAILS

Forecasting is done with the estimated (or evaluated) model. If the **estimate** spec is not present, the **forecast** spec will force estimation (with default options) to be performed before forecasting. The model used for forecasting is that specified by the **regression** and **arima** specs. If the **outlier** spec is present, the model is augmented by additional regression variables for any automatically identified outliers. Detected outliers can affect forecasts indirectly, through their effect on model parameter estimates, as well as directly, when outliers found near the end of the series affect the computation of the forecasts.

If the model includes one or more moving average operators then the forecasts will depend on the residuals from the estimated model. The **exact** argument of the **estimate** spec determines whether these are computed corresponding to exact likelihood (the default) or a form of conditional likelihood.

Forecast standard errors include an adjustment for error arising from estimation of any regression parameters in the model, but do not include an adjustment for error arising from estimation of AR and MA parameters.

If the model contains user-defined regression variables, values for these must be provided for all time points in the forecast period.

Prediction intervals on the transformed scale are defined as

$$point\ forecast\ \pm\ K\ \times\ forecast\ standard\ error,$$

where K denotes the standard error multiplier (from a table of the normal distribution) corresponding to the specified coverage probability. Point forecasts and prediction interval limits on the original scale are obtained by inverse transformation of those on the transformed scale, allowing for both transformation (Box-Cox or logistic) and prior-adjustment factors (including the length-of-month or length-of-quarter adjustment implied if `variables = td` is included in the **regression** spec). If the **transform** spec includes user-defined prior-adjustment factors, these must be provided through the forecast period for the results to be inverse transformed. If they are not provided through the forecast period, then they will be assumed to be 1 in the forecast period. In this case, effects of the user-defined adjustments on the forecasts will not be (and cannot be) undone.

A reason for using **exclude** > 0 is to produce forecasts for some time points whose data are withheld for purposes of evaluating the forecast performance of the model. `X-12-ARIMA` facilitates such comparisons by printing actual forecast errors (*observation* − *point forecast*) at all time points in the forecast period for which corresponding (transformed) observed data exist. Setting **exclude** > 0 produces *within-sample* comparisons, since the data that are withheld from forecasting are not withheld from model estimation. More realistic *out-of-sample* forecast comparisons are produced by withholding data from both model estimation and forecasting, which can be accomplished by using the **span** argument of the **series** spec. (See Example 4.)

Whenever forecasts and/or backcasts are generated in an `X-12-ARIMA` run in which seasonal adjustment is performed, they are appended to the original series, and the seasonal adjustment procedures are applied to the forecast and/or backcast extended series. If a seasonal adjustment is specified in a run in which a regARIMA model is used but the **forecast** spec is not, one year of forecasts are generated from the model. The only way to specify a seasonal adjustment without forecast extension is to set `maxlead = 0`.

If preadjustments for regARIMA estimated trading day, outlier, holiday or user-defined regression effects are prior adjusted from the original series, they are also adjusted out of the forecasts and backcasts.

Warning: if seasonal adjustment is specified by the **x11** spec, **exclude** cannot be used to exclude observations from the end of the series. In case it is used, **exclude** will be set to zero and a warning message will be printed.

EXAMPLES

The following examples show complete spec files.

Example 1 Forecast up through 12 steps ahead from the end of a monthly time series, and produce 95 percent prediction intervals (point forecast $\pm$ 1.96 standard errors). These are all default options. Though the **estimate** spec is absent, the presence of the **forecast** spec forces model estimation with default estimation options. The point forecasts and prediction interval limits for the transformed series are exponentiated and then multiplied by $m_t/\bar{m}$ (to undo the length-of-month adjustment produced by `variables = td` in the **regression** spec) to convert them back to the original scale.

```
SERIES { TITLE = "Monthly sales"  START = 1976.JAN
         DATA = (138 128 ... 297) }
TRANSFORM { FUNCTION = LOG}
REGRESSION { VARIABLES = TD }
ARIMA { MODEL = (0 1 1)(0 1 1)12 }
FORECAST { }
```

Example 2 Forecast up through 24 steps ahead from the end of the same series used in Example 1. Since the **outlier** spec is present, the estimated model used in forecasting will include any AO or LS outliers detected, in addition to the trading-day variables specified by the **regression** spec.

```
Series { Title = "Monthly Sales"  Start = 1976.jan
         Data = (138 128 ... 297) }
Transform { Function = Log}
Regression { Variables = Td }
Arima { Model = (0 1 1)(0 1 1)12 }
Estimate { }
Outlier { }
Forecast { Maxlead = 24 }
```

Example 3 Exclude 10 data points and forecast up through 15 steps ahead. The entire time series is used for parameter estimation, including the ten data points excluded at the end of the series when forecasting. For these last 10 data points, the *within-sample* forecast errors will be printed. At each forecast lead the prediction interval limits are obtained as the point forecasts plus or minus 1.645 times the corresponding forecast standard error, which corresponds to the requested 90 percent coverage probability.

```
series { title = "Monthly sales"  start = 1976.jan
         data = (138 128 ... 297) }
transform { function = log }
regression { variables = td }
arima { model = (0 1 1)(0 1 1)12 }
estimate { }
forecast { maxlead = 15
           probability = .90
           exclude = 10 }
```

Example 4 The series ends in March, 1992, but the last 24 observations are excluded from model estimation by using a **span** argument in the **series** spec. Then, using the model with these parameter estimates, the last 24 observations are forecast from March, 1990, the end of the span. The *out-of-sample* errors in forecasting the last 24 observations will be printed out. (Contrast this with Example 3.)

```
series { title = "Monthly sales"  start = 1976.jan
         data = (138 128 ... 297)
         span = ( ,1990.mar) }
transform { function = log}
regression { variables = td }
arima { model = (0 1 1)(0 1 1)12 }
estimate { }
forecast { maxlead = 24 }
```

Example 5 Forecast up through 12 months ahead from the end of a monthly time series, and produce 95 percent prediction intervals (point forecast $\pm$ 1.96 standard errors). These are all default options. Also produce 12 backcasts of the series, and perform a default multiplicative seasonal adjustment of the forecast- and backcast-extended original series, prior-adjusted for trading day effects.

```
series { title = "monthly sales"  start = 1980.jan
         file = "ussales.dat"     }
transform { function = log}
regression { variables = td }
arima { model = (0 1 1)(0 1 1)12 }
forecast {   maxback=12  }
x11{   }
```

HISTORY

DESCRIPTION

Optional spec for requesting a sequence of runs from a sequence of truncated versions of the time series for the purpose of creating historical records of (i) revisions from initial (concurrent or projected) seasonal adjustments, (ii) out-of-sample forecast errors, and (iii) likelihood statistics. The user can specify the beginning date of the historical record and the choice of records (i) - (iii). If forecast errors are chosen, the user can specify a vector of forecast leads. **Warning:** Generating the history analysis can substantially increase the program's run time.

USAGE

```
history {estimates = ( sadj   fcst   trend )
         sadjlags = (1,2,3,12)
         trendlags = (1,2,3)
         target = final
         start = 1975.jan
         fstep = ( 1   2 )
         fixmdl = no
         fixreg = outlier
         endtable = 1990.Jan
         print = ( all   -revvalsa )
         save = ( sar   trr   fcsterrors )
         savelog = ( aveabsrevsa   aveabsrevtrend )
        }
```

ARGUMENTS

endtable Specifies the final date of the output tables of the revisions history analysis of seasonal adjustment and trend estimates and their period-to-period changes. This can be used to ensure that the revisions history analysis summary statistics are based only on final (or nearly final) seasonal adjustments or trends. If **endtable** is not assigned a value, it is set to the date of the observation immediately before the end of the series or to a value one greater than the largest lag specified in **sadjlags** or **trendlags**. This option has no effect on the historical analyses of forecasts and likelihood estimates. Example: `endtable=1990.jun`.

estimates Determines which estimates from the regARIMA modelling and/or the X-11 seasonal adjustment will be analyzed in the history analysis. Example: `estimates=(sadj aic)`. The default is the seasonally adjusted series (`sadj`). Table 6-10 gives a description of the available estimates.

Table 6-10: Choices Available for the estimates Argument

name	*description of option*
sadj	Final seasonally adjusted series (and indirect seasonally adjusted series, if composite seasonal adjustment is performed.
sadjchng	Month-to-month (or quarter-to-quarter) changes in the final seasonally adjusted series.
trend	Final Henderson trend component.
trendchng	Month-to-month (or quarter-to-quarter) changes in the final Henderson trend component.
seasonal	Final and projected seasonal factors.
aic	AICCs and maximum log likelihoods for the regARIMA model.
fcst	Forecasts and evolving mean square forecast errors generated from the regARIMA model. **Warning:** This option can be used only when forecasts are produced, see the **forecast** spec.

fixmdl Specifies whether the regARIMA model will be reestimated during the history analysis. If `fixmdl=yes`, the ARIMA parameters and regression coefficients of the regARIMA model will be fixed throughout the analysis at the values estimated from the entire series (or model span, if one is specified via the **modelspan** argument). If `fixmdl=no`, the regARIMA model parameters will be restimated each time the end point of the data is changed. The default is `fixmdl=no`. This argument is ignored if no regARIMA model is fit to the series.

fixreg Specifies the fixing of the coefficients of a regressor group, either within a regARIMA model or an irregular component regression. These coefficients will be fixed at the values obtained from the model span (implicit or explicitly) indicated in the series or composite spec. All other coefficients will be reestimated for each history span. Trading day (**td**), holiday (**holiday**), outlier (**outlier**), or other user-defined (**user**) regression effects can be fixed. This argument is ignored if neither a regARIMA model nor an irregular component regression model is fit to the series, or if `fixmdl=yes`.

fstep Specifies a vector of up to four (4) forecast leads that will be analyzed in the history analysis of forecast errors. Example: `fstep=(1 2 12)` will produce an error analysis for the 1-step, 2-step, and 12-step ahead forecasts. The default is (1 12) for monthly series or (1 4) for quarterly series. **Warning:** The values given in this vector cannot exceed the specified value of the **maxlead** argument of the **forecast** spec, or be less than one.

print and **save** The tables available for output are listed on Table 6-11.

sadjlags Specifies a vector of up to 5 revision lags (each greater than zero) that will be analyzed in the revisions analysis of lagged seasonal adjustments. The calculated revisions for these revision lags will be those of the seasonal adjustments obtained using this many observations beyond the time point of the adjustment. That is, for each value $revisionlag_i$ given in **sadjlags**, series values through time $t + revisionlag_i$ will be used to

Table 6-11: Available Output Tables for History

name	*default*	*brief*	*ext*	*description of table*
header	+	+	·	header for history analysis
outlierhistory	+	+	rot	record of outliers removed and kept for the revisions history (printed only if automatic outlier identification is used)
sfilterhistory	·	·	sfh	record of seasonal filter selection for each observation in the revisions history (printed only if automatic seasonal filter selection is used)
sarevisions	+	·	sar	revision from concurrent to most recent estimate of the seasonally adjusted data
sasummary	+	+	·	summary statistics for seasonal adjustment revisions
saestimates	·	·	sae	concurrent and most recent estimate of the seasonally adjusted data
chngrevisions	+	·	chr	revision from concurrent to most recent estimate of the month-to-month (or quarter-to-quarter) changes in the seasonally adjusted data
chngsummary	+	+	chs	summary statistics for revisions in the month-to-month (or quarter-to-quarter) changes in the seasonally adjusted data
chngestimates	·	·	che	concurrent and most recent estimate of the month-to-month (or quarter-to-quarter) changes in the seasonally adjusted data
indsarevisions	+	·	iar	revision from concurrent to most recent estimate of the indirect seasonally adjusted series
indsasummary	+	+	·	summary statistics for indirect seasonal adjustment revisions
indsaestimates	·	·	iae	concurrent and most recent estimate of the indirect seasonally adjusted data
trendrevisions	+	·	trr	revision from concurrent to most recent estimate of the trend component
trendsummary	+	+	·	summary statistics for trend component revisions
trendestimates	·	·	tre	concurrent and most recent estimate of the trend component
trendchngrevisions	+	·	tcr	revision from concurrent to most recent estimate of the month-to-month (or quarter-to-quarter) changes in the trend component
trendchngsummary	+	+	·	summary statistics for revisions in the month-to-month (or quarter-to-quarter) changes in the trend component
trendchngestimates	·	·	tce	concurrent and most recent estimate of the month-to-month (or quarter-to-quarter) changes in the trend component

Name gives the name of each table for use with the **print** and **save** arguments.
Default indicates which tables are printed (+) or not printed (·) by default.
Brief indicates which tables are printed (+) or not printed (·) when the **brief** print level is specified. See section 5.1 for more information on print levels.
Ext gives the file extension used if the table is saved. A dot indicates that the table cannot be saved. The file extensions given can also be used as short names for the tables in the **print** and **save** arguments.

Table 6-11: Available Output Tables for History (continued)

name	*default*	*brief*	*ext*	*description of table*
sfrevisions	+	·	sfr	revision from concurrent to most recent estimate of the seasonal factors, as well as the revision from concurrent to most recent estimate of the projected seasonal factors
sfsummary	+	+	·	summary statistics for seasonal factors
sfestimates	·	·	sfe	concurrent, projected, and most recent estimate of the seasonal factors
lkhdhistory	+	+	lkh	Log-Likelihood and AICC Values for each observation in the revisions history.
fcsterrors	+	+	fce	accumulating sums of squared forecast errors and evolving mean square forecast errors for each observation in the revisions history.
fcsthistory	·	·	fch	concurrent forecasts and forecast errors for each observation in the revisions history.

Name gives the name of each table for use with the **print** and **save** arguments.
Default indicates which tables are printed (+) or not printed (·) by default.
Brief indicates which tables are printed (+) or not printed (·) when the **brief** print level is specified. See section 5.1 for more information on print levels.
Ext gives the file extension used if the table is saved. A dot indicates that the table cannot be saved. The file extensions given can also be used as short names for the tables in the **print** and **save** arguments.

obtain the adjustment for time t whose revision will be calculated. For more information, see DETAILS.

This option is meaningful only if the revisions history of the seasonally adjusted series or month-to-month (quarter-to-quarter) changes in the seasonally adjusted series is specified in the **estimates** argument. The default is no analysis of revisions of lagged seasonal adjustments.

savelog The diagnostics available for output to the log file (see section 2.6) are listed on Table 6-12.

start Specifies the starting date of the revisions history analysis. If this argument is not used, its default setting depends on the length of the longest seasonal filter used, provided that a seasonal adjustment is being performed (if there is no conflict with the requirement that sixty earlier observations be available when a regARIMA model is estimated and `fixmdl=no`, the default for kwfixmdl). The default starting date is six (6) years after the start of the series, if the longest filter is either a a 3x3 or stable filter, eight (8) years for a 3x5 filter, and twelve (12) years for a 3x9 filter. If no seasonal adjustment is done, the default is 8 years after the start of the series. Example: `start=1990.jun`.

target Specifies whether the deviation from the concurrent estimate or the deviation from the final estimate defines the revisions of the seasonal adjustments and trends calculated at the lags specified in **sadjlags** or **trendlags**. The default is `target=final`; the alternative is `target=concurrent`.

Table 6-12: Available Log File Diagnostics for History

name	*short*	*description of diagnostic*
aveabsrevsa	asa	average absolute revision of the seasonally adjusted series
aveabsrevchng	ach	average absolute revision of the month-to-month (or quarter-to-quarter) changes in the seasonally adjusted data
aveabsrevindsa	iaa	average absolute revision of the indirect seasonally adjusted series
aveabsrevtrend	atr	average absolute revision of the final trend component
aveabsrevtrendchng	atc	average absolute revision of the month-to-month (or quarter-to-quarter) changes in the trend component
aveabsrevsf	asf	average absolute revision of the final seasonal factors
aveabsrevsfproj	asp	average absolute revision of the projected seasonal factors

Name gives the name of each diagnostic for use with the **savelog** argument.
Short gives a short names for the diagonstics in the **savelog** arguments.

trendlags Similar to **sadjlags**, this argument prescribes which lags will be used in the revisions history of the lagged trend components. Up to 5 integer lags greater than zero can be specified.

This option is meaningful only if the revisions history of the final trend component or month-to-month (quarter-to-quarter) changes in the final trend component is specified in the **estimates** argument. The default is no analysis of revisions lagged trend estimates.

RARELY USED ARGUMENTS

fixx11reg Specifies whether the irregular component regression model specified in the **x11regression** spec will be reestimated during the history analysis. If `fixx11reg=yes`, the regression coefficients for the irregular component regression model are fixed throughout the analysis at the values estimated from the entire series. If `fixx11reg=no`, the irregular component regression model parameters will be restimated each time the end point of the history interval is advanced. The default is `fixx11reg=no`. This argument is ignored if no irregular component regression model is specified.

outlier Specifies that automatic outlier detection is to be performed whenever the regARIMA model is reestimated during the revisions history analysis. This argument has no effect if the **outlier** spec is not used.

If `outlier=keep`, all outliers automatically identified using the full series are kept in the regARIMA model during the revisions history analysis. The coefficients estimating the effects of these outliers are reestimated whenever the other regARIMA model parameters are reestimated. No additional outliers are automatically identified and estimated. This is the default setting.

If `outlier=remove`, those outlier regressors that were added to the regression part of the regARIMA model when automatic outlier identification was performed on the full series are removed from the regARIMA model during the revisions history analysis. Consequently, their effects are not estimated and removed from the series. This option gives the user a way to investigate the consequences of not doing automatic outlier idenfication.

If `outlier=auto`, among outliers automatically identified for the full series, only those that fall in the time period up to **outlierwin** observations before the starting date of the revisions history analysis are automatically included in the regARIMA model. In each run of the estimation procedure with a truncated version of the original series, automatic outlier identification is performed only for the last **outlierwin** +1 observations. An outlier that is identified is used for the current run, but is only retained for the subsequent runs of the historical analysis if it is at least **outlierwin** observations from the end of the subsequent span of data being analyzed.

outlierwin Specifies how many observations before the end of each span will be used for outlier identification during the revisions history analysis. The default is 12 for monthly series and 4 for quarterly series. This argument has an effect only if the **outlier** spec is used, and if `outlier=auto` in the **history** spec.

refresh Specifies which of two sets of initializing values is used for the regARIMA model parameter estimation. If `refresh=yes`, the parameter estimates from the last model evaluation are used as starting values for the current regARIMA model estimation done during the revisions history. If `refresh=no`, then the initial values of the regARIMA model parameters will be set to the estimates derived from the entire series. The default is `refresh=no`.

x11outlier Specifies whether the AO outlier identification will be performed during the history analysis for all irregular component regressions that result from the **x11regression** spec. If `x11outlier=yes`, AO outlier identification will be performed for each of the history runs. Those AO outlier regressors that were added to the irregular component regression model when automatic AO outlier identification was done for the full series are removed from the irregular component regression model prior to the history runs. If `x11outlier=no`, then the AO outlier regressors automatically identified are kept for each of the history runs. If the date of an outlier detected for the complete span of data does not occur in the data span of one of the history runs, the outlier will be dropped from the model for that run. The coefficients estimating the effects of these AO outliers are reestimated whenever the other irregular component regression model parameters are reestimated. However, no additional AO outliers are automatically identified and estimated. This option is ignored if the **x11regression** spec is not used, if the selection of the **aictest** argument results in the program not estimating an irregular component regression model, or if the **sigma** argument is used in the **x11regression** spec. The default is `x11outlier=yes`.

DETAILS

When this option is invoked, X-12-ARIMA generates revisions between the initial estimate and the most recent estimate, for several quantities derived from seasonally adjusting a time series (see Table 6-10). X-12-ARIMA can also generate historical out-of-sample forecast errors and likelihood statistics derived from regARIMA model estimation. These revisions and historical values are obtained as follows.

For a given series y_t where $t = 1, \ldots, T$, we define $A_{t|n}$ to be the seasonal adjustment of y_t calculated from the series $y_1, y_2, \ldots, y_n$, where $t \leq n \leq T$. The concurrent seasonal adjustment of observation t is $A_{t|t}$ and the most recent or "final" adjustment of observation t is $A_{t|T}$. The percent revision of the seasonally adjusted series is defined to be

$$R_t = \frac{A_{t|T} - A_{t|t}}{A_{t|t}}$$

and this is what is reported by the program. The revisions of the trend component and of seasonal factors derived from multiplicative or log-additive seasonal adjustment are also reported as percent revisions.

With additive seasonal adjustments, R_t is calculated the same way if all values $A_{t|t}$ have the same sign (the analagous statement holds for trends). Otherwise, differences are calculated:

$$R_t = A_{t|T} - A_{t|t}$$

In the additive adjustment case, revisions of seasonal factors are always calculated as differences, $S_{t|T} - S_{t|t}$, or, with projected seasonal factors $S_{t|T} - S_{t|t^*}$, where t^* denotes ending date of the series used to obtain the projected factor for month t.

Let $C_{t|n}$ denote the month-to-month (or quarter-to-quarter) change in the seasonally adjusted series at time t calculated from the series $y_1, y_2, \ldots, y_n$, or

$$C_{t|n} = \frac{A_{t|n} - A_{t-1|n}}{A_{t-1|n}}$$

The revision of these changes is defined to be

$$R_t = C_{t|T} - C_{t|t}$$

Revisions for the month-to-month changes in the trend component are computed in the same manner.

The **sadjlags** and **trendlags** arguments produce an analysis of the revisions history for different lags past the concurrent observation. The target for this revisions analysis depends on the value of the **target** argument. Table 6-13 shows how the lagged revisions are calculated for for the different values of **target**.

Table 6-13: Revision Measure Calculated for Revison lag analysis

Estimate	*Concurrent Target*	*Final Target*
Seasonally Adjusted Series	$(A_{t\|t+lag_i} - A_{t\|t})/A_{t\|t}$	$(A_{t\|T} - A_{t\|t+lag_i})/A_{t\|t+lag_i}$
Final Trend Component	$(T_{t\|t+lag_i} - T_{t\|t})/T_{t\|t}$	$(T_{t\|T} - T_{t\|t+lag_i})/T_{t\|t+lag_i}$
Change in Seasonally Adjusted Series (or Trend)	$C_{t\|t+lag_i} - C_{t\|t}$	$C_{t\|T} - C_{t\|t+lag_i}$

Estimate gives the estimate from the seasonal adjustment.
Concurrent Target gives the formula for the lagged revision history where the target is assumed to be the concurrent estimate.
Final Target gives the formula for the lagged revision history where the target is assumed to be the final estimate.
$A_{t|i}$ is the value of the seasonally adjusted series at time t calculated from the series up to time i.
$T_{t|i}$ is the value of the trend component at time t calculated from the series up to time i.
$C_{t|i}$ is the value of the change in the seasonally adjusted series at time t calculated for the series up to time i.

If lags corresponding to one and two years (12 and 24 for monthly data, 4 and 8 for quarterly data) are included in **sadjlags**, then the revision between the seasonal adjustment calculated one year after time t and the adjustment 2 years after time t is also calculated, or:

$$RY_t = \frac{A_{t|t+24} - A_{t|t+12}}{A_{t|t+12}}$$

for monthly series. This is done only for the seasonally adjusted series and the month-to-month (quarter-to-quarter) change of the seasonally adjusted series.

The analysis of the lagged revisions can give a useful picture of the behaviour of the revisions over time. Using the concurrent estimate as the target shows how much a given adjustment changes as you add more data; using the final estimate as the target shows how quickly a given estimate converges to the final value.

Another motivation for the **sadjlags** and **trendlags** options is the fact that concurrent estimates are often based on preliminary data for the current month (or quarter). If the final data for the month are not available until two additional months have passed, then it would be appropriate to set `sadjlags = 2` in order to study the revisions to the adjustment based on the final datum for each month. For trends, there is the additional motivation that concurrent trend estimates are often unstable. For this reason, some analysts wait until several subsequent months of data are available for trend estimation before examining the `X-12-ARIMA` trend for a recent month. For an analyst who waits three months, `trendlags = 3` will provide the revisions of the trend estimates of interest.

When a revision history analysis of the seasonally adjusted series is specified for a composite seasonal adjustment, the revisions of both the direct and indirect seasonally adjusted series are produced. The revision history analysis must also be specified for each of the component series, even for those component series that are not seasonally adjusted, see the Examples section of the description of the **composite** spec. The revision history of the indirect seasonally adjusted series (**sadj** in Table 6-10) is the only revision history available for indirect seasonal adjustments.

If the automatic seasonal filter selection option is used, the program will redo the choice of seasonal filter each time the data span is changed in the revision history analysis. If the seasonal filter should change in the course of the analysis, a warning message will be printed out, and a table of the seasonal filter lengths chosen for each data span will be printed out.

The starting date for the forecast revisions depends on the values given for **fstep**. The starting date for a history of n-step-ahead forecast errors is n periods after the starting date of the history analysis. **Example:** if `fstep = (1 12)` and `start = 1992.jan`, the history for the 1-step and 12-step ahead forecasts will start in February of 1992 and January of 1993, respectively.

In some situations, the program automatically switches to using fixed model coefficients for the history analyis. This happens when the start of the revisions history analysis (which can be set by the user with the **start** argument) causes some truncated data span to have fewer than sixty observations for regARIMA model estimation, either because of the series length or a **span** or **modelspan** argument value (in the **series** or **composite** spec). In this case, the coefficients (ARIMA and regression) of the regARIMA model will be held fixed throughout the analysis at the values estimated from the entire series (or model span, if one is specified).

Fixing of the coefficients will also occur for every truncated data span that contains data later than the ending date specified in a **modelspan** argument. In particular, in the extreme case, when the ending date of the model span is earlier than the starting date of the history analysis, the coefficients of the regARIMA model will be fixed throughout the history analysis.

Regression models from the **x11regression** spec are treated similarly. For example, their coefficients are fixed if some truncated data span has fewer than sixty observations because of a date assigned to the **span** argument of **x11regression**.

If an outlier specified by the user occurs in the period after the starting date of the revision history, that outlier will be dropped from the model at the start of the revision history analysis. It will be re-introduced into the regARIMA model when enough data have been added for the outlier variable to be defined. User-defined regressors are treated in the same way.

EXAMPLES

The following examples show complete spec files.

Example 1 A multiplicative monthly seasonal adjustment is to be performed with 3x9 seasonal moving averages for all months without using regARIMA model forecasts, backcasts, or regression outlier adjustments. A revision history of just the seasonally adjusted series will be performed (remember, this is the default history) for all data, after a startup period of twelve years (because 3x9 seasonal factors are used), with an additional analysis on the estimates made 2 periods after the concurrent observation.

```
Series {  Title = "Sales Of Livestock"  Start = 1967.1
          File = "cattle.ori"  }
X11 {  SeasonalMA = S3X9   }
History  {  sadjlags = 2  }
```

Example 2 Utilize a seasonal ARIMA model with regression variables for outlier and level shift preadjustment. The specified regression variables are a constant, trading day effects, and two level shifts, one in May 1972 and one in Sept. 1976. The ARIMA part of the model is (0,1,2)(1,1,0)12. Generate a history of the 1-step ahead forecast errors. Start the analysis in January of 1975; this means the first 1-step ahead forecast error in the analysis is for February of 1975.

```
series     {  title = "Exports of Leather goods"
              start = 1969.jul  file = "expleth.dat"  }
regression {  variables = (const td ls1972.may ls1976.oct)  }
arima      {  model = (0 1 2)(1 1 0)  }
estimate   {  }
history    {  estimates = fcst  fstep = 1  start=1975.jan  }
```

Example 3 Using the same regARIMA model and data as in Example 2, generate a history of the 1-step and 12-step ahead forecast errors. Start the history in January of 1975. Save the history to a file. In this file, zeros will be printed for the estimates where the 12-step ahead forecast errors are not defined (in this case, February to December of 1975) in order to maintain a uniform format for the file.

```
series     {  title = "Exports of Leather goods"
              start = 1969.jul
              file = "expleth.dat"  }
regression {  variables = (const td ls1972.may ls1976.oct)  }
arima      {  model = (0 1 2)(1 1 0)  }
estimate   {  }
history    {  estimates = fcst  save = r6  start = 1975.jan  }
```

Example 4 A multiplicative monthly seasonal adjustment is to be performed, with 3x3 seasonal moving averages, using regARIMA model forecasts to extend the series. The regARIMA model will be fit to the data up to the last December available to the series. A revision history of the seasonally adjusted series and the trend component will be calculated starting after the sixth year of the series, with the regARIMA model parameters restimated every December. Also, the history of the seasonal adjustment revisions of this series is integrated into the revision history calculation of the indirect seasonal adjustment of the composite series of which this series is a component. (The spec file for the composite series in the metafile must include an appropriate history spec, see Example 5.)

```
series {  title = "Housing Starts in the Mid-West"
          start = 1967.1
          file = "hsmwtot.ori"
          modelspan = (,0.Dec)
          comptype=add
}
regression {  variables = td  }
arima { model = (0 1 2)(0 1 1)  }
x11 {  seasonalMA = S3X3   }
history  {  estimates = (sadj trend)  }
```

Example 5 A composite monthly seasonal adjustment is to be performed with 3x3 seasonal moving averages for all months using regARIMA model forecasts to extend the composite series. The regARIMA model will be fit to the data up to the last December available to the series. A revision history of both the direct and indirect seasonally adjusted series and the trend component from the direct seasonal adjustment will be performed, with the regARIMA model parameters restimated every December. The percent revisions for each of the estimates will be stored in separate files.

```
composite{ title = "Total Housing Starts in the US"
           modelspan = (,0.Dec)
}
regression {  variables = td  }
arima { model = (0 1 1)(0 1 1)  }
x11 {  seasonalMA = S3X3   }
history  {  estimates = (sadj trend)
            save = (sar iar trr)  }
```

IDENTIFY

DESCRIPTION

Specification to produce tables and line printer plots of sample ACFs and PACFs for identifying the ARIMA part of a regARIMA model. Sample ACFs and PACFs are produced for all combinations of the nonseasonal and seasonal differences of the data specified by the **diff** and **sdiff** arguments. If the **regression** spec is present, the ACFs and PACFs are calculated for the specified differences of a series of regression residuals. If the **regression** spec is not present, the ACFs and PACFs are calculated for the specified differences of the original data.

USAGE

identify{ **diff** = `(0, 1)`
sdiff = `(0, 1)`
maxlag = `36`
print = `(none +acf +acfplot +pacf +pacfplot)` }

ARGUMENTS

diff Orders of nonseasonal differencing specified. The value 0 specifies no differencing, the value 1 specifies one nonseasonal difference $(1 - B)$, the value 2 specifies two nonseasonal differences $(1 - B)^2$, etc. The specified ACFs and PACFs will be produced for *all* orders of nonseasonal differencing specified, in combination with *all* orders of seasonal differencing specified in **sdiff**. The default is `diff=(0)`.

maxlag The number of lags specified for the ACFs and PACFs for both tables and plots. The default is 36 for monthly series, 12 for quarterly series.

print and **save** The tables available for output are listed on Table 6-14.

sdiff Orders of seasonal differencing specified. The value 0 specifies no seasonal differencing, the value 1 specifies one seasonal difference $(1 - B^s)$, etc. The specified ACFs and PACFs will be produced for *all* orders of seasonal differencing specified, in combination with *all* orders of nonseasonal differencing specified in **diff**. The default is `sdiff=(0)`.

Table 6-14: Available Output Tables for Identify

name	*default*	*brief*	*ext*	*description of table*
acf	+	+	iac	sample autocorrelation function(s), with standard errors and Ljung-Box Q-statistics for each lag
acfplot	+	·	·	line printer plot of sample autocorrelation function(s) with ± 2 standard error limits shown on the plot
pacf	+	+	ipc	sample partial autocorrelation function(s) with standard errors for each lag
pacfplot	+	·	·	line printer plot of sample partial autocorrelation function(s) with ± 2 standard error limits shown on the plot
regcoefficients	·	·	·	Regression coefficients removed from the transformed series before ACFs and PACFs were generated.

Name gives the name of each table for use with the **print** and **save** arguments.
Default indicates which tables are printed (+) or not printed (·) by default.
Brief indicates which tables are printed (+) or not printed (·) when the **brief** print level is specified. See section 5.1 for more information on print levels.
Ext gives the file extension used if the table is saved. A dot indicates that the table cannot be saved. The file extensions given can also be used as short names for the tables in the **print** and **save** arguments.

DETAILS

If the **regression** spec is present, the program differences the series (after processing by the **transform** spec) and the regression variables using the maximum order of differencing specified by the **diff** and **sdiff** arguments. The differenced series is then regressed on the differenced regression variables. The resulting regression coefficients ($\tilde{\beta}_i$) are then used to calculate *undifferenced* regression effects ($\sum_i \tilde{\beta}_i x_{it}$), which are then subtracted from the *undifferenced* data (y_t) to produce a time series of undifferenced regression errors ($\tilde{z}_t = y_t - \sum_i \tilde{\beta}_i x_{it}$). This regression error time series and its differences as specified by **diff** and **sdiff**, are then used to produce the ACFs and PACFs.

There is one exception to the above. If a constant term is specified in the **regression** spec (`variables = (const ...)`), it is included when the regression is done but not when the regression effects are subtracted from the series. See Section 3.4 for more discussion.

ACFs and PACFs are produced for all combinations of nonseasonal and seasonal differencing orders specified in **diff** and **sdiff**. For example, if `diff = (0, 1)` and `sdiff = 1` are specified, then ACFs and PACFs are computed for $(1-B^s)\tilde{z}_t$ and $(1-B)(1-B^s)\tilde{z}_t$, where $\tilde{z}_t$ is the series of regression errors, as discussed above, and s is the seasonal period specified in the **series** spec. If `diff = (0, 1, 2)` and `sdiff = (0, 1)` are specified, then ACFs and PACFs are computed for six series: $\tilde{z}_t$, $(1-B)\tilde{z}_t$, $(1-B)^2\tilde{z}_t$, $(1-B^s)\tilde{z}_t$, $(1-B)(1-B^s)\tilde{z}_t$, and $(1-B)^2(1-B^s)\tilde{z}_t$.

If both the **identify** and **estimate** specs are present, the **identify** spec is processed first. Note that the **identify** spec uses information from the **regression** spec, if present, but ignores the **arima** spec.

Users should make sure that differencing does not produce a singularity among the regression variables, including any user-defined regression variables, as singularities will cause a fatal error. One way this would arise is if **sdiff** was assigned a positive value (e.g., 1), while `variables = (seasonal)` was included in the **regression** spec.

If the number of lags requested for ACFs and PACFs equals or exceeds the length of the series (or the differenced series), the ACF and PACF will be computed only through the highest lag possible.

EXAMPLES

The following examples show complete spec files.

Example 1 Produce ACF tables useful for identifying the degree of differencing required for the monthly series $y_t = \log(Y_t)$, where Y_t is the original data input in the **series** spec. The ACFs are calculated for y_t, $(1 - B)y_t$, $(1 - B^{12})y_t$, and $(1 - B)(1 - B^{12})y_t$. The **regression** spec is absent so no regression effects are removed. ACFs are calculated through lag 36, the default for a monthly time series.

```
series { title = "Monthly Sales"  start = 1976.jan
         data = (138 128 ... 297) }
transform { function = log }
identify { diff = (0, 1)
         sdiff = (0, 1)
         print = (none +acf) }
```

Example 2 Remove fixed seasonal effects before computing sample ACFs and sample PACFs. The **regression** spec includes a trend constant as well as the fixed seasonal variables. The **identify** spec removes the fixed seasonal effects by regressing $(1 - B)y_t$ on the differenced regression variables $(1 - B)x_{it}$, and computing undifferenced regression residuals $\tilde{z}_t = y_t - \sum_{i=2}^{12} \tilde{\beta}_i x_{it}$ (not subtracting out the trend constant term $\tilde{\beta}_1 x_{1t}$). It then computes ACFs and PACFs of $\tilde{z}_t$ and $(1 - B)\tilde{z}_t$. The constant term allows for an overall nonzero mean in $(1 - B)y_t$, so it is a linear trend constant, i.e., $x_{1t} = t$.

```
SERIES     { TITLE = "MONTHLY SALES"  START = 1976.JAN
             DATA = (138 128 ... 297) }
REGRESSION { VARIABLES = (CONST SEASONAL) }
IDENTIFY   { DIFF = (0,1) }
```

Example 3 Produce ACF and PACF plots to identify the AR and MA parts of a regARIMA model. Do not print ACF and PACF tables. Suppose Y_t is the same series as in Example 1, that one nonseasonal and one seasonal difference are chosen, and that the model will include trading-day and Easter holiday effects. Because the **regression** spec is present, the **identify** spec first regresses $(1-B)(1-B^{12})y_t$ on $(1-B)(1-B^{12})x_{it}$, where the x_{it} are the regression variables for the trading-day and Easter holiday effects, and y_t consists of the logarithms of the original data Y_t adjusted for length-of-month effects. (See the description of `td` in the **regression** spec.) If $\tilde{\beta}_i$ denote the estimated regression coefficients, then this **identify** spec produces ACF and PACF plots for the regression residual series $(1-B)(1-B^{12})(y_t - \sum_i \tilde{\beta}_i x_{it})$. The ACFs and PACFs are computed through lag 30.

```
Series { Title = "Monthly Sales"  Start = 1976.Jan
         Data = (138 128 ... 297) }
Transform { Function = Log }
Regression { Variables = (Td Easter[14])}
Identify { Diff = (1)  Sdiff = (1)  Maxlag = 30
         Print = (None +ACFplot +PACFplot)  }
```

Example 4 Produce ACFs and PACFs (through lag 16) for model identification, and also estimate a tentative model for a quarterly series. There is a known level shift in the first quarter of 1971. Its effect is estimated by regressing $(1-B)(1-B^4)y_t$ on the differenced level shift variable. This regression effect is then removed to produce the (undifferenced) regression residual series, $\tilde{z}_t = y_t - \tilde{\beta}\text{LS71.1}_t$, and ACFs and PACFs are calculated for $\tilde{z}_t$, $(1-B)\tilde{z}_t$, $(1-B^4)\tilde{z}_t$, and $(1-B)(1-B^4)\tilde{z}_t$. The **identify** spec ignores the information in the **arima** spec.

The spec file below also specifies estimation and standard diagnostic checks of the regARIMA model, $(1-B)(1-B^4)(y_t - \beta\text{LS71.1}_t) = (1-\theta B)(1-\Theta B^4)a_t$. Such an estimation of a tentative model on the same run that produces ACFs and PACFs for model identification is sometimes useful, if one has a prior idea what ARIMA model might be appropriate. This might be the case if the series had been modelled previously, but new data has since extended the series. If the diagnostic checks suggest that the tentative model is inadequate, the user will have information from both the diagnostic checks and the **identify** spec output to use in selecting a new model.

```
series { title = "Quarterly Sales"  start = 1963.1  period = 4
           data = (56.7 57.7 ... 68.0) }
regression { variables = (ls1971.1) }
arima { model = (0 1 1)(0 1 1) }
identify { diff = (0, 1)  sdiff = (0, 1)  maxlag = 16 }
estimate {  }
check {  }
```

OUTLIER

DESCRIPTION

Specification to perform automatic detection of additive (point) outliers, temporary change outliers, level shifts, or any combination of the three using the specified model. After outliers (referring to any of the outlier types mentioned above) have been identified, the appropriate regression variables are incorporated into the model as “Automatically Identified Outliers”, and the model is reestimated. This procedure is repeated until no additional outliers are found. If two or more level shifts are detected (or are present in the model due to the specification of level shift(s) in the **regression** spec), t-statistics can be computed to test null hypotheses that each run of two or more successive level shifts cancels to form a temporary level shift.

USAGE

outlier { **types** = `all`
critical = `3.75`
method = `addall`
span = `(1983.may, 1992.sep)`
lsrun = `0`
print = `(none +header)`
save = `(tests)` }

ARGUMENTS

critical Sets the value to which the absolute values of the outlier t-statistics are compared to detect outliers. The default critical value is determined by the number of observations in the interval searched for outliers (see the **span** argument below). It is obtained by a modification of the asymptotic formula of Ljung (1993) that interpolates critical values for numbers of observations between three and ninety-nine. Table 6-15 gives default critical values for various outlier span lengths.

If only one value is given for this argument (`critical = 3.5`), then this critical value is used for all types of outliers. If a list of up to three values is given (`critical = (3.5, 4.0, 4.0)`), then the critical value for additive outliers is set to the first list entry (3.5 in this case), the critical value for level shift outliers is set to the second list entry (4.0), and the critical value for temporary change outliers is set to the third list entry (4.0). A missing value, as in `critical = (3.25,,3.25)`, is set to the default critical value. Raising the critical value decreases the sensitivity of the outlier detection routine, possibly decreasing the number of observations treated as outliers.

Table 6-14 : Default Critical Values for Outlier Identification

Number of Observations Tested	Outlier Critical Value	Number of Observations Tested	Outlier Critical Value
1	1.96	48	3.63
2	2.24	72	3.73
3	2.44	96	3.80
4	2.62	120	3.85
5	2.74	144	3.89
6	2.84	168	3.92
7	2.92	192	3.95
8	2.99	216	3.97
9	3.04	240	3.99
10	3.09	264	4.01
11	3.13	288	4.03
12	3.16	312	4.04
24	3.42	336	4.05
36	3.55	360	4.07

lsrun Compute t-statistics to test null hypotheses that each run of $2, \ldots,$ **lsrun** successive level shifts cancels to form a temporary level shift. The t-statistics are computed as the sum of the estimated parameters for the level shifts in each run divided by the appropriate standard error. (See Otto and Bell (1993).) Both automatically identified level shifts and level shifts specified in the **regression** spec are used in the tests. **Lsrun** may be given values from 0 to 5; 0 and 1 request no computation of temporary level shift t-statistics. If the value specified for **lsrun** exceeds the total number of level shifts in the model following outlier detection, then **lsrun** is reset to this total. The default value for **lsrun** is 0, i.e., no temporary level shift t-statistics are computed.

For details on handling temporary level shifts, see DETAILS.

method Determines how the program successively adds detected outliers to the model. The choices are `method = addone` or `method = addall`. See DETAILS for a description of these two methods. The default is `method = addone`.

print and **save** The tables available for output are listed on Table 6-16. **Note:** The entry for an outlier t-statistic in the **finaltests** table is set to zero whenever testing for the outlier (regressor) causes the regression matrix to be singular, and for any outliers specified in the **variables** argument of the **regression** spec. Also, when the **finaltests** table is saved, the t-statistics for all automatically identified outliers are also set to zero.

Table 6-16: Available Output Tables for Outlier

name	*default*	*brief*	*ext*	*description of table*
header	+	·	·	options specified for outlier detection including critical value, outlier span, and types of outliers searched for
iterations	·	·	oit	detailed results for each iteration of outlier detection including outliers detected, outliers deleted, model parameter estimates, and robust and non-robust estimates of the residual standard deviation
tests	·	·	·	t-statistics for every time point and outlier type on each outlier detection iteration
temporaryls	+	+	·	summary of t-statistics for temporary level shift tests
finaltests	·	·	fts	t-statistics for every time point and outlier type generated during the final outlier detection iteration

Name gives the name of each table for use with the **print** and **save** arguments.
Default indicates which tables are printed (+) or not printed (·) by default.
Brief indicates which tables are printed (+) or not printed (·) when the **brief** print level is specified. See section 5.1 for more information on print levels.
Ext gives the file extension used if the table is saved. A dot indicates that the table cannot be saved. The file extensions given can also be used as short names for the tables in the **print** and **save** arguments.

span Specifies start and end dates of a span of the time series to be searched for outliers. The start and end dates of the span must both lie within the series and within the model span if one is specified by the **modelspan** argument of the **series** spec, and the start date must precede the end date. A missing value, e.g., `span = (1976.jan, )`, defaults to the start date or end date of the series, as appropriate. (If there is a **span** argument in the **series** spec, then, in the above remarks, replace the start and end dates of the series by the start and end dates of the span given in the **series** spec.)

types Specifies the types of outliers to detect. The choices are: `types = ao`, detect additive outliers only; `types = ls`, detect level shifts only; `types = tc`, detect temporary change outliers only; `types = all`, detect additive outliers, temporary change outliers, and level shifts simultaneously; or `types = none`, turn off outlier detection (but not t-statistics for temporary level shifts). The default is `types = (ao ls)`.

RARELY USED ARGUMENTS

tcrate Defines the rate of decay for the temporary change outlier regressor. This value must be a number greater than zero and less than one. The default value is `tcrate=0.7 ** (12/period)`, where period is the number of observations in one year (for monthly time series, 4 for quarterly time series. This formula for the default value of **tcrate** ensures the same rate of decay over an entire year for series of different periodicity. If this argument is specified in the **regression** spec, it is not necessary to include it in this spec.

DETAILS

A level shift at the first data point cannot be estimated since the level of the series prior to the given data is unknown. Therefore, no LS test statistics is calculated for the first data point. Also, an LS at the last data point cannot be distinguished from an AO there, and an LS at the second data point cannot be distinguished from an AO at the first data point. Thus, no LS statistics are calculated for the second and last data points if AOs are also being detected. LS statistics that are not calculated are set to and printed out as 0.

Similarly, a temporary change (TC) outlier at the last data point cannot be distinguished from an AO there, so no TC statistics is calculated for the last data point if AOs are also being detected. TC statistics that are not calculated are set to and printed out as 0.

The ***addone*** method works in the following way. The program calculates t-statistics for each type of outlier specified (AO, TC and/or LS) at all time points for which outlier detection is being performed. If the maximum absolute outlier t-statistic exceeds the critical value, then an outlier has been detected and the appropriate regression variable is added to the model. The program then estimates the new model (the old model with the detected outlier added) and looks for an additional outlier. This process is repeated until no additional outliers are found. At this point, a backward deletion process is used to delete "insignificant" outliers (those whose absolute t-statistics no longer exceed the critical value) from the model. This is done one at a time beginning with the least significant outlier, until all outliers remaining in the model are signficant. During backward deletion the usual (non-robust) residual variance estimate is used, which can yield somewhat different outlier t-statistics than those obtained during outlier detection.

The ***addall*** method follows the same general steps as the `addone` method, except that on each outlier detection pass the `addall` method adds to the model *all* outliers with absolute t-statistics exceeding the critical value. Typically several of the outliers added this way will be found to be insignificant when the new model is estimated. The `addall` method thus depends heavily on the backward deletion process (much more than does the `addone` method) to remove unnecessary outliers added to the model in the detection phase.

The differences between the `addone` and `addall` schemes can produce different final sets of detected outliers. Two practical differences between the methods are worth noting. First, the `addone` method generally takes more computation time than does `addall`. Second, the `addall` method may add so many outliers on a detection pass that it exceeds the maximum number of regression variables allowed in a model. In this case the program prints an error message to this effect and stops. Suggested remedies are to raise the cutoff value so fewer outliers are detected, or to switch to the `addone` method, for which this phenomenon is much less likely.

For either method, the outlier t-statistics for all possible time points on each detection pass can be printed by specifying `print=tests`. This option generates considerable output.

Choosing the critical value requires both judgement and experience. Based on a simulation study involving series of length up to 200 generated from low order nonseasonal ARIMA models, Chang, Tiao, and Chen (1988) recommended critical values of 3 for high sensitivity in detection of AO outliers, 3.5 for medium sensitivity, and 4 for low sensitivity.

Outlier detection begins with the model specified by the **regression** and **arima** specs and with estimated parameters. If the **estimate** spec is absent, the **outlier** spec forces estimation of the model (with default estimation options) prior to outlier detection.

If outliers are suspected at specific known time points, then they may be included in the model by adding the appropriate AO, TC, or LS regression variables to the model in the **regression** spec.

Outlier detection results can vary depending on the regARIMA model specified: observations are classified as outliers because the model fits them less well than most of the other observations. Therefore a very inadequate regARIMA model can yield inappropriate outlier adjustments.

If one or more temporary level shift t-statistics indicate that a run of 2 (or more) successive level shifts cancels, a user-defined regressor can be used to capture the temporary level shift effect. In this way, two or more level shifts can be replaced by one user-defined regressor. The **usertype** argument should be set to **ls** for this regressor, so the user defined regressor is treated as a level shift. For more information on how to specify user-defined regressors, see the **regression** spec.

Another technique can be used if the span of observations affected by the temporary level shift is small. Individual AO outliers can be specified starting at the point of the first level shift and stopping with the point before the final level shift.

EXAMPLES

The following examples show complete spec files.

Example 1 Simultaneously search for both AO and LS outliers over the entire time series, using the `addone` method and a critical value that depends on the number of observations in the interval searched for outliers (default options). If the number of level shifts present in the model following outlier detection is two or more, compute t-statistics to test whether each run of 2, ..., 5 successive level shifts cancels to form a temporary level shift. Though the **estimate** spec is absent, the presence of the **outlier** spec forces model estimation with default estimation options.

```
series { title = "Monthly sales"  start = 1976.jan
         data = (138 128 ... 297) }
arima { model = (0 1 1)(0 1 1)12 }
outlier {lsrun = 5  types=(ao ls) }
```

Example 2 Search only for AO outliers using the `addall` method and a critical value of $t = 4.0$. Because the **span** argument is present in the **series** spec, only the time frame given there (January 1980 through December 1992) is used in model estimation and in outlier detection. The two level shifts specified in the **regression** spec are not tested for cancellation into a temporary level shift since **lsrun** takes on its default value of 0.

```
Series { Title = "Monthly Sales"  Start = 1976.Jan
         Data = (138 128 ... 297)
         Span = (1980.Jan, 1992.Dec) }
Regression { Variables = (LS1981.Jun LS1990.Nov) }
Arima { Model = (0 1 1)(0 1 1)12 }
Estimate {  }
Outlier { Types = AO   Method = Addall  Critical = 4.0 }
```

Example 3 Estimate the model using the same span as in Example 2, but search only for LS outliers in 1987 and 1988. The default `addone` method is used, but with a critical value of $t = 3.0$. Each pair of successive LSs is tested for possible cancellation into a temporary LS.

```
series { title = "Monthly sales"  start = 1976.jan
        data = (138 128 ... 297)
        span = (1980.jan, 1992.dec) }
arima { model = (0 1 1)(0 1 1)12 }
estimate { }
outlier { types = ls
          critical = 3.0
          lsrun = 2
          span = (1987.jan, 1988.dec) }
```

Example 4 Estimate the model using the same span as in Examples 2 and 3, but search for AO, TC, and LS outliers. The default `addone` method is used, but with a critical value of $t_{AO} = 3.0$ for AO outliers, $t_{LS} = 4.5$ for LS outliers, and $t_{TC} = 4.0$ for TC outliers.

```
series { title = "Monthly sales"  start = 1976.jan
        data = (138 128 ... 297)
        span = (1980.jan, 1992.dec) }
arima { model = (0 1 1)(0 1 1)12 }
estimate { }
outlier { critical = (3.0, 4.5, 4.0)   }
```

REGRESSION

DESCRIPTION

Specification for including regression variables in a regARIMA model, or for specifying regression variables whose effects are to be removed by the **identify** spec to aid ARIMA model identification. Predefined regression variables are selected with the **variables** argument. The available predefined variables provide regressors modeling a constant effect, fixed seasonality, trading-day and holiday variation, additive outliers, level shifts, and temporary changes or ramps. change-of-regime regression variables can be specified for seasonal and trading-day regressors. User-defined regression variables can be added to the model with the **user** argument. Data for any user-defined variables must be supplied, either in the **data** argument, or in a file specified by the **file** argument (not both). The **regression** spec can contain both predefined and user-defined regression variables.

USAGE

```
regression { variables = ( const
                           seasonal  or  sincos[1, 2, 3]
                           td  or  tdnolpyear  or  tdstock[31]  or
                                td1coef  or  td1nolpyear
                           lom  or  loq
                           lpyear
                           easter[8]
                           labor[8]
                           thank[1]
                           ao1967.apr
                           ls1972.sep
                           tc1979.sep
                           rp1965.nov-1968.may )
             print = (none)
             save = (rmx)
             savelog = aictest
             user = (temp precip)
             usertype = holiday
             start = 1955.jan
             data = (25 0.1 ... )  or   file = "weather.dat"
                                        format = "(2f5.1)"
             aictest  =  (  easter user
                           td  or  tdnolpyear  or  tdstock  or
                                td1coef  or  td1nolpyear )
           }
```

ARGUMENTS

aictest Specifies that an AIC-based selection will be used to determine if a given set of regression variables will be included with the regARIMA model specified. The only entries allowed for this variable are **td**, **tdnolpyear**, **tdstock**, **td1coef**, **td1nolpyear**, **easter**, and **user**. If a trading day model selection is specified, for example, then AIC values (with a correction for the length of the series, henceforth referred to as AICC) are derived for models with and without the specified trading day variable. By default, the model with smaller AICC is used to generate forecasts, identify outliers, etc. If more than one type of regressor is specified, the AIC-tests are performed sequentially in this order: (a) trading day regressors, (b) easter regressors, (c) user-defined regressors. If there are several variables of the same type (for example, several td regressors), then the **aictest** procedure is applied to them as a group. That is, either all variables of this type will be included in the final model or none. See DETAILS for more information on the testing procedure. If this option is not specified, no automatic AIC-based selection is performed.

data Assigns data values to the user-defined regression variables. The time frame of the data values must cover the time frame of the series (or of the span specified by the **span** argument of the **series** spec, if present). It must also cover the time frame of forecasts and backcasts requested in the **forecast** spec. The data values are read in free format. The numerical values given in this argument are assigned in the order in which the user-defined variables are named in the **user** argument. This assignment proceeds through all the user-defined variables for the first time point, then through all the variables for the second time point, etc. If the **data** argument is used, the **file** argument cannot be used.

file Name of the file containing data values for *all* user-defined regression variables. The filename must be enclosed in quotes. If the file is not in the current directory, the path must also be given. As with the **data** argument, the time frame of the data values must cover both the series and any forecasts and backcasts requested. If the **file** argument is used, the **data** argument cannot be used.

format Denotes the format used when reading the data for the regression variables from the file named in the **file** argument. Four types of input are accepted:

(a) a valid FORTRAN format, which must be enclosed in quotes and must include the initial and terminal parentheses (example: `format="(6f12.0)"`);
(b) “datevalue” format, in which the year, month or quarter, and the associated value for each of the user-defined regression variables for a given observation are given in this order in free format on individual lines in the data file. Thus, a line of the data file with three regressors

having the values 0, 0, and 1 respectively for July of 1991 would have the form `1991 7 0 0 1`. All the user-defined regressors must be on the same record, and in the order of their appearance in the **user** argument (example: `format="datevalue"`);

(c) the "x12save" format `X-12-ARIMA` uses to save a table. This allows the user to read in a file saved from a previous `X-12-ARIMA` run (example: `format="x12save"`).

If no **format** argument is given the data will be read in free format. In *free format*, all numbers on a line will be read before continuing to the next line, and the numbers must be separated by one or more spaces (not by commas or tabs). **Format** cannot be used with the **data** argument, only with **file**.

print and **save** The following optional output tables are available.

Table 6-17: Available Output Tables for Regression

name	*default*	*brief*	*ext*	*description of table*
regressionmatrix	·	·	rmx	values of regression variables with associated dates
aictest	+	+	·	output from AIC-based test(s) for trading day, easter, and user-defined regression variables
outlier	+	+	otl	combined regARIMA outlier factors
aoutlier	+	+	ao	regARIMA additive (or point) outlier factors
levelshift	+	+	ls	regARIMA level change and ramp outlier factors
temporarychange	+	+	tc	regARIMA temporary change outlier factors
tradingday	+	+	td	regARIMA trading day factors
holiday	+	+	hol	regARIMA holiday factors
regseasonal	+	+	a10	regARIMA user-defined seasonal factors
userdef	+	+	usr	factors from user-defined regression variables

Name gives the name of each table for use with the **print** and **save** arguments.
Default indicates which tables are printed (+) or not printed (·) by default.
Brief indicates which tables are printed (+) or not printed (·) when the **brief** print level is specified. See section 5.1 for more information on print levels.
Ext gives the file extension used if the table is saved. A dot indicates that the table cannot be saved. The file extensions given can also be used as short names for the tables in the **print** and **save** arguments.

savelog Setting `savelog=aictest` or `savelog=ats` causes the results of the AIC-based selection procedure specified by the **aictest** argument to be output to the log file (see section 2.6 for more information on the log file).

start The start date for the data values for the user-defined regression variables. The default is the start date of the series. Valid values are any date up to the start date of the series (or up to the start date of the span specified by the **span** argument of the **series** spec, if present).

user Specifies names for any user-defined regression variables. Names are required for all user-defined variables to be included in the model. The names given are used to label estimated coefficients in the program's output. Data values for the user-defined variables must be supplied, using either the **data** or **file** argument (not both). The maximum number of user-defined regression variables is 52. (This limit can be changed—see Section 2.8.)

usertype Assigns a type of model-estimated regression effect to each user-defined regression variable. It causes the variable and its estimated effects to be used and be output in the same way as a predefined regressor of the same type. This option is useful when trying out alternatives to the regression effects provided by the program.

The type of the user-defined regression effects can be defined as a constant (**constant**), seasonal (**seasonal** or **userseasonal**), trading day (**td**), stock trading day (**tdstock**), length-of-month (**lom**), length-of-quarter (**loq**), leap year (**lpyear**), holiday (**holiday**, **easter**, or the US holidays **thanks** and **labor**), outlier (**ao**, **ls**, **rp** or **tc**), or other user-defined (**user**) regression effects. One effect type can be specified for all the user-defined regression variables defined in the **regression** spec (`usertype=td`), or each user-defined regression variable can be given its own type (`usertype=(td td td td td td holiday user)`). Once a type other than user has been assigned to a user-defined variable, further specifications for the variable in other arguments, such as **aictest** or **noapply**, must use this type designation, not **user**. If this option is not specified, all user-defined variables have the type **user**. See DETAILS for more information on assigning types to user-defined regressors.

variables List of predefined regression variables to be included in the model. Data values for these variables are calculated by the program, mostly as functions of the calendar. See DETAILS for a discussion and a table of the available predefined variables. Also see Section 3.3 for additional information and a table defining the actual regression variables used.

RARELY USED ARGUMENTS

aicdiff Defines the amount by which the AIC value (corrected for the length of the series, or AICC) of the model with the regressor(s) specified in the **aictest** argument must fall below the AICC of the model without these regressor(s) in order for the model with the regressors to be chosen. The default value is `aicdiff=0.0`. For more information on how this option is used in conjuction with the **aictest** argument, see DETAILS.

b Specifies initial values for regression parameters in the order that they appear in the **variables** and **user** arguments. If present, the **b** argument must assign initial values to *all* regression coefficients in the regARIMA model. Initial values are assigned to parameters either by specifying the value in the argument list or by explicitly indicating that it is missing as in the example below. Missing values take on their default value of 0.1. For example, for a model with two regressors, `b=(0.7, )` is equivalent to `b=(0.7,0.1)`, but `b=(0.7)` is not allowed. For a model with three regressors, `b=(0.8,,-0.4)` is equivalent to `b=(0.8,0.1,-0.4)`. To hold a parameter fixed at a specified value, immediately follow the value in the **b** list with an '`f`', e.g., `b=(0.7f, 0.1)`.

centeruser Specifies the removal of the (sample) mean or the seasonal means from the user-defined regression variables. If `centeruser=mean`, the mean of each user-defined regressor is subtracted from the regressor. If `centeruser=seasonal`, means for each calendar month (or quarter) are subtracted from each of the user-defined regressors. If this option is not specified, the user-defined regressors are assumed to already be in an appropriately centered form and are not modified.

eastermeans Specifies whether long term (400 year) monthly means are used to deseasonalize the Easter regressor associated with the variable `easter[w]`, as described in footnote 5 of Table 3-1 (`eastermeans=yes`), or, instead, monthly means calculated from the span of data used for the calculation of the coefficients of the Easter regressors (`eastermeans=no`). The default is `eastermeans=yes`. This argument is ignored if no built-in Easter regressor is included in the regression model, or if the only Easter regressor is `sceaster[w]` (see Details).

noapply List of the types of regression effects defined in the **regression** spec whose model-estimated values are **not** to be removed from the original series before the seasonal adjustment calculations specified by the **x11** spec are performed.

Applicable types are all modelled trading day effects (**td**), Easter, Labor Day, and Thanksgiving-Christmas holiday effects (**holiday**), point outliers (**ao**), level changes and ramps (**ls**), temporary changes (**tc**), and the set of user-defined regression effects (**user**).

tcrate Defines the rate of decay for the temporary change outlier regressor. This value must be a number greater than zero and less than one. The default value is `tcrate=0.7 ** (12 / period)`, where period is the number of observations in one year (for monthly time series, 4 for quarterly time series. This formula for the default value of **tcrate** ensures the same rate of decay over an entire year for series of different periodicity.

DETAILS

If forecasting is performed, `X-12-ARIMA` creates data values for the selected predefined regression variables for the entire forecast period. If there are any user-defined regression variables, then data values must also be supplied for them for the entire forecast period (similarly for the backcasts). In addition to the limit of 52 user-defined regression variables, there is an overall limit of 80 regression variables in the model. (These limits can be changed—see Section 2.8.) The latter limit is on the total number of predefined and user-defined regression variables plus the number of regression variables added automatically by the outlier spec. The maximum length of the series of user-defined regression variables, not including the forecast period, is 600. (This limit can also be changed—see Section 2.8.)

If *`const`* is specified in the **variables** argument, then the resulting regression variable allows for an a constant term in the series resulting from any differencing operations in the ARIMA model. If the ARIMA model involves no differencing, this is simply the usual regression constant term for a nonzero overall mean; if the ARIMA model does involve differencing, this regressor is called a trend constant. In the latter case the actual regression variable created is defined such that, after differencing, it yields a column of ones. See Section 3.3 for discussion.

We generally recommend specifying *`td`* in the **variables** argument when trading-day effects are thought to be present in a monthly *flow* time series, that is, a series whose values are monthly accumulations of daily values. In this case, how the program handles leap-year effects depends on information from the **transform** spec. If the series is transformed (Box-Cox or logistic transformation) then leap-year effects are removed by prior adjustment: the series is divided before transformation by a set of factors lp_t where $lp_t = 28.25/29$ if t is a leap year February, $lp_t = 28.25/28$ if t is a non-leap year February, and $lp_t = 1.00$ otherwise.

If the series is not transformed, then the leap-year regression variable *`lpyear`* is included in the model. Its values, denoted by LP_t are given by $LP_t = 29 - 28.25$ if t is a leap year February, $LP_t = 28 - 28.25$ if t is a non-leap year February, and $LP_t = 0.00$ otherwise. In both cases, the *`tdnolpyear`* regression variables, (no. of *Mondays*) − (no. of *Sundays*), …, (no. of *Saturdays*) − (no. of *Sundays*), are also included in the model.

The leap year regressor is the nonseasonal component for the length-of-month (quarter) regressor. If the user prefers to model length-of-month effects in a transformed series through the `lom` regression variable, this can be done by specifying both `lom` and `tdnolpyear`, i.e., `variables = (lom tdnolpyear ...)`. If the user prefers to prior adjust an untransformed series for length-of-month effects, this can be done by specifying `variables = (tdnolpyear ...)` in the regression spec and `adjust = lom` in the **transform** spec.

Table 6-18: Predefined regression variables

Variable	Description
const	Trend constant regression variable to allow for a nonzero overall mean for the differenced data.
seasonal	Fixed seasonal effects parameterized via $s-1$ seasonal contrast variables (s = seasonal period). The resulting variables allow for month-to-month (or quarter-to-quarter, etc.) differences in level, but have no net effect on overall level. `Seasonal` cannot be used with `sincos` and also not in models with seasonal differencing except as a partial change of regime variable (see DETAILS).
sincos[ ]	Fixed seasonal effects (for s = seasonal period) parameterized via trigonometric regression variables of the form $\sin(\omega_j t)$ and $\cos(\omega_j t)$ at seasonal frequencies $\omega_j = (2\pi j/s)$ for $1 \leq j \leq s/2$ (dropping $\sin(\omega_j t) \equiv 0$ for $j = s/2$ for s even). Each frequency to be included must be specified, i.e., for monthly series `sincos[1, 2, 3, 4, 5, 6]` includes all seasonal frequencies while `sincos[1, 2, 3]` includes only the first three. `Sincos[ ]` cannot be used with `seasonal` or in models with seasonal differencing.
td	Estimate monthly (or quarterly) flow trading-day effects by including the `tdnolpyear` variables (see below) in the model, and by handling leap-year effects either by rescaling (for transformed series) or by including the `lpyear` regression variable (for untransformed series). `Td` can only be used for monthly or quarterly series, and cannot be used with `tdnolpyear`, `td1coef`, `td1nolpyear`, `lpyear`, `lom`, `loq`, or `tdstock[]`. If `td` is specified, do not specify `adjust = lpyear` or `adjust = lom` (`adjust = loq`) in the **transform** spec.
tdnolpyear	Include the six day-of-week contrast variables (monthly and quarterly flow series only): (no. of *Mondays*) − (no. of *Sundays*), . . . , (no. of *Saturdays*) − (no. of *Sundays*). `Tdnolpyear` cannot be used with `td`, `td1coef`, `td1nolpyear`, or `tdstock`.
td1coef	Estimate monthly (or quarterly) flow trading-day effects by including the `td1nolpyear` variable (see below) in the model, and by handling leap-year effects either by rescaling (for transformed series) or by including the `lpyear` regression variable (for untransformed series). `Td1coef` can only be used for monthly or quarterly series, and cannot be used with `td`, `tdnolpyear`, `td1nolpyear`, `lpyear`, `lom`, `loq`, or `tdstock[]`. If `td1coef` is specified, do not specify `adjust = lpyear` or `adjust = lom` (`adjust = loq`) in the **transform** spec.
td1nolpyear	Include the weekday-weekend contrast variable (monthly and quarterly flow series only): (no. of *weekdays*) $-\frac{5}{2}$ (no. of *Saturdays and Sundays*). `Td1nolpyear` cannot be used with `td`, `tdnolpyear`, `td1coef`, or `tdstock`.
lpyear	Include a contrast variable for leap-year (monthly and quarterly flow series only): 0.75 for leap-year Februaries (first quarters), -0.25 for non-leap year Februaries (first quarters), 0.0 otherwise. `Lpyear` cannot be used with `td`, `td1coef`, or `tdstock`.
lom	Include length-of-month as a regression variable. If `lom` is requested for a quarterly series, `X-12-ARIMA` uses `loq` instead. Requesting `lom` when $s \neq 12$ or 4 results in an error. `Lom` cannot be used with `td`, `td1coef`, or `tdstock`.
loq	Include length-of-quarter as a regression variable. If `loq` is requested for a monthly series, `X-12-ARIMA` uses `lom` instead. The same restrictions that apply to `lom` apply to `loq`.
tdstock[w]	Estimate day-of-week effects for inventories or other stocks reported for the w^{th} day of each month. The value w must be supplied and can range from 1 to 31. For any month of length less than the specified w, the `tdstock` variables are measured as of the end of the month. Use `tdstock[31]` for end-of-month stock series. `Tdstock` can be used only with monthly series and cannot be used with `td`, `tdnolpyear`, `td1coef`, `td1nolpyear`, `lom`, or `loq`.
easter[w]	Easter holiday regression variable (monthly or quarterly flow data only) which assumes the level of daily activity changes on the $w-$th day before Easter and remains at the new level until the day before Easter. The value w must be supplied and can range from 1 to 25. To estimate complex effects, several of these variables, differing in their choices of w, can be specified.
labor[w]	Labor Day holiday regression variable (monthly flow data only) that assumes the level of daily activity changes on the $w-$th day before Labor Day and remains at the new level until the day before Labor Day. The value w must be supplied and can range from 1 to 25.
thank[w]	Thanksgiving holiday regression variable (monthly flow data only) that assumes the level of daily activity changes on the $w-$th day before or after Thanksgiving and remains at the new level until December 24. The value w must be supplied and can range from -8 to 17. Values of $w < 0$ indicate a number of days after Thanksgiving; values of $w > 0$ indicate a number of days before Thanksgiving.

Table 6-18: Predefined regression variables (continued)

Variable	Description
sceaster[w]	Statistics Canada Easter holiday regression variable (monthly or quarterly flow data only) assumes that the level of daily activity changes on the $(w-1)-$th day and remains at the new level through Easter day. The value w must be supplied and can range from 1 to 24. To estimate complex effects, several of these variables, differing in their choices of w, can be specified.
ao*date*	Additive (point) outlier variable, AO, for the given date or observation number. For series with associated dates, AOs are specified as `ao`*date*. For monthly series this is `ao`*year.month* (e.g., `ao1985.jul` or `ao1985.7`), for quarterly series this is `ao`*year.quarter* (e.g., `ao1985.1` for an AO in the first quarter of 1985), and for annual series this is `ao`*year* (e.g., `ao1922`). For series without associated dates, AOs are specified as `ao`*observation number*, e.g., `ao50` for an AO at observation 50. More than one AO may be specified. All specified outlier dates must occur within the series. (AOs with dates within the series but outside the span specified by the **span** argument of the **series** spec are ignored.)
ls*date*	Regression variable for a constant level shift (in the transformed series) beginning on the given date, e.g., `ls1990.oct` for a level shift beginning in October 1990. More than one level shift may be specified. Dates are specified as for AOs and the same restrictions apply with one addition: level shifts cannot be specified to occur on the start date of the series (or of the span specified by the **span** argument of the **series** spec).
tc*date*	Regression variable for a temporary level change (in the transformed series) beginning on the given date, e.g., `tc1990.oct` for a temporary change beginning in October 1990. More than one temporary change may be specified. Dates are specified as for AOs, and the same restrictions apply.
rp*date*-*date*	Ramp effect which begins and ends on the given dates, e.g., `rp1988.apr-1990.oct`. More than one ramp effect may be specified. All dates of the ramps must occur within the series. (Ramps specified within the series but with both start and end dates outside the span specified by the **span** argument of the **series** spec are ignored.) Ramps can overlap other ramps, AOs, and level shifts.

If `adjust=lom` is specified in the **transform** spec, then including either `td` or `lom` in the **variables** list leads to a conflict. The conflict occurs either because two requests have been made to rescale the series by dividing by length of month, or because both a length-of-month rescaling and the `lom` regression variable have been requested (which will generally lead to a singular system of equations for the regression coefficients). In this case, the user should either (i) remove `adjust=lom` from the **transform** spec, or (ii) in the **variables** list, replace `td` by `tdnolpyear`, or drop `lom`.

For quarterly flow time series the same trading-day options are available, and the above comments apply with `lom` replaced by `loq`.

The values `lom` and `loq` are equivalent —if either is specified, the seasonal period specified in the **series** spec determines which is used. Thus, `period = 12` implies `lom` and `period = 4` implies `loq`. Also, note that `lom` or `loq` can be specified without `tdnolpyear`. This could be done to account for fixed seasonality due to length-of-month (or length-of-quarter) effects for a series with no day-of-week specific effects. Predefined length-of-period variables are available only for monthly or quarterly flow series.

For *stock* series, such as inventories, the program can estimate trading-day effects only for monthly series. *`Tdstock[w]`*, where w can range from 1 to 31, creates six regression variables contrasting six days of the week

Table 6-19: Change of Regime Regressor Types and Syntax

Type	*Syntax*	*Example*
Full change of regime regressor	*reg/date/*	**td/1990.jan/**
Partial change of regime regressor, zero before change date	*reg//date/*	**td//1990.jan/**
Partial change of regime regressor, zero on and after change date	*reg/date//*	**td/1990.jan//**

with the seventh - see Section 3.3. The value w must be specified; it denotes the day of the month for which the stock is reported or the last day of the month, whichever is smaller. Therefore, `tdstock[31]` is used for end-of-month stocks.

The holiday effect regression variables (for Easter, Labor Day, and Thanksgiving) are for flow series. The Easter variable can be specified for either monthly or quarterly series. The Labor Day and Thanksgiving variables are only for monthly series.

Change-of-regime regression variables can be specified for seasonal (`seasonal`), trigonometric seasonal (`sincos`), trading day (`td, td-nolpyear, or tdstock`), leap year (`lpyear`), length-of-month (`lom`), and length-of-quarter (`loq`) regression variables. Two types of change-of-regime regressors are available: full and partial.

As Table 6-19 shows, change of regime regressors are specified by appending the change date, surrounded by one or two slashes, to the name of a regression variable in the **variables** argument of the **regression** spec. The date specified for the change of regime divides the series being modeled into two spans, an early span containing the data for times prior to the change date and a late span containing the data from on and after this date. Partial change of regime variables are restricted to one of these two spans, being zero in the complementary span. The full change of regime variables estimate both the basic regression of interest and the partial change of regime regression for the early span. For example, the full change of regime specification `variables = (td/1990.jan/)`is equivalent to the specification `variables = (td td/1990.jan//)`. It causes the program to output the coefficients estimated for `td` and for `td/1990.jan//` along with trading day factors for their combined effects.

The coefficients resulting from use of a full change of regime regression have convenient interpretations: Let the basic regressors be denoted by X_{jt}, and let t_0 be the change point. Then the partial change of regime regressors for the early regime are

$$X_{jt}^{E} = \begin{cases} X_{jt} & t < t_0 \\ 0 & \text{for } t \geq t_0 \end{cases},$$

and those for the late regime can be calculated as $X_{jt}^{L} = X_{jt} - X_{jt}^{E}$. For the data transformed as indicated in the **transform** spec, the effect estimated by the full change of regime regression has the form

$$\sum_j a_j X_{jt} + \sum_j b_j X_{jt}^E = \sum_j a_j X_{jt}^L + \sum_j (a_j + b_j) X_{jt}^E.$$

From the right-hand-side formula, we observe that the coefficients a_j of the basic regressors X_{jt} can be interpreted at the coefficients of the late-span regressors X_{jt}^L, and the coefficients b_j of the X_{jt}^E can be interpreted as measuring the change in the coefficients of the late-span regressors required to obtain coefficients for the early-span effects. Therefore, statistically significant b_j indicate the nature of the change of regime.

We illustrate two other natural uses for partial change of regime variables. First, the specification `variables = (td//1990.jan/)` can be used to estimate the trading day component of a series that has no statistically significant trading day effects prior to 1990, but possibly significant effects beginning in this year. Second, when an ARIMA model with seasonal differencing is specified in the **arima** spec, or in the models estimated by the **automdl** spec, then the specification `variables = (seasonal//1990.jan/)` can be used to estimate a fixed change in a somewhat variable seasonal pattern that takes place in January of 1990 and to test for the statistical significance of the estimated change.

The effect of the argument **aictest** can be to delete a regressor set named in the **variables** list from this list, or to add a regressor set to the **variables** list. The effect of a nonzero (positive) value of **aicdiff** is to make it more difficult for the **aictest** procedure to include in the model the variable being tested. Let Δ_{AICC} denote the value associated with the **aicdiff** argument, which by default is zero. Let $AICC^{with}$ (and $AICC^{without}$) denote the AICC value of the model with (or without) a set of regressors specified int he **aictest** argument. If this set is not named in the **variables** list, it will be added to the regression model if

$$AICC^{with} + \Delta_{AIC} < AICC^{without}.$$

If this set is named in the variables list, it will be retained in the regARIMA model only if this inequality holds.

In the second case, if `aictest = (tdstock)`, then the end-of-month stock variables, specified by `tdstock[31]`, are the variables added, because 31 is the default value for w in $tdstock[w]$.

There are more possibilities if `aictest = (easter)` and no Easter effect regressors appear in the **variables** list. Then three additional models are considered, the three models obtained by augmenting the specified regARIMA model with the regressor $easter[w]$ for $w = 1, 8, 15$ respectively. The Easter regressor whose model has the smallest AICC is retained if its AICC is smaller than the model with no Easter regressors by at least the amount $\Delta_{AIC} = 0$; otherwise, the model without Easter regressors is selected.

Simulation experiments we have conducted suggest that AICC does not distinguish with high reliability between $easter[w]$ regressors whose w values differ by less than seven. The out-of-sample forecast diagnostics produced by the **history** spec can sometimes distinguish between such regressors by showing that one provides persistently more accurate forecasts, and therefore presumably better describes the Easter effect in the data.

When trading day regressors appear in both the **aictest** and **variables** arguments, the type of regressors specified must be identical. The sample day for stock trading day variables and the date specified for change-of-regime regressors should **not** be included in the **aictest** argument; they will be assumed from the entry in the **variables** argument. For example, if `variables=(tdstock[15] ao1995.jan)`, then the entry for **aictest** should be `tdstock`.

User-defined variables should be input to the program in deseasonalized form (unless they are seasonal regressors). The deseasonalization method described for additive decompositions in the Details section for x11regression is likely to be the appropriate one, because regressors are additive components of the regARIMA model. If deseasonalization is not done, then the seasonal factors will not include all estimated seasonal effects. Another problem is that regressors with seasonal components are likely to have estimated coefficients, and estimated effects, that are more correlated with one another and therefore more difficult to interpret.

If a type is assigned in to a user-defined variable with the **usertype** argument, the factor derived from the user-defined regression variables of that type will be combined with the regression factor from variables of the same type specified in the **regression** spec. The resulting factor will be adjusted out of the series for the seasonal adjustment factor calculations determined by the **x11** spec unless the type name appears in the **noapply** argument.

Setting `usertype=seasonal` will cause the user-defined regressors to be treated exactly like seasonal regressors specified in the **variables** argument: the seasonal effect estimated from these regressors will not adjusted out of the series prior to seasonal adjustment. By contrast, setting `usertype=userseasonal` will cause seasonal factors to be created from the user-defined regressors that will be adjusted out of the original series before the seasonal adjustment specified by the **x11** spec is calculated. Combined seasonal factors are created from the X-11 and regression factors.

Note that if `format = "datevalue"` or `format = "x12save"`, the starting date of the user defined regressor(s) is automatically read from the data file. Therefore, the starting date need not be specified with the **start** argument of the **regression** spec.

Trading day and/or holiday regressors may not be specified simultaneously in **regression** and **x11regression** unless the **noapply** option is used to specify that the effects estimated by either **regression** or **x11regression** not be used to adjust the series.

The two choices for the argument **eastermeans** yield noticeably different holiday factors. But the choice has no effect on forecasts (provided the regARIMA model used includes seasonal differencing or the fixed seasonal regressors) and usually has only negligible effects on the combined seasonal and holiday factors, because the seasonal factors change to compensate for the differences between the choices.

The monthly means used to obtain deseasonalized Easter regressors under `eastermeans=yes` were generated from frequencies of the date of Easter for the first 400 year period of the Gregorian calendar, 1583-1982. These frequencies are given in Montes (1997); the algorithm used to compute the date of Easter for the Gregorian calendar is given in Duffet-Smith (1981).

For a nonseasonal time series, an adjustment for trading day and holiday effects estimated by means of this spec can be obtained by setting `type=trend` in the **x11** spec.

When the `b=()` argument is used to fix coefficients, *AIC* and the other model selection statistics may become invalid, see the Details section of **estimate**.

EXAMPLES

The following examples show complete spec files.

Example 1 Estimate a model with ARIMA (0 1 1) errors, fixed seasonal effects, and a trend constant.

```
SERIES      { TITLE = "Monthly sales"  START = 1976.JAN
               DATA = (138 128 ... 297) }
REGRESSION { VARIABLES = (CONST SEASONAL) }
ARIMA { MODEL = (0 1 1) }
ESTIMATE { }
```

Example 2 Specify a model to fit sine and cosine variables with the 4th and 5th seasonal frequency by ordinary least squares to the final irregular component of a series to test if "visually significant" spectrum peaks at these frequencies are statistically signifcant.

```
series { title = "Irregular Component of Monthly Sales"
         start = 1976.jan
         file = "sales.d13"
         format = "x12save"
       }
regression { variables = (const sincos[4,5]) }
estimate { }
```

Example 3 Specify regression variables for trading-day, Easter, Labor Day, and Thanksgiving effects in a monthly time series. The duration in number of days is specified for each holiday effect. Since `td` is specified and the series is log transformed, the original series (before transformation) is divided by the leap-year factors, and the `tdnolpyear` regression variables are fit to the transformed series. The regression coefficients are estimated by the **identify** spec through a regression of the maximally differenced series (after transformation and length-of-month adjustment) on the correspondingly differenced regression variables. The **identify** spec then produces various sample ACFs and PACFs (of the regression residuals) to be used for identifying an ARIMA model for the regression errors.

```
Series { Title = "Monthly Sales"  Start = 1976.Jan
          Data = (138 128 ... 297) }
Transform { Function = Log }
Regression { Variables = (TD Easter[8]
                          Labor[10] Thank[3]) }
Identify { Diff = (0 1) SDiff = (0 1) }
```

Example 4 Estimate a model including the same regressors as in Example 3, and also the `lom` regression variable in place of the division of the series by standard leap-year effects that the argument value `td` invokes. (Replacing the value of `td` with `tdnolpyear` prevents the division by the standard leap year effects.) Perform a test (using AICC) of the significance of the trading-day and Easter regressors. An ARIMA (0 1 1)(0 1 1)$_{12}$ model is used for the regression error series.

```
series      { title = "Monthly sales"  start = 1976.jan
               data = (138 128 ... 297) }
transform { function = log }
regression { variables = (tdnolpyear lom easter[8]
                          labor[10] thank[3])
             aictest = (tdnolpyear easter) }
arima { model = (0 1 1)(0 1 1) }
estimate    { }
```

Example 5 Estimate a model with trading-day effects, two AOs, and two LSs for a quarterly seasonal series. Accounging for these effects, the transformed series follows an ARIMA (0 1 1)(0 1 1)$_4$ model.

```
Series {Title = "Quarterly Sales"  Start = 1963.1  Period = 4
         Data = (1039 1241 ...  2210) }
Transform { Function = Log }
Regression { Variables = (AO1967.1 LS1985.3 LS1987.2 AO1978.1
                          TD) }
Arima { Model = (0 1 1)(0 1 1) }
Estimate    { }
```

Example 6 Estimate a user-defined regression variable for a temporary level-shift from the third quarter of 1985 through the first quarter of 1987. The effect of the temporary level shift is removed through the regression performed by the **identify** spec, prior to the computation of ACFs and PACFs for identification of the ARIMA part of the model.

```
series {title = "Quarterly sales"  start = 1981.1
         data = (301 294 ...  391)  period = 4  }
regression {user = tls
        data =  (0 0 0 0 0 0 0 0 0 0 0 0 ...
                  0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 ... 0) }
identify    { diff = (0 1) sdiff = (0 1) }
```

Example 7 Estimate a model that involves a constant, fixed seasonal effects, and two user-defined regression variables. The data for the latter two variables is stored in the file `weather.dat` in the current directory. This file includes data on several other variables not being used in the model. The data for the two user-defined regression variables is extracted from this larger file using a FORTRAN format that skips the first 16 columns in the file. The start date is specified since the data set of user-defined regression variables begins before the data for the time series being modelled.

```
 series {title = "Monthly Riverflow"  start = 1970.1
          data = (8.234 8.209 ...  8.104) period = 12 }
 regression { variables = (seasonal const)
             user = (temp precip)
             file = "weather.dat"
             format = "(t17,2f8.2)"
             start = 1960.1 }
arima { model = (3 0 0)(0 0 0) }
estimate { }
```

Example 8 Estimate a model for a monthly retail inventory series with end-of-month stock trading-day effects and one AO. The transformed series, minus the regression effects, follows an ARIMA (0 1 0)(0 1 1)$_{12}$ model. Decide (using AICC) if the stock trading-day regressors should be kept in the model.

```
series {title = "Retail Inventory - Family Apparel"
        start = 1967.1  period = 12
        data = (1893 1932 ... 3201 ) }
transform { function = log }
regression { variables = (tdstock[31] ao1980.jul)
              aictd=tdstock }
arima { model = (0 1 0)(0 1 1) }
estimate    { }
```

Example 9 Estimate a model for a monthly retail sales series with stable seasonal and trading day regressors. Include regressors for a change-of-regime in both sets of regressors in December of 1985. The transformed series, minus the regression effects, follows an ARIMA (0 1 1) model.

```
series {title = "Retail Sales - Televisions"
        start = 1976.1  period = 12
        file = 'tvsales.ori'   }
transform { function = log }
regression { variables = (td/1985.dec/ seasonal/1985.dec/) }
arima { model = (0 1 1) }
estimate    { }
```

Example 10 Similar to example 9, only partial change-of-regime regressors are used in conjuction with the seasonal and trading day regressors so that the extra regressors are set to zero before December of 1985.

```
series {title = "Retail Sales - Televisions"
        start = 1976.1  period = 12
        file = 'tvsales.ori'   }
transform { function = log }
regression { variables = (td td//1985.dec/
                          seasonal seasonal//1985.dec/) }
arima { model = (0 1 1) }
estimate    { }
```

Example 11 Specified regression variables are a trend constant and trading day effects. Use the automatic modelling procedure to select an ARIMA model. Additively seasonally adjust the series after preadjusting for the trading day regression effects.

```
series     {  title = "Exports of pasta products"
              start = 1980.jan   data = "pasta.dat"  }
regression {  variables = (const td)                 }
automdl    {  mode = both                            }
x11        {  mode = add                             }
```

Example 12 The regression effects selected are seasonal means, a constant, several outliers, trading day, and an Easter effect. There are user-defined regression variables for special sales promotions in 1988, 1989 and 1990, which are located in the file `promo.dat` in 3f12.0 format. The ARIMA part of the model is (2,1,0). Seasonally adjust the series after pre-adjusting for all the regression effects. Remove the Easter effects and trading day effects from the final seasonally adjusted series. Generate 24 forecasts.

```
series{  title = "Retail sales of children’s apparel"
    file = "capprl.dat"    start = 1975.1           }
transform{      function = log     }
regression{
    variables = (const td ao1976.oct ls1991.dec easter[8]
                 seasonal)
    user = (sale88 sale89 sale90)
    start = 1975.1    file = "promo.dat"    format = "(3f12.0)"
}
arima{   model = (2 1 0)   }
forecast{      maxlead = 24        }
x11{   save=seasonal  appendfcst=yes  }
```

Example 13 The same as Example 12, except that the user-defined regression effect will be handled the same way as additive outliers with regard to prior adjustments, final adjustments, print files, and save files.

```
series{  title = "Retail sales of children’s apparel"
    file = "capprl.dat"    start = 1975.1           }
transform{      function = log     }
regression{
    variables = (const td ao1976.oct ls1991.dec easter[8]
                 seasonal)
    user = (sale88 sale89 sale90)
    start = 1975.1    file = "promo.dat"    format = "(3f12.0)"
    usertype = ao
}
arima{   model = (2 1 0)   }
forecast{      maxlead = 24        }
x11{   save=seasonal  appendfcst=yes   }
```

Example 14 Specify a regARIMA model with trading day and outlier terms. Specify starting values for the regression coefficients, and hold the coefficients of the outlier regressors fixed at these values. Use this model to generate 12 forecasts. Perform a default multiplicative seasonal adjustment, after prior adjustment for trading day and outlier factors.

```
series{
  period=12             format='2L'
  title='MidWest Total Starts'
  file='mwtoths.dat'    name='MWTOT '
}
transform{  function=log  }
arima{  model=(0 1 2 )(0 1 1 )  }
estimate{  save=mdl  }
forecast{     maxlead=12     }
regression{
  variables=(ao1977.jan ls1979.jan ls1979.mar ls1980.jan td)
  b=(  -0.7946F  -0.8739F  0.6773F  -0.6850F  0.0209
        0.0107   -0.0022   0.0018    0.0088  -0.0074       )
}
x11{  }
```

Example 15 Read in the data from a file using a predefined X-11 data format. Note that the starting date is taken from the information provided in the data file, so it does not have to be specified. Specify a regARIMA model with trading day and holiday terms. Perform automatic outlier identification, and print out model diagnostics. Use this model to generate 12 forecasts. Perform a multiplicative seasonal adjustment, using a 3x3 seasonal moving average, after prior adjustment for trading day, outlier and holiday factors. Remove the holiday and trading day factors from the final seasonally adjusted series. Save the trading day and holiday factors in individual output files.

```
Series { Format='1L'  File='bdptrs.dat'  Name='BDPTRS'
         Title='Department Store Sales'   }
Transform {  Function=Log  }
Regression {  Variables=( Td Easter[8] )
              Save = ( Td Holiday )  }
Arima {  Model=(0 1 1)(0 1 1)  }
Outlier {    }
Estimate {    }
Check {    }
Forecast {  }
X11 {   Mode = Mult   Seasonalma = S3X3
        Title=( 'Department Store Retail Sales Adjusted For'
                'Outlier, Trading Day, And Holiday Effects' )
}
```

SERIES

DESCRIPTION

Required spec that provides `X-12-ARIMA` with the time series data, a descriptive title for the series, the starting date of the series, the seasonal period (12 for monthly data, 4 for quarterly data,) and an optional restricted span (subset) within the time series to be used for the analysis. The data can either be included in the **series** spec by using the **data** argument, or they can be obtained from a file by using the **file** argument. Note that if `X-12-ARIMA` is run using a data metafile, the series should not be specified in this spec, since datafiles are being specified in the data metafile (for more details, see Section 2.5.2).

USAGE

```
series{title = "Example Series"
       start = 1967.1
       period = 12
       span = (1970.1,)
       modelspan = (1985.Jan, 0.Dec)
       name = "tstsrs"
       data = (480 ... 1386)   or   file = "example.dat"
                                    format = "2r"
       decimals = 2
       precision = 1
       comptype=add
       compwt=1.0
       spectrumstart = 1975.Jan
       print = (none +header)
       save = (spn)
      }
```

ARGUMENTS

comptype Indicates how a component series of a composite (also called aggregate) series is incorporated into the composite. These component series can be *added into* the (partially formed) composite series (`comptype=add`), *subtracted from* the composite series (`comptype=sub`), *multiplied by* the composite series (`comptype=mult`), or *divided into* the composite series (`comptype=div`). The default is no aggregation.

compwt Specifies that the series is to be multiplied by a constant before aggregation. This constant must be greater than zero (for example, `compwt=0.5`). This argument can only be used in conjunction with `comptype`. The default composite weight is one.

data Vector containing the time series data. The data are read row-wise in the following format: there must be at least one blank space, comma, or carriage return separating each of the data values. The number of observations is automatically determined as the length of the data vector supplied. If the **data** argument is used, the **file** argument cannot be used.

decimals Specifies the number of decimals that will appear in the seasonal adjustment tables of the main output file. This value must be an integer between 0 and 5, inclusive (for example, `decimals=5`). The default number of decimals is zero.

file Name of the file containing the time series data. The filename must be enclosed in quotes. If the file is not in the current directory, the complete filename including the path must be given. Valid path and filenames depend on the computer operating system. If the **file** argument is used, the **data** argument cannnot be used.

format Denotes the format to be used in reading the time series data from the named file, when the data are not in free format. Five types of input can be used:

(a) a valid FORTRAN format, which should be enclosed in quotes and must include the initial and terminal parentheses (example: `format="(6f12.0)"`);
(b) a two character code which corresponds to a set of data formats used in previous versions of X-11 and X-11-ARIMA (example: `format="1r"`);
(c) "datevalue" format, where the year, month or quarter, and value of each observation are found in this order in free format on individual lines. Thus, a line of the data file containing the value 32531 for July of 1991 would have the form `1991 7 32531`. The number of preceeding blanks can vary (example: `format="datevalue"`);
(d) the format `X-12-ARIMA` uses to save a table. This allows the user to read in a file saved from a previous `X-12-ARIMA` run (example: `format="x12save"`).
(e) the format that the `TRAMO` and `SEATS` programs use to read in a series and its descriptors. This enables `X-12-ARIMA` to read in a data file formatted for the `TRAMO` modelling program or the `SEATS` seasonal adjustment program. (example: `format="tramo"`).

In the predefined X-11 data formats mentioned in (b), the data is stored in 6 or 12 character fields, along with a year and series label associated with each year of data. For a complete list of these formats and how they are used, see DETAILS. If no format argument is given, the data will be read in free format. In *free format*, all numbers on a line will be read before continuing to the next line, and the numbers must be separated by one or more spaces (not by commas or tabs). **Format** cannot be used with the **data** argument, only with **file**.

modelspan Specifies the span (data interval) of the data to be used to determine all regARIMA model coefficients. This argument can be utilized when, for example, the user does not want data early in the series to affect the forecasts, or, alternatively, data late in the series to affect regression estimates used for preadjustment before seasonal adjustment. As with the **span** spec detailed above, the **modelspan** argument has two values, the start and end date of the desired span. A missing value defaults to the corresponding start or end date of the span of the series being analyzed. For example, for monthly data, the statement `modelspan=(1968.1, )` causes whatever regARIMA model is specified in other specs to be estimated from the time series data starting in January, 1968 and ending at the end date of the analysis span. A comma is necessary if either the start or end date is missing. The start and end dates of the model span must both lie within the time span of data specified for analysis in the **series** spec, and the start date must precede the end date.

Another end date specification, with the form *0.per*, is available to set the ending date of **modelspan** to always be the most recent occurrence of a specific calendar month (quarter for quarterly data) in the span of data analyzed, where *per* denotes the calendar month (quarter). Thus, if the span of data considered ends in a month other than December, `modelspan=(,0.dec)` will cause the model parameters to stay fixed at the values obtained from data ending in the next-to-final calendar year of the span.

name The name of the time series. The name must be enclosed in quotes and may contain up to 64 characters. Up to the first 16 characters will be printed as a label on every page. When specified with the predefined formats of the **format** argument, the first six (or eight, if `format="cs"`) characters of this name are also used to check if the program is reading the correct series, or to find a particular series in a file where many series are stored.

period Seasonal period of the series. If seasonal adjustments are generated, the only values currently accepted by the program are 12 for monthly series and 4 for quarterly series. Otherwise, any seasonal period up to 12 can be specified. (This limit can be changed—see Section 2.8.) The default value for **period** is 12.

precision The number of decimal digits to be read from the time series. This option can only be used with the predefined formats of the **format** argument. This value must be an integer between 0 and 5, inclusive (for example, `precision=5`). The default is zero. If **precision** is used in a **series** spec that does not use one of the predefined formats, the argument is ignored.

print and **save** Table 6-20 shows the tables available for output in the **series** spec.

Table 6-20: Available Output Tables for Series

name	*default*	*brief*	*ext*	*description of table*
header	+	+	·	summary of options selected for this run of X-12-ARIMA
span	+	+	a1	time series data, with associated dates (if the **span** argument is present, data are printed and/or saved only for the specified span)
seriesplot	·	·	·	plot of the original series
specfile	+	+	spc	contents of input specification file used for this run
savefile	+	+	·	list of files to be produced by the X-12-ARIMA run
specorig	+	·	sp0	spectral plot of the first-differenced original series
missingvaladj	+	+	mva	original series with missing values replaced by regARIMA estimates
outlieradjori	·	·	a19	original series adjusted for regARIMA outliers

Name gives the name of each table for use with the **print** and **save** arguments.
Default indicates which tables are printed (+) or not printed (·) by default.
Brief indicates which tables are printed (+) or not printed (·) when the **brief** print level is specified. See section 5.1 for more information on print levels.
Ext gives the file extension used if the table is saved. A dot indicates that the table cannot be saved. The file extensions given can also be used as short names for the tables in the **print** and **save** arguments.

savelog The only diagnostic available for output to the log file (see section 2.6) shows which spectral plots (if any) had visually distinct peaks. Specifying `savelog=peaks` or `savelog=spk` will store this information into the log file.

span Limits the data utilized for the calculations and analyses to a span (data interval) of the available time series. The **span** argument has two input values, the start and end date of the span. A missing value defaults to the corresponding start or end date of the input time series. For example, assuming monthly data, the statement `span=(1968.1, )` specifies a span starting in January, 1968 and ending at the end date of the series input through the **data** or **file** argument. A comma is necessary if either the start or end date is missing. The start and end dates of the span must both lie within the series, and the start date must precede the end date.

spectrumstart The starting date of the span of data to be used to estimate the spectra the original, seasonally adjusted, and modified irregular series. This date must be in the format `spectrumstart=`*year.seasonal period*. This can be used to determine if there are residual trading day or seasonal effects in the adjusted data from, say, the last seven years. Residual effects can occur when seasonal or trading day patterns are evolving. The default starting date for the spectral plots of monthly series is set to be eight years from the end of the series, unless the span of data to be analyzed is less than eight years long, in which case, it is set to the starting date of this span of data. For quarterly series, the default starting date is set to the starting date of the span of data. Example: `spectrumstart=1987.Jan`.

start The start date of the time series in the format `start=`*year.seasonal period*. (See Section 5.2 and the examples below.) The default value of **start** is `1.1`. (See DETAILS.)

title A title describing the time series. The title must be enclosed in quotes and may contain up to 79 characters. It will be printed on each page of the output (unless the **-p** option is evoked; see Section 2.7).

RARELY USED ARGUMENTS

diffspectrum If `diffspectrum=no`, the spectrum of the (transformed) original series or seasonally adjusted series is calculated. The default (`diffspectrum=yes`) produces the spectrum of the month-to-month (quarter-to-quarter) differences of these series.

divpower An integer value used to rescale the input time series prior to analysis. The program divides the series by ten raised to the specified value. For example, setting `divpower = 2` will divide the original time series by 10^2, while `divpower = -4` will divide the series by 10^{-4}. Integers from -9 to 9 are acceptable values for **divpower**. If this option is not specified, the time series will not be rescaled.

missingcode A numeric value in the input time series that the program will interpret as a missing value. This option can only be used in input specification files requiring a regARIMA model to be estimated or identified automatically. The default value is -99999. Example: `missingcode=0.0`.

saveprecision The number of decimals stored when saving a table to a separate file with the **save** argument. The default value of **saveprecision** is 15. Example: `saveprecision=10`.

spectrumtype The type of spectral estimate used in the spectral plots produced by `X-12-ARIMA`. If `spectrumtype = periodogram`, the periodogram of the series is calculated and plotted. The default (`spectrumtype=arspec`) produces an autoregressive model spectrum of the series.

trimzero If `trimzero=no`, zeroes at the beginning or end of a time series entered via the file argument are treated as series values. The default (`trimzero=yes`) causes leading and trailing zeros to be ignored. Note that when the format argument is set to either `datevalue`, `x12save`, or `tramo`, all values input are treated as series values, regardless of the value of trimzero.

yr2000 If `yr2000=yes`, a "century cutoff" for 2-digit years from data stored in "X-11 formats" is set at 1945. Years 00-45 are interpreted as 20xx, and years 46-99 are interpreted as 19xx. This is the default for the program.

If `yr2000=no`, the program assumes all 2-digit years fall in the 20th century and will convert them to 4-digit years accordingly.

DETAILS

The number of observations and the series end date are determined by the program after reading in the data. `X-12-ARIMA` accepts a maximum of 600 observations. (This limit can be changed—see Section 2.8.)

If spec files are copied from one directory to another or from one computer system to another, verify that the path and filenames in their **file** arguments remain valid.

The **series** spec cannot appear in a spec file with the **composite** spec. The latter signifies that a seasonal adjustment of a composite series is to be calculated.

The following table gives a description of the default formats for each of the valid two-character X-11 format codes for the **format** argument, as well as the corresponding FORTRAN format.

Table 6-21: Default Formats for Each X-11 Format Code

Code	*FORTRAN Format for Monthly Data*	*FORTRAN Format for Quarterly Data*	*Description*
1r	(12f6.0,I2,A6)	(4(12x,f6.0),I2,A6)	Year and identifier on the right, data in 6-digit fields
2r	(6f12.0,/,6f12.0,I2,A6)	(4f12.0,24x,I2,A6)	Year and identifier on the right of the second line, data in 12-digit fields
1l	(A6,I2,12f6.0)	(A6,I2,4(12x,f6.0))	Year and identifier on the left, data in 6-digit fields
2l	(A6,I2,6f12.0,/,8x,6f12.0)	(A6,I2,4f12.0)	Year and identifier on the left of the first line, data in 12-digit fields
2l2	(A8,I4,6f11.0,2x,/,12x,6f11.0,2x)	(A8,I4,4f11.0,2x)	Four digit year and identifier on the left of the first line, data in 11-digit fields
cs	(A8,I2,10X,12E16.10,18X)	(A8,I2,10X,12E16.10,18X)	Data in CANSIM data base utility format, data in 16-digit fields
cs2	(A8,I4,12X,12E16.10,14X)	(A8,I4,12X,12E16.10,14X)	Data in the new CANSIM data base utility format (called CANSIM2), data in 16-digit fields

These formats can be modified by using the **decimals** argument. If **decimals** is used in a **series** spec that doesn't use an X-11 format code, this argument is ignored.

Note that if one of the X-11 format codes is specified (or if `format = "datevalue"` or `format = "x12save"`), the start of the series is automatically read from the data file. Therefore, the starting date need not be specified with the **start** argument of the **series** spec.

If a data metafile is used to process a group of input files using a single input spec file, the X-11 formats should be avoided. These formats require the name of the series (specified **name**) to verify that the data is in the file. This implies that all the data files in the data metafile would be required to use the same series name. This is often not desirable.

When doing a *formatted* read of a data *file*, `X-12-ARIMA` discards sequences of zeroes at the ends of the series (unless `trimzero=no`). This convention is used to allow input of series stored in certain formats—Example 3 below gives an illustration. If the zeros at the ends of the series are true data values, `trimzero=no` will cause them to be treated as such. However, if the zeroes at the beginning of a given series are real and the zeroes implied at the end of the series are not (due to blanks at the end of the line), then the file must be modified so that it can be read in free format. Example 4 below demonstrates this conversion.

The **span** and **modelspan** arguments can be used with the **forecast** spec to generate *out-of-sample* forecast comparisons by excluding data at the end of the series. When either of these arguments are present, model estimation will use data only for the specified span. Forecasting then (by default) proceeds from the end of the span, producing comparisons of the withheld data with the forecasts. (See Example 4 of the **forecast** spec.)

Note that if the beginning date specified in the **modelspan** argument is not the same as the starting date in the **span** argument, backcasts cannot be generated by the program, regardless of the value of the **maxback** argument of the **forecast** spec.

When the program encounters a value equal to the value of **missingcode** in the original series, it inserts an additive outlier for that observation time into the set of regression variables of the model the series and then replaces the missing value code with a value large enough to be considered an outlier during model estimation. After the regARIMA model is estimated, the program adjusts the original series using factors generated from these missing value outlier regressors. The adjusted values are estimates of the missing values.

EXAMPLES

Note: The following examples, except for Example 8, do not show "complete" spec files in the sense that useful output is not produced unless additional specs (e.g., **x11** or **arima** and **estimate**) are also included.

Example 1 Specify a time series with the **data** argument.

```
series{
      title="A Simple Example"
      start=1967.jan                # period defaults to 12
      data=( 480 467 514 505 534 546 539 541 551 537 584 854
             522 506 558 538 605 583 607 624 570 609 675 861
              .
              .
              .
            1684 1582 1512 1508 1574 2303 1425 1386) }
```

Example 2 Drop observations from both the beginning and end of a quarterly series that starts in 1940 and ends in 1993. The first six years of data are dropped to restrict the analysis to post-WWII data. The data held out for 1991–93 could be used to examine out-of-sample forecast performance.

```
series { data = (879 899 985 ...)  # There are 216 data values
        start = 1940.1             #       ending in 1993.4
        period = 4                 # Quarterly series
        span = (1946.1, 1990.4) }
```

Example 3 This example shows how the X-12-ARIMA program can read data from files stored in a format adopted from the X-11-ARIMA seasonal adjustment program. Here the data are available from July, 1970 through February, 1993, and are stored in the file `c:\data\sales1.dat` as follows:

```
                                    146.4 109.2 132.1 144.8 116.1 100.370SALES1
 142.9 158.8 196.2 244.0 251.6 245.5 244.2 213.8 188.9 197.2 181.2 161.371SALES1
   .
   .
   .
 148.8 177.2     0     0     0     0     0     0     0     0     0     093SALES1
```

The data are stored in `(12f6.1,i2,a6)` format, with the last eight columns in each line providing the year and series ID.

```
SERIES{ TITLE = "Monthly data in an X-11 format"
        PERIOD = 12
        FILE = "C:\DATA\SALES1.DAT"    # a DOS path and file
        PRECISION = 1
        FORMAT = "1r" }
```

Since FORTRAN formatted reads treat blanks as zeroes, the input of the series obtains six zeroes at the beginning. The input series also contains the ten zeroes at the end. As noted in DETAILS, `X-12-ARIMA` discards the zeroes read in from both the beginning and end of the series by default so that only the actual data are retained and assigned to the correct months (146.4 to July, 1970, etc.). Also note that since the year is given on each line, the user does not have to enter a **start** argument.

Example 4 This example illustrates the rare case of a data file that must be modified for correct input to `X-12-ARIMA` . The original data file contains data for February, 1980 through November, 1990 stored in `(6f4.0,1x,i4)` format as follows.

```
       0 342-256 491   0 0001
-234 922-111   2   0 199 0002
  .
  .
  .
 581-987-423  10   0     0022
```

This file cannot be read in free format because several of the data entries run together and because the file contains record counters (0001, 0002, . . .) in columns 26–29. A free format read would treat the record counters as data. The file cannot be read with `(6f4.0)` format with a start date of February, 1980 because `X-12-ARIMA` with the default `trimzero = yes` would incorrectly drop the zeroes at the first and last observations, and then erroneously assign the value 342 to February 1980. Using `trimzero = no` would add extra zeroes to the series, as the blank spaces at the beginning and end of the data set would be read as zero.

The solution is to reformat the data file so it can be read in free format. This requires removal of the record counters and separation of the data entries. The modified file, `example4.new`, is as follows:

```
        0  342 -256  491    0
-234  922 -111    2    0  199
  .
  .
  .
 581 -987 -423   10    0
```

Then the following **series** spec will correctly read the data from the file `example4.new`.

```
series {title = "Data read correctly in with trimzero = no"
        start = 1980.2   period = 12
        file = "example4.new" }    # file is in current directory
```

Example 5 This example shows how the X-12-ARIMA program can read data in "date-value" format. The data are available from July, 1970 through February, 1993, and are stored in the file `c:\data\sales1.edt` as follows:

```
1970   7   14624
1970   8   10952
1970   9   13251
1970  10   14408
  .
  .
  .
1993    1  14838
1993    2  17762
```

Each data record contains the year, month and value of a given observation of the time series.

```
SERIES{ TITLE = "Monthly data in a datevalue format format"
        PERIOD = 12
        FILE = "C:\DATA\SALES1.EDT"    # a DOS path and file
        FORMAT = "DATEVALUE" }
```

Note that as in the X-11-ARIMA format shown in Example 3 above, the starting date can be read directly from the input file, so the user does not have to include a **start** argument.

Example 6 The same as example 5, but this series will be used as a component in a composite adjustment. The number of decimals displayed in the output is set to be 2, and the span of data to be modelled will be set to be the start of the series through December, 1992

```
SERIES{ TITLE = "Monthly data in a datevalue format format"
        PERIOD = 12
        FILE = "C:\DATA\SALES1.EDT"    # a DOS path and file
        FORMAT = "DATEVALUE"
        COMPTYPE = ADD
        DECIMALS = 2
        MODELSPAN = (,1992.DEC)
      }
```

Example 7 This example shows how the X-12-ARIMA program handles missing data. The same data format is used as in the previous two examples, except a missing value code is inserted for January of 1990:

```
1970   7   14624
1970   8   10952
1970   9   13251
1970  10   14408
  .
  .
  .
1990    1  -99999.
  .
  .
  .
1993    1  14838
1993    2  17762
```

The series spec below will replace the missing value code for January 1990 with a number large enough to be considered an outlier, assuming a regARIMA model is estimated later in the input specification file.

```
SERIES{ TITLE = "Monthly data in a date-value format format"
        PERIOD = 12
        FILE = "C:\DATA\SALES1.EDT"    # a DOS path and file
        FORMAT = "DATEVALUE"
}
```

Example 8 This example shows how to obtain a spectrum plot of the first differences (month-to-month differences) of the logarithms of the series to check if the series has seasonal or trading day effects. This is a complete spec file.

```
series{ title = "Spectrum analysis of Building Permits Series"
        start = 1967.Jan
        file = "permits.dat"
        format = "(12f6.0)"
        spectrumstart = 1987.Jan
        print = (none +specori)
      }
transform{   function = log  }
```

Example 9 This example shows how the X-12-ARIMA program can read data from a file previously saved by X-12-ARIMA. In a previous run, the outlier adjusted original series was stored in the file c:\data\sales1.a11.

```
SERIES{ TITLE = "Monthly data in a file saved by X-12-ARIMA"
        PERIOD = 12
        FILE = "C:\DATA\SALES1.A11"    # a DOS path and file
        FORMAT = X12SAVE }
```

Note that as in the `X-11-ARIMA` format shown in Example 3 and the "datevalue" format shown in Example 5 above, the starting date can be read directly from the input file, so a **start** argument is not included.

SLIDINGSPANS

DESCRIPTION

Optional spec providing sliding spans stability analyses. These compare different features of seasonal adjustment output from overlapping subspans of the time series data. The user can specify options to control the starting date for sliding spans comparisons (**start**), the length of the sliding spans (**length**), the threshold values determining sliding spans statistics (**cutseas**, **cutchng**, **cuttd**), how the values of the regARIMA model parameter estimates will be obtained during the sliding spans seasonal adjustment runs (**fixmdl**), and whether regARIMA automatic outlier identification is performed (**outlier**).

USAGE

```
slidingspans {start = 1975.jan
              length = 132
              numspans = 3
              cutseas=3.0
              cutchng=3.0
              cuttd=2.0
              outlier = yes
              fixmdl = no
              fixreg = outlier
              print = (long -ssheader)
              save = (sfspans)
              savelog = (percent)
             }
```

ARGUMENTS

cutchng Threshold value for the month-to-month, quarter-to-quarter, or year-to-year percent changes in seasonally adjusted series. For a month (quarter) common to more than one span, if the maximum absolute difference of its period-to-period percent changes from the different spans exceeds the threshold value, then the month (quarter) is flagged as having an unreliable estimate for this period-to-period change. This value must be greater than 0; the default value is 3.0. Example: `cutchng=5.0`

cutseas Threshold value for the seasonal factors and seasonally adjusted series. For a month (quarter) common to more than one span, if the maximum absolute percent change of its estimated seasonal factors or adjustments from the different spans exceeds the threshold value, then this month's (quarter's) seasonal factor or adjustment is flagged as unreliable. This value must be greater than 0; the default value is 3.0. Example: `cut-seas=5.0`

cuttd Threshold value for the trading day factors. For a month (quarter) common to more than one span, if the maximum absolute percent change of its estimated trading day factors from the different spans exceeds the threshold value, then this month's (quarter's) trading day factor is flagged as unreliable. This value must be greater than 0; the default value is 2.0. Example: `cuttd=1.0`

fixmdl Specifies how the initial values for parameters estimated in regARIMA models are to be reset before seasonally adjusting a sliding span. This argument is ignored if a regARIMA model is not fit to the series.

If `fixmdl=yes`, the values for the regARIMA model parameters for each span will be set to the parameter estimates taken from the original regARIMA model estimation. These parameters will be taken as fixed and not reestimated. This is the default for **fixmdl**.

If `fixmdl=no`, the program will restore the initial values to what they were when the regARIMA model estimation was done for the complete series. If they were fixed in the **estimate** spec, they remain fixed at the same values.

If `fixmdl=clear`, initial values for each span will be set to be the defaults, namely 0.1 for all coefficients, and all model parameters will be reestimated.

fixreg Specifies the fixing of the coefficients of a regressor group in either a regARIMA model or an irregular component regression. These coefficients will be fixed at the values obtained from the model span (implicit or explicitly) indicated in the series or composite spec. All other regression coefficients will be reestimated for each sliding span. Trading day (**td**), holiday (**holiday**, outlier (**outlier**), or other user-defined (**user**) regression effects can be fixed. This argument is ignored if niether a regARIMA model nor an irregular component regression is fit to the series, or if `fixmdl=yes`.

length The length of each span, in months or quarters (in accordance with the sampling interval) of time series data used to generate output for comparisons. A length selected by the user must yield a span greater than 3 years long and less than 17 years long. If the length of the span is not specified by the user, the program will choose a span length based on the length of the seasonal filter selected by the user (or by the program if a seasonal filter was not specified by the user). Monthly data example: `length=96`

numspans Number of sliding spans used to generate output for comparisons. The number of spans selected by the user must be between 2 and 4, inclusive. If this argument is not specified by the user, the program will choose the maximum number of spans (up to 4) that can be formed based on the length of the sliding spans given by the user (or selected by the program if the **length** argument is not used). Example: `numspans=4`

outlier Specifies that automatic outlier detection is to be performed when the regARIMA model is reestimated during the processing of each span. This argument has no effect if the **outlier** spec is not used.

If this argument is not specified, automatic outlier identification will not be performed during any regARIMA model estimation that occurs for each of the sliding spans. If outlier terms are included in the **regression** spec, these will included in the model estimated for the spans, but any outliers identified by the automatic outlier identification procedure will be removed.

If `outlier=yes`, the program performs automatic outlier identification whenever a regARIMA model is estimated for a span of data.

If `outlier=keep`, the program carries over any outliers automatically identified in the original estimation of the regARIMA model for the complete time series, and does not perform automatic outlier identification when a regARIMA model is estimated for one of the sliding spans. If the date of an outlier detected for the complete span of data does not occur in one of the sliding spans, the outlier will be dropped from the model for that span.

print and **save** The tables available for output are listed on Table 6-22.

savelog The only diagnostic available for output to the log file (see section 2.6) is the percentage of observations flagged as unstable for each of the estimates from the seasonal adjustment estimates tested by the sliding spans analysis.

Specifying `savelog=percent` or `savelog=pct` will store this information into the log file.

start The starting date for sliding spans comparisons. The default is the beginning month of the second span. Example: `start=1990.jan`

RARELY USED ARGUMENTS

additivesa Specifies whether the sliding spans analysis of an additive seasonal adjustment will be calculated from the maximum differences of the seasonally adjusted series (`additivesa = difference`) or from the maximum of an implied adjustment ratio of the original series to the final seasonally adjusted series (`additivesa = percent`). This option will also determine if differences (`additivesa = difference`) or percent changes

Table 6-22: Available Output Tables for Slidingspans

name	*default*	*brief*	*ext*	*description of table*
header	+	+	·	header text for the sliding spans analysis
ssftest	+	·	·	F-tests for stable and moving seasonality estimated over each of the sliding spans
factormeans	+	·	·	range analysis for each of the sliding spans
percent	+	+	·	table showing the percent of observations flagged as unstable for the seasonal and/or trading day factors, final seasonally adjusted series (if necessary), and the month-to-month (or quarter-to-quarter) changes
yypercent	·	·	·	additional entry for the percent of observations flagged as unstable for the year-to-year changes
summary	+	·	·	tables, histograms and hinge values summarizing the percentage of observations flagged for unstable seasonal and/or trading day factors, final seasonally adjusted series (if necessary), and month-to-month (or quarter-to-quarter) changes
yysummary	·	·	·	additional tables, histograms and hinge values summarizing the percentage of observations flagged for the year-to-year changes
sfspans	·	·	sfs	seasonal factors from all sliding spans
chngspans	·	·	chs	month-to-month (or quarter-to-quarter) changes from all sliding spans
saspans	·	·	sas	seasonally adjusted series from all sliding spans
ychngspans	·	·	ycs	year-to-year changes from all sliding spans
tdspans	·	·	tds	trading day factors from all sliding spans
indfactormeans	+	+	·	range analysis for the implicit adjustment factors of the indirectly seasonally adjusted series
indpercent	+	+	·	tables of the percent of observations flagged as unstable for the seasonal factors and month-to-month (or quarter-to-quarter) changes of the indirect seasonal adjustment
indyypercent	·	·	·	additonal entry for the percent of observations flagged as unstable for the year-to-year (or quarter-to-quarter) changes of the indirect seasonal adjustment
indsummary	+	·	·	tables, histograms and hinge values summarizing the percentage of observations flagged for unstable seasonal factors, month-to-month (or quarter-to-quarter) and year-to-year changes for the indirect adjustment
indyysummary	·	·	·	additional tables, histograms and hinge values summarizing the percentage of observations flagged for the year-to-year changes of the indirect seasonal adjustment

Name gives the name of each table for use with the **print** and **save** arguments.
Default indicates which tables are printed (+) or not printed (·) by default.
Brief indicates which tables are printed (+) or not printed (·) when the **brief** print level is specified. See section 5.1 for more information on print levels.
Ext gives the file extension used if the table is saved. A dot indicates that the table cannot be saved. The file extensions given can also be used as short names for the tables in the **print** and **save** arguments.

Table 6-22: Available Output Tables for Slidingspans (continued)

name	*default*	*brief*	*ext*	*description of table*
indsfspans	·	·	sis	indirect seasonal factors from all sliding spans
indchngspans	·	·	cis	indirect month-to-month (or quarter-to-quarter) changes from all sliding spans
indsaspans	·	·	ais	indirect seasonally adjusted series from all sliding spans
indycngspans	·	·	yis	indirect year-to-year changes from all sliding spans

Name gives the name of each table for use with the **print** and **save** arguments.
Default indicates which tables are printed (+) or not printed (·) by default.
Brief indicates which tables are printed (+) or not printed (·) when the **brief** print level is specified. See section 5.1 for more information on print levels.
Ext gives the file extension used if the table is saved. A dot indicates that the table cannot be saved. The file extensions given can also be used as short names for the tables in the **print** and **save** arguments.

(`additivesa = percent`) are generated in the analysis of the month-to-month, quarter-to-quarter, or year-to-year changes in seasonally adjusted series. The default is `additivesa = differences`. If the seasonally adjusted series for any of the spans contains values that are less than or equal to zero, the sliding spans analysis will be performed on the differences.

fixx11reg Specifies whether the irregular component regression model specified in the **x11regression** spec will be reestimated during the sliding spans analysis. If `fixx11reg=yes`, the regression coefficients of the irregular component regression model are fixed throughout the analysis at the values estimated from the entire series. If `fixx11reg=no`, the irregular component regression model parameters will be restimated for each span. The default is `fixx11reg=yes`.

x11outlier Specifies whether the AO outlier identification will be performed during the sliding spans analysis for all irregular component regressions that result from the **x11regression** spec. If `x11outlier=yes`, AO outlier identification will be done for each span. Those AO outlier regressors that were added to the irregular component regression model when automatic AO outlier identification was done for the full series are removed from the irregular component regression model prior to the sliding spans run. If `x11outlier=no`, then the automatically identified AO outlier regressors for the full series are kept for each sliding spans run. If the date of an AO outlier detected for the complete span of data does not occur in one of the sliding spans, the outlier will be dropped from the model for that span. The coefficients estimating the effects of these AO outliers are reestimated whenever the other irregular component regression model parameters are reestimated. However, no additional AO outliers are automatically identified and estimated. This option is ignored if the **x11regression** spec is not used, if the selection of the **aictest** argument results in the program not estimating an irregular component regression model, or if the **sigma** argument is used in the **x11regression** spec. The default is `x11outlier=yes`.

DETAILS

This section is divided into two subsections, the first providing a description of the diagnostics and the second some additional information about the arguments.

The main diagnostics and their interpretation

The sliding spans diagnostics are described in detail and compared with other quality diagnostics in the articles Findley, Monsell, Shulman and Pugh (1990) and Findley and Monsell (1986). An abbreviated presentation will be given here. The basic diagnostics are descriptive statistics of how the seasonal adjustments and their month-to-month changes vary when the span of data used to calculate them is altered in a systematic way: any two neighboring spans differ to the extent that one starts and ends a year later than the other. The span length is determined by the length of the seasonal filter utilized for the adjustment. The ending date of the last span is usually the date of the most recent datum in the time series. Four spans are used if enough data are available. The index value $j = 1$ is assigned to the span with the earliest starting date, $j = 2$ to the span with the next earliest starting date, $\ldots$.

For series whose seasonally adjusted values are all positive, the two most important sliding spans statistics, $A(\%)$ and $MM(\%)$, are calculated as follows. For a month t belonging to at least two spans, one of which is the j-th span, let A_t denote its seasonally (and, if applicable, trading day and holiday) adjusted value obtained from the complete series, and let A_t^j denote the adjusted value obtained when the seasonal adjustment procedure being considered (the procedure determined by the software options selected) is applied to only data in the j-th span. The seasonal adjustment A_t is called (unacceptably) unstable if

$$\frac{\max_j A_t^j - \min_j A_t^j}{\min_j A_t^j} > .03. \tag{1}$$

Further, for months t such that both t and $t-1$ belong to at least two spans, the "seasonally adjusted month-to-month percent change" $100 \times \left(A_t - A_{t-1}\right)/A_{t-1}$ is called unstable if

$$\max_j \frac{A_t^j}{A_{t-1}^j} - \min_j \frac{A_t^j}{A_{t-1}^j} > .03. \tag{2}$$

In (1), the index j ranges over all spans containing month t; in (2), the j-th span must contain month $t-1$ also.

$A\,(\%)$ is used to denote the percent of months with unstable adjustments calculated with respect to the number of month for which the left hand side of (1) is defined (the number of months common to at least two spans). The analogous quantity for (2) is denoted $MM(\%)$. We recommend that, except in special circumstances of the sort discussed below, the seasonal adjustment produced by the procedure chosen should not be used if $A\,(\%) > 25.0$ (> 15.0 is considered problematic) or if $MM(\%) > 40.0$.

There is a similarly defined statistic $YY(\%)$ for year-to-year percent changes in the seasonally adjusted data, $100 \times (A_t - A_{t-12})\,/A_{t-12}$, based on the same threshold used to define unstable adjustments and month–to-month changes, usually the default .03 shown in (1) and (2). Because these year-to-year changes in the adjusted series can be misleading indicators of trend direction when turning points occur between months t and $t-12$, they are rather less important than the adjusted values themselves and month-to-month changes in the adjusted values. The statistic $YY(\%)$ is correspondingly less important than the others, but it is included in the output of X-12-ARIMA because of the interest some data users have in year-to-year changes. The output text describes values of $YY(\%)$ greater than 10.0 as extreme, but this information is usually redundant in the sense that series with such a value have, in our experience, usually also had excessive values of $A(\%)$ or $MM(\%)$. In any case, we would not reject an adjustment based solely on the value of $YY(\%)$.

Sometimes, the causes of large values of $A(\%)$ or $MM(\%)$ can be identified and seen to be not very problematic. For example, this could be the case when the months with unstable adjustments or changes are heavily concentrated in a known problem period several years back from the current year, or in one or two fixed calendar months each year that all data users can be expected to regard as quite problematic, such as winter months in series known to be very sensitive to differences in winter weather conditions. The sliding spans output makes it easy to identify such concentrations.

The output can show when a "mild" increase in the threshold beyond 0.03 will dramatically decrease the values of $A(\%)$ and $MM(\%)$ to quite acceptable levels: we have identified a few series for which increasing the threshold to .05 seemed justifiable, because most of the months for which the left hand sides of (1) and (2) were between .03 and .05 were months with very large seasonal movements, where users would be tolerant of more uncertainty, and not many months had values of these statistics substantially larger than .05.

This experience stimulated us to carry out a limited exploratory study with a variety of Census Bureau series focussed on the goal of finding a statistical relationship between appropriate threshold values and seasonal factor size, a relationship we could use to increase or decrease the threshold according to the size of the seasonal movements. However, within the set of series considered, we found no correlation between appropriate threshold values and the size of the seasonal movements. For example, there were relatively many series with quite large seasonal movements for which good values of $A(\%)$ and $MM(\%)$ obtained with the .03 threshold and there were a number of series with only moderately large seasonal movements for which the use of the .05 threshold did not lead to acceptable values of $A(\%)$ and $MM(\%)$. In fact, simulation experiments readily show that in a series with fixed seasonal effects (every January has the same seasonal factor, etc.), the values of the seasonal adjustment are quite sensitive to the variability of the irregulars component and quite insensitive to the size of the seasonal movements.

More often than not, when a choice of adjustment options for a series produces an adjustment that sliding spans diagnostics classify as unacceptable, there will be a different choice of options, perhaps with different seasonal filter lengths, or different trading day adjustment or forecast extension options, that will result in an adjustment that is classified as acceptable. When no choice of options produces an acceptable adjustment, the issue is not whether the series is "seasonal" in some sense, but whether its seasonal behavior is repetitive enough, or revealed clearly enough in the available time series data, that it can be estimated with adequate reliably by X-12-ARIMA under any of the options considered.

More information about the arguments

Different adjustment quantities are examined in a sliding spans analysis, depending on the mode of the seasonal adjustment and whether trading day adjustment is done. For a multiplicative or log-additive seasonal adjustment, the seasonal factors, and the month-to-month and year-to-year changes of the seasonally adjusted series are analyzed. For a multiplicative or log-additive seasonal and trading day adjustment, the trading day factors and seasonally adjusted series are analyzed as well. For an additive seasonal adjustment without trading day adjustment, the seasonally adjusted series and the month-to-month and year-to-year changes of the seasonally adjusted series are analyzed. If trading day adjustment is done, these analyses are performed for the seasonal and trading day adjusted series. **WARNING:** In the additive adjustment case, the presence of relatively small values or negative values in the adjusted series can render unusable the percent change values which are the basis of almost all of the sliding spans statistics. In this situation, usually only a subjective analysis of the spans of adjusted series obtained by using **saspan** in the **print** or **save** arguments can be used to detect excessive instability. Further research is needed to develop more useful sliding spans statistics for additive adjustments.

For more details on the sliding spans procedure, see Findley, Monsell, Shulman, and Pugh (1990).

If the automatic ARIMA modelling option is specified in the **arima** spec, then the model selected by the procedure is used for all the sliding spans. If no model is selected by the procedure, then no model will be estimated during the sliding spans analysis.

While many of the tables in this spec cannot be saved as individual files, specifying the seasonal adjustment diagnostic summary option with the **-s** flag at runtime allows the user to store information from the sliding spans analysis into a diagnositc summary file (with the file extension **.xdg**). In addition, the **savelog** argument can write selected diagnostics into the log file for a given run (with the file extension **.log**). For more information, see Section 2.

If a sliding spans analysis of the direct and indirect adjustments of a composite seasonal adjustment is desired, the sliding spans analysis option must be specified for each of the component series. If the seasonal filter length is not the same for each component, then the user will have to use the **length** argument defined above in each of the input files of the component series to ensure that the spans analyzed for these series are of the same length.

If the automatic seasonal filter selection option is used, the seasonal filters used to generate the original seasonal adjustment will be used for the seasonal adjustment of each of the spans.

If an outlier specified by the user does not occur in a given span, that outlier will be dropped from the model for that span, and will be re-introduced into the regARIMA model if it is defined in future spans. User-defined regressors are checked to see if they are not constant in each span (i.e., all values of the regressor equal to zero).

EXAMPLES

The following examples show complete spec files.

Example 1 Multiplicative monthly seasonal adjustment, 3x9 seasonal factors for all calendar months. Sliding spans analysis performed with default settings for all options.

```
SERIES { FILE = "TOURIST.DAT"   START = 1976.1   }
X11 {   SEASONALMA =  S3X9    }
SLIDINGSPANS {       }
```

Example 2 Log-additive seasonal adjustment of quarterly data, 3x9 seasonal filters for first two quarters, 3x5 seasonal filters for last two quarters, 7-term Henderson trend filter. Sliding spans analysis performed with threshold values for selected tests set to 5.0.

```
Series       {
    File = "qstocks.dat"
    Start = 1967.1
    Title = "Quarterly stock prices on NASDAC"
    Freq = 4
}
X11          {
    Seasonalma = (  S3x9 S3x9 S3x5 S3x5  )
    Trendma = 7
    Mode = Logadd
}
Slidingspans {
    Cutseas = 5.0
    Cutchng = 5.0
}
```

Example 3 Seasonal ARIMA model with regression variables used for trading day adjustment and for automatic outlier identification and adjustment. Specified regression variables are a constant, trading day effects, and ramp between May 1982 and Sept. 1982. The ARIMA part of the model is (0,1,2)(1,1,0)12. Additively seasonally adjust the series after preadjusting for the outliers and the trading day regression effects. Perform sliding spans analysis; incorporate any outliers found by the application of the automatic identification procedure to the full series into the regARIMA model reestimated for each of the sliding spans.

```
series       {  title = "Number of employed machinists"
                start = 1980.jan  file = "machine.emp"
             }
regression   {  variables = (const td rp82.may-82.oct)  }
arima        {  model = (0 1 2)(1 1 0)  }
outlier      {  }
estimate     {  }
check        {  }
forecast     {  }
x11          {  mode = add  }
slidingspans {  outlier = keep  }
```

Example 4 The predefined regression effects to be estimated are a constant, trading day and a fixed seasonal. The ARIMA part of the model is $(3, 1, 0)$. Generate 60 forecasts. Seasonally adjust the series after pre-adjusting for the estimated trading day. Perform sliding spans analysis. Re−estimate the values of the REGARIMA model parameters for each span.

```
series { title = "Cheese Sales in Wisconsin"
   file = "cheez.fil"   start = 1975.1   }
transform {  function = log  }
regression {  variables = (const seasonal tdnolpyear)  }
arima {  model = (3 1 0)  }
forecast {   maxlead = 60  }
x11 {  save = seasonal   appendfcst = yes  }
slidingspans  {  fixmdl = no  }
```

Example 5 Sliding spans analysis will be performed on the multiplicative seasonal adjustment specified, using 3 sliding spans of length forty quarters as specified. This would allow the user to get some indication of seasonal adjustment stability, even though the series is not long enough for a complete sliding spans analysis with spans of the length most appropriate for 3x9 seasonal filters (44 quarters).

```
Series       {
    File = "qstocks.dat"
    Start = 1987.1
    Title = "Quarterly stock prices on NASDAC"
    Freq = 4
}
X11          {
    Seasonalma = S3x9
}
Slidingspans {
    Length = 40
    Numspans = 3
}
```

TRANSFORM

DESCRIPTION

Specification used to transform or adjust the series prior to estimating a regARIMA model. With this spec the series can be Box-Cox (power) or logistically transformed, length-of-month adjusted, and divided by user-defined prior-adjustment factors. Data for any user-defined prior-adjustment factors must be supplied, either in the **data** argument, or in a file specified by the **file** argument (not both). For seasonal adjustment, a set of permanently removed factors can be specified and also a set of factors that are temporarily removed until the seasonal factors are calculated.

USAGE

```
transform {function = log  or  power = 0.0
           adjust = lom
           title = "prior adjustment factors"
           start = 1975.jan
           data = (1.25 ... 1.90)  or  file = "prioradj.dat"
                                       format = "(6f12.3)"
           name = "Adjfac"
           mode = ratio
           type = temporary
           print = (none)
           save = (prioradj)
           savelog = atr
           }
```

ARGUMENTS

adjust Perform length-of-month adjustment on monthly data (`adjust = lom`), length-of-quarter adjustment on quarterly data (`adjust = loq`), or leap year adjustment of monthly or quarterly data (`adjust = lpyear`). (See DETAILS.)

Do not use the **adjust** argument if `td` or `td1coef` is specified in the **variables** argument of the **regression** or **x11regression** specs, or if additive or pseudo-additive seasonal adjustment is specified in the **mode** argument of the **x11** spec. Leap year adjustment (`adjust = lpyear`) is only allowed when a log transformation is specified in either the **power** or **function** arguments.

data An array containing one or two series of preadjustment factors which, unless `mode=diff` (see below), must have positive values intended for division into the corresponding values of the input time series. The default value is a vector of ones (no prior adjustment). When **data** (or **file**) is used, an adjustment factor must be supplied for every observation in the series (or for the span specified by the **span** argument of the **series** spec, if present). Generally, an adjustment factor must also be supplied for each forecast (and backcast) desired. (See DETAILS.) The adjustment factors are read in free format. If a start date is supplied for the adjustment factors, then they may start before the beginning of the series. If the **data** argument is used, the **file** argument cannot be used. When `mode = diff`, the values in **data** are subtracted from the series, and they need not be positive.

Two series can be input via the **data** argument when both permanent and temporary prior-adjustment factors are specified in the **type** set - see DETAILS for more information.

file Name of the file containing the user-defined prior-adjustment factors. The filename must be enclosed in quotes. If the file is not in the current directory, the path must also be given. If the **file** argument is used, the **data** argument cannot be used. The value restrictions are the same as for **data**. If the data in the file are not in free format, the **format** argument must be used.

If both permanent and temporary prior-adjustment factors are specified in the **type** set, the factors can be input from a single file or from two files - see DETAILS for more information.

format Denotes the format used to read the prior adjustment factors from a file. Five types of input are accepted:

(a) a valid FORTRAN format, which must be enclosed in quotes and must include the initial and terminal parentheses (example: `format="(6f12.0)"`);
(b) a two character code which corresponds to a choice from a set of data formats used in previous versions of X-11 and X-11-ARIMA (example: `format="1r"`);
(c) "datevalue" format, in which the year, month or quarter, and value of each observation are given in this order in free format on individual lines in the data file (example: `format="datevalue"`);
(d) the format `X-12-ARIMA` uses to save a table. This allows the user to read in a file saved from a previous `X-12-ARIMA` run (example: `format="x12save"`).
(e) the format that the `TRAMO` and `SEATS` programs use to read in a series and its descriptors. This enables `X-12-ARIMA` to read in a data file formatted for the `TRAMO` modelling program or the `SEATS` seasonal adjustment program. (example: `format="tramo"`).

In the predefined X-11 data formats mentioned in (b), the data is stored in 6 or 12 character fields, with a year and series label associated with each year of data. For a complete list of these formats, see the DETAILS section of the **series** spec. If no format argument is given, the data will be read in free format. In *free format*, all numbers on a line will be read before reading the next line, and the numbers must be separated by one or more spaces (not by commas or tabs). **Format** can only be used with the **file** argument, not with **data**.

If permanent and temporary prior-adjustment factors are input from two different files with distinct formats, then up to two formats can be specified - see DETAILS for more information.

function Transform the series Y_t input in the **series** spec using a log, square root, inverse, or logistic transformation. Alternatively, perform an AIC-based selection to decide between a log transformation and no transformation (`function=auto`) using either the regARIMA model specified in the **regression** and **arima** specs or the first model in the automatic model file specified in the **automdl** spec (see DETAILS). The default is no transformation (`function = none`). Do not include both the **function** and **power** arguments. **Note:** there are restrictions on the values used in these arguments when preadjustment factors for seasonal adjustment are generated from a regARIMA model; see DETAILS.

Table 6-23: Transformations Available Using the **function** *Argument*

value	*transformation*	*range for Y_t*	*equivalent* **power** *argument*
`none`	Y_t	*all values*	`power = 1`
`log`	$\log(Y_t)$	$Y_t > 0$ for all t	`power = 0`
`sqrt`	$\frac{1}{4} + 2(\sqrt{Y_t} - 1)$	$Y_t \geq 0$ for all t	`power = 0.5`
`inverse`	$2 - \frac{1}{Y_t}$	$Y_t \neq 0$ for all t	`power = -1`
`logistic`	$\log(\frac{Y_t}{1-Y_t})$	$0 < Y_t < 1$ for all t	*none equivalent*

mode Specifies the way in which the user-defined prior adjustment factors will be applied to the time series. If prior adjustment factors to be divided into the series are not given as percents (e.g., (`100 100 50` $\cdots$)), but rather as ratios (e.g., (`1.0 1.0 .5` $\cdots$)), set `mode=ratio`. If the prior adjustments are to be subtracted from the original series, set `mode=diff`. If `mode=diff` is used when the mode of the seasonal adjustment is set to be multiplicative or log additive in the **x11** spec, the factors are assumed to be on the log scale. The factors will be exponentiated to put them on the same basis as the original series. If this argument is not specified, then the prior adjustment factors are assumed to be percents (`mode=percent`).

If both permanent and temporary prior-adjustment factors are specified in the **type** argument, then up to two values can be specified for this argument, provided they are compatible (e.g., `diff` cannot be specified along with ratio or percent). See DETAILS for more information.

name The name of the prior adjustment factors. The name must be enclosed in quotes and may contain up to 64 characters. Up to the first 16 characters will be printed as a label for the prior adjustment factors. When specified with the X-11 formats of the **format** argument, the first six (or eight, if `format="cs"`) characters of this name are also used with the predefined formats to check that the program is reading the correct series, or to find a particular series in a file where many series of factors are stored.

If both permanent and temporary prior-adjustment factors are specified in the **type** set, then the user can either specify series names for both sets of prior adjustment factors, or no name should be entered - see DETAILS for more information.

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

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

This formula for the Box-Cox power transformation is constructed so that its values will be close to Y_t when λ is near 1 and close to $log Y_t$ when λ is near zero. It also has the property that the transformed value is positive when Y_t is greater than 1.

The power λ must be given (e.g., `power = .33`). The default is no transformation ($\lambda = 1$), i.e., `power = 1`. The log transformation (`power = 0`), square root transformation (`power = .5`), and the inverse transformation (`power = -1`) can alternatively be given using the **function** argument. Do not use both the **power** and the **function** arguments in the same spec file. **Note:** there are restrictions on the values used in these arguments when preadjustment factors for seasonal adjustment are generated from a regARIMA model; see DETAILS.

precision The number of decimal digits to be read from the file of prior adjustment factors. This option can only be used with the predefined formats of the **format** argument. This value must be an integer between 0 and 5, inclusive (for example, `precision=5`). The default is zero. If **precision** is used in a **transform** spec that does not use one of the predefined formats, the argument is ignored.

If both permanent and temporary prior-adjustment factors are specified in the **type** argument, then up to two values can be specified for this argument - see DETAILS for more information.

Table 6-24: Available Output Tables for Transform

name	*default*	*brief*	*ext*	*description of table*
priorfactors	+	+	a2	prior-adjustment factors, with associated dates
prioradjusted	·	·	a3	prior-adjusted series, with associated dates
transformed	·	·	trn	prior-adjusted and transformed data, with associated dates

Name gives the name of each table for use with the **print** and **save** arguments.
Default indicates which tables are printed (+) or not printed (·) by default.
Brief indicates which tables are printed (+) or not printed (·) when the **brief** print level is specified. See section 5.1 for more information on print levels.
Ext gives the file extension used if the table is saved. A dot indicates that the table cannot be saved. The file extensions given can also be used as short names for the tables in the **print** and **save** arguments.

print and **save** The following optional output tables are available. The tables available for output are listed on Table 6-24.

savelog Setting `savelog=autotransform` or `savelog=atr` causes the result of the automatic transformation selection procedure to be output to the log file (see section 2.6 for more information on the log file).

start The start date of the user-defined prior-adjustment factors. The default is the start date of the series. Valid values are any date up to the start date of the series (or up to the start date of the span specified by the **span** argument of the **series** spec, if present).

If both permanent and temporary prior-adjustment factors are specified in the **type** set, then up to two starting dates can be specified to read in the two sets of prior adjustment factors - see DETAILS for more information.

title A title for the set of user-defined prior-adjustment factors. The title must be enclosed in quotes and may contain up to 79 characters.

type Specifies whether the user-defined prior-adjustment factors are permanent factors (removed from the final seasonally adjusted series as well as the original series) or temporary factors (removed from the original series for the purposes of generating seasonal factors but not from the final seasonally adjusted series). If only one value is given for this argument (`type = temporary`), then only one set of user-defined prior-adjustment factors will be expected. If both types of user-defined prior-adjustment factors are given (`type = (temporary permanent)`), then two sets of prior adjustment factors will be expected, for more information see DETAILS. The default is `type = permanent`.

RARELY USED ARGUMENTS

aicdiff Defines the difference in AICC needed to accept no transformation when the automatic transformation selection option is invoked (`function=auto`). The default value is `aicdiff = -2.0`. For more information on how this option is used to select a transformation see DETAILS.

trimzero If `trimzero=no`, zeroes at the beginning or end of a time series entered via the file argument are treated as series values. The default (`trimzero=yes`) causes leading and trailing zeros to be ignored. Note that when the **format** argument is set to either `datevalue`, `x12save`, or `tramo`, all values input are treated as series values, regardless of the value of **trimzero**.

DETAILS

If a Box-Cox or logistic transformation is specified in conjunction with a length-of-month (or leap year) adjustment and/or user-defined prior-adjustment factors, the time series is first adjusted for length-of-month and/or prior factors, and then Box-Cox or logistically transformed. If both length-of-month and prior-adjustment factors are specified, then combined adjustment factors (length-of-month $\times$ prior adjustment) are used. Length-of-quarter and leap year adjustments are handled in the same way.

If either `lom` and `loq` of the **adjust** argument is specified, the correct adjustment factor is determined by the **period** specified in the **series** spec. In the case of a monthly input series Y_t, each observation is first divided by the number of days in that month (m_t) and then multiplied by the average length-of-month (30.4375), resulting in $(30.4375 \times Y_t)/m_t$. Length-of-quarter adjustments are performed in a similar manner, resulting in $(91.3125 \times Y_t)/q_t$, where q_t is the length in days of quarter t. Forecasts of the transformed and length-of-month adjusted data are transformed back to the original scale for output (see the documentation of the **forecast** spec).

`X-12-ARIMA` accepts a maximum of 600 user-defined prior-adjustment factors of each type (temporary or permanent), not including the forecast period. (This limit can be changed—see Section 2.8.)

If adjustment factors are supplied for the forecast period, then forecasts of the prior-adjusted series will be inverse-transformed (multiplied or, if `mode = diff`, added to) with these factors. If adjustment factors are not supplied for the forecast period, then inverse-transformation of forecasts will only account for a Box-Cox or logistic transformation and for any length-of-month (or length-of-quarter) adjustment—this effectively assumes values of 1 for the user-defined prior-adjustment factors throughout the forecast period (or 0 if `mode = diff`).

When seasonal adjustment is requested with the **x11** spec, any value of **power** or **function** can be used for the purpose of forecasting the series with a regARIMA model. However, this is not the case when factors generated from the regression coefficients are used to adjust either the original series or the final seasonally adjusted series. In this case, the only accepted transformations are the log transformation (for multiplicative or log-additive seasonal adjustments) and no transformation, which can be specified as power = 1 (for additive seasonal adjustments).

This restriction of the transformation done because factors derived from the regression coefficients must be the same type as factors generated by the seasonal adjustment procedure, so that combined adjustment factors can be derived and adjustment diagnostics can be generated. If the regARIMA model is applied to a log-transformed series, the regression factors are expressed in the form of ratios, which is the same form as the seasonal factors generated by the multiplicative (or log-additive) adjustment modes. Conversely, if the regARIMA model is fit to the original series, the regression factors are measured on the same scale as the original series, which matches the scale of the seasonal factors generated by the additive adjustment mode.

If no seasonal adjustment is done, any power transformation can be used.

When `function=auto` and the series being processed has all positive values, the program will choose between no transformation and a log transformation by fitting a regARIMA model to the untransformed and transformed series and choosing the log transformation except when

$$AICC_{nolog} - AICC_{log} < \Delta_{AICC} \ or \ AICC_{log} + \Delta_{AICC} > AICC_{nolog}$$

where $AICC_{log}$ is the value of AICC from fitting the regARIMA model to the transformed series, $AICC_{nolog}$ is the value of AICC from fitting the regARIMA model to the untransformed series, and Δ_{AICC} is the value entered for the **aicdiff** argument, with a default of -2. Negative values of Δ_{AICC} bias the selection in favor of the log transformation. No transformation is used if the series has a zero or negative value.

If a regARIMA model has been specified in the **regression** and/or **arima** specs, then the procedure will use this model to generate the AICC statistics needed for the test. If no model is specified, or the automatic model identification procedure is specified via the **automdl** spec, the program will read the first model stored in the automatic model file and use this model to generate the AICC statistics. If the automatic model file is not specified in the **file** argument of the **automdl** spec, the program uses a file named `x12a.mdl` in the current directory. For more details on the syntax and structure of this file, see the documentation for the **automdl** spec. If seasonal adjustment is specified via the **x11** or **x11regression** spec, the program will set the seasonal adjustment mode to one that is appropriate for the transformation selected (multiplicative for a log transformation, additive for no transformation).

The program currently does not allow the use of user-defined prior-adjustment factors with the automatic transformation selection option.

Users specifying both temporary and permanent user-defined prior-adjustment factors must take advantage of some special features built into the **transform** spec. For the arguments related to data input, the user can specify an entry for each type of prior-adjustment factor. The **type** variable tells the program which type of prior factor is being referred to by a given entry. For example, in the input specified below

```
transform{
        type=(temporary permanent)
        file=("temp.fil" "perm.fil")
        format=("(6F12.5)" "(F15.3)")
        start=(1980.jan 1975.jul)
        mode=(ratio  percent)
}
```

the series of temporary prior-adjustment factors is read from `temp.fil` using a `(6F12.5)` format. These factors start in January of 1980. The series of permanent prior-adjustment factors, which starts in July of 1975, is read from `perm.fil` using a `(F15.3)` format.

If two entries are given for the **file** arguments but only one entry for each of the **format**, **start**, **mode** and **precision** arguments, then the values given are assumed to apply to both sets of factors. The number of values given for the **name** argument must match the number of prior-adjustments factors implied by the **type** argument.

When the **data** argument is used to input two sets of prior-adjustment factors, the data is assumed to be a matrix of two columns. The type assignment for the data columns is implied by the **type** argument. In the example below

```
transform{
        type=(temporary permanent)
        data=(  1.055   1.000
                  .       .
                  .       .
                  .       .
                1.033   1.000  )
        start=1980.jan
        mode=ratio
}
```

the first column of data is interpreted to be a temporary prior-adjustment factor (with values of 1.055, 0.990, and 1.025), and the second column of data is interpreted to be a permanent prior-adjustment factor. The same assumption is made when only one data file is given for two adjustment types, as in the input below

```
transform{
        type=(temporary permanent)
        file="both.fil"
        start=1980.jan
        mode=ratio
}
```

EXAMPLES

Note: The following examples do not show "complete" spec files. Useful output is not produced unless additional specs (e.g., **x11**, **identify**, or **arima** and **estimate**) are added.

Example 1 Specify a user-defined prior adjustment for a strike in March and April of 1967, as well as a length-of-month adjustment.

```
series { data = (879 899 462 670 985 973 ...)
        start = 1967.jan }
transform { data  = (1  1 .5 .75   1   1 ...)
           mode = ratio
           adjust = lom }
```

Example 2 Specify a logarithmic transformation and also a user-defined adjustment by a price deflator that changes current dollars to constant (real) dollars. A start date is specified for the deflator series since it begins before the time series being modelled.

```
series {title = "Total U.S. Retail Sales --- Current Dollars"
        file = "retail.dat"
        start = 1980.jan}
transform {function = log
           title = "Consumer Price Index"
           start = 1970.jan  # adj. factors start January, 1970
           file = "cpi.dat"
           format  =  "(12f6.3)"}
```

Example 3 Same as Example 2, only a pre-defined format is used to read in the user-defined adjustment factors, and the factors are applied as temporary prior-adjustment factors.

```
series {title = "Total U.S. Retail Sales --- Current Dollars"
        file = "retail.dat"
        start = 1980.jan}
transform {function = log
           title = "Consumer Price Index"
           start = 1970.jan  # adj. factors start January, 1970
           file = "cpi.dat"
           format  =  "1R"
           precision = 3
           name = "cpi"
           type = temporary
          }
```

Example 4 Specify a cube root transformation to stabilize the variation of a quarterly time series.

```
SERIES {TITLE="Annual Rainfall"
        FILE="RAIN.DAT"
        PERIOD=4
        START=1901.1}
TRANSFORM {POWER=.3333}
```

Example 5 This example uses two sets of user-defined prior-adjustment factors: one for the Consumer Price Index that will be removed from the final seasonlly adjusted series to convert the value of the series to current dollars (a permanent prior effect), and a set of strike effects (a temporary prior effect). Each set of factors is read from its own file. Since the files have the same format, single values are entered for **format** and **precision**.

```
series {title = "Retail Sales of computers --- Current Dollars"
        file = "rscomp.dat"
        start = 1980.jan
       }
transform {function = log
           title = "Consumer Price Index & Strike Effect"
           type = (permanent temporary)
           start = 1970.jan  # adj. factors start January, 1970
           file = ("cpi.dat" "strike.dat")
           format  = "1R"
           precision = 3
           name = ("cpi" "strike")
          }
```

Example 6 Use the automatic transformation selection procedure to determine if a log transformation should be used to transform the series. Since a **model** spec is not given, the program will read the first model out of the **x12a.mdl** file used for the automatic model selection procedure to generate the AICC values needed for the test. The AICC difference for the test has been reset to zero, so the program will choose the transformation based on which model estimation yields the smaller value of AICC.

```
series {title = "Total U.K. Retail Sales"
        file = "ukretail.dat"
        start = 1978.jan
       }
transform {function = auto
           aicdiff = 0.0
          }
```

X11

DESCRIPTION

An optional spec for invoking seasonal adjustment by an enhanced version of the methodology of the Census Bureau X-11 and X-11Q programs. The user can control the type of seasonal adjustment decomposition calculated (**mode**), the seasonal and trend moving averages used (**seasonalma** and **trendma**), and the type of extreme value adjustment performed during seasonal adjustment (**sigmalim**). The user can also choose options to specify Easter adjustments (**x11easter**) calculated using a methodology developed by Bateman and Mayes for X-11 in the 1970s. The output options, specified by **print** and **save**, include final tables and diagnostics for the X-11 seasonal adjustment method. In `X-12-ARIMA`, additional specs can be used to diagnose data and adjustment problems, to develop compensating prior regression adjustments, and to extend the series by forecasts and backcasts. Such operations can result in a modified series from which the X-11 procedures obtain better seasonal adjustment factors. For more details on the X-11 seasonal adjustment diagonstics, see Shiskin, Young, Musgrave (1967) and Lothian and Morry (1978). Trading day effect adjustments and other holiday adjustments can be obtained from the **x11regression** spec.

USAGE

```
x11 {mode = pseudoadd
     seasonalma = s3x9
     trendma = 13
     sigmalim=(1.25   2.75)
     title="3x9 moving average, mad"
     appendfcst=yes
     x11easter = yes
     force = totals
     final = user
     print = ( brief +b2 )
     save = ( d10 d11 )
     savelog = ( m7 q )
    }
```

ARGUMENTS

appendfcst Determines if forecasts will be included in certain X-11 tables selected for storage with the save option. If `appendfcst=yes`, then forecasted values will be stored with tables a16, b1, d10, d16, and h1 of the x11 spec, tables a6, a7, a8, a8.tc, a9, and a10 of the regression spec, and tables c16 and c18 of the x11regression spec. If `appendfcst=no`, no forecasts will be stored. The default is to not include forecasts.

final List of the types of prior adjustment factors, obtained from the **regression** and **outlier** specs, that are to be removed from the final seasonally adjusted series. Additive outliers (`final=ao`), level change and ramp outliers (`final=ls`), temporary change (`final=tc`), and factors derived from user-defined regressors (`final=user`) can be removed. If this option is not specified, the final seasonally adjusted series will contain these effects.

force Specifies that the seasonally adjusted series be modified to (a) force the yearly totals of the seasonally adjusted series and the original series to be the same (`force=totals`), (b) adjust the seasonally adjusted values for each calendar year so that the sum of the rounded seasonally adjusted series for any year will equal the rounded annual total (`force=round`), or (c) first force the yearly totals, then round the adjusted series (`force=both`). By default, the program will not adjust the seasonally adjusted values to force the yearly totals to agree with either the yearly totals of the original series or the rounded total of the seasonally adjusted series.

When `force=totals`, the differences between the annual totals is distributed over the seasonally adjusted values in a way that approximately preserves the month-to-month (or quarter-to-quarter) movements of the original series. For more details see Huot (1975) and Cholette (1978). This forcing procedure is not recommended if the seasonal pattern is changing or if trading day adjustment is performed; see DETAILS.

mode Determines the mode of the seasonal adjustment decomposition to be performed. There are four choices: multiplicative (`mode=mult`), additive (`mode=add`), pseudo-additive (`mode=pseudoadd`), and log-additive (`mode=logadd`) decomposition. The default mode is `mult`, unless the automatic transformation selection procedure is invoked in the **transform** spec; in the latter case, the mode will match the transformation selected for the series (**mult** for the log transformation and **add** for no transformation).

print and **save** Table 6-25 gives the output tables that are available by default; Table 6-26 gives other tables that can be printed or saved using this argument, while Table 6-27 shows the line printer plots that can be specified using the **print** argument.

Table 6-25: Default Output Tables for X11

name	*brief*	*ext*	*description of table*
x11easter	+	h1	X-11 Easter adjustment factors
combholiday	+	chl	combined holiday prior adjustment factors, A16 table
adjoriginal	+	b1	original series, adjusted for prior effects and forecast extended
irrwt	·	c17	final weights for the irregular component
unmodsi	·	d8	final unmodified si-ratios (differences)
ftestd8	+	·	F-tests for stable and moving seasonality, D8
replacsi	·	d9	final replacement values for extreme si-ratios (differences), D iteration
movseasrat	·	·	moving seasonality ratios for each period
seasonal	+	d10	final seasonal factors
seasonaldiff	+	fsd	final seasonal difference (only for pseudo-additive seasonal adjustment)
seasadj	+	d11	final seasonally adjusted series
seasadjtot	+	saa	final seasonally adjusted series with constrained yearly totals (if **force = round** or **force = both**)
saround	+	rnd	rounded final seasonally adjusted series (if **force = round**) or the rounded final seasonally adjusted series with constrained yearly totals (if **force = both**)
residualseasf	·	·	F-test for residual seasonality
trend	·	d12	final trend-cycle
irregular	·	d13	final irregular component
adjustfac	+	d16	combined seasonal and trading day factors
adjustdiff	+	fad	final adjustment difference (only for pseudo-additive seasonal adjustment)
calendar	+	d18	combined holiday and trading day factors
yrtotals	·	e4	ratio of yearly totals of original and seasonally adjusted series
origchanges	·	e5	percent changes (differences) in original series
sachanges	·	e6	percent changes (differences) in seasonally adjusted series
revsachanges	·	e6a	percent changes (differences) in seasonally adjusted series with revised yearly totals
rndsachanges	·	e6r	percent changes (differences) in rounded seasonally adjusted series
trendchanges	·	e7	percent changes (differences) in final trend component series
x11diag	+	·	summary of seasonal adjustment diagnostics
qstat	+	·	quality control statistics
tdaytype	+	·	trading day factors printed by type of month
specsa	+	sp1	spectral plot of differenced, seasonally adjusted series (or of the logged seasonally adjusted series if **mode = logadd** or **mode = mult**)
specirr	+	sp2	spectral plot of outlier-modified irregular series

Name gives the name of each table for use with the **print** and **save** arguments.

Brief indicates which tables are printed (+) or not printed (·) when the **brief** print level is specified. See section 5.1 for more information on print levels.

Ext gives the file extension used if the table is saved. A dot indicates that the table cannot be saved. The file extensions given can also be used as short names for the tables in the **print** and **save** arguments.

Table 6-26: Other Output Tables for X11

name	*ext*	*description of table*
ftestb1	·	F-test for stable seasonality, B1 table
trendb2	b2	preliminary trend-cycle, B iteration
sib3	b3	preliminary unmodified si-ratios (differences)
replacsib4	·	preliminary replacement values for extreme si-ratios (differences), B iteration
seasonalb5	b5	preliminary seasonal factors, B iteration
seasadjb6	b6	preliminary seasonally adjusted series, B iteration
trendb7	b7	preliminary trend-cycle, B iteration
sib8	b8	unmodified si-ratios (differences)
replacsib9	·	replacement values for extreme si-ratios (differences), B iteration
seasonalb10	b10	seasonal factors, B iteration
seasadjb11	b11	seasonally adjusted series, B iteration
irregularb	b13	irregular component, B iteration
irrwtb	b17	preliminary weights for the irregular component
tdadjorigb	b19	original series adjusted for preliminary trading day
extremeb	b20	extreme values, B iteration
adjoriginalc	c1	original series modified for outliers, trading day and prior factors, C iteration
trendc2	c2	preliminary trend-cycle, C iteration
modsic4	c4	modified si-ratios (differences), C iteration
seasonalc5	c5	preliminary seasonal factors, C iteration
seasadjc6	c6	preliminary seasonally adjusted series, C iteration
trendc7	c7	preliminary trend-cycle, C iteration
replacsic9	c9	modified si-ratios (differences), C iteration
seasonalc10	c10	preliminary seasonal factors, C iteration
seasadjc11	c11	seasonally adjusted series, C iteration
irregularc	c13	irregular component, C iteration
tdadjorig	c19	original series adjusted for final trading day
extreme	c20	extreme values, C iteration
adjoriginald	d1	original series modified for outliers, trading day and prior factors, D iteration
trendd2	d2	preliminary trend-cycle, D iteration
modsid4	d4	modified si-ratios (differences), C iteration
seasonald5	d5	preliminary seasonal factors, D iteration
seasadjd6	d6	preliminary seasonally adjusted series, D iteration
trendd7	d7	preliminary trend-cycle, D iteration
unmodsiox	d8b	final unmodified SI ratios, with labels for outliers and extreme values
autosf	·	automatic seasonal factor selection
trendadjls	tal	final trend-cycle adjusted for level shift outliers
irregularadjao	iao	final irregular component adjusted for point outliers
modoriginal	e1	original series modified for zero-weighted extreme values
modseasadj	e2	seasonally adjusted series modified for zero-weighted extreme values
modirregular	e3	irregular component modified for zero-weighted extreme values
robustsa	e11	robust final seasonally adjusted series
mcdmovavg	f1	MCD moving average of the final seasonally adjusted series

Name gives the name of each table for use with the **print** and **save** arguments.
Ext gives the file extension used if the table is saved. A dot indicates that the table cannot be saved. The file extensions given can also be used as short names for the tables in the **print** and **save** arguments.

Table 6-27: Plots Specified by the print Argument

name	*description of plot*
adjorigplot	plot of the prior adjusted original series augmented by prior-adjusted forecasts (if specified); if no prior factors or forecasts are used, the original series is plotted
seasonalplot	seasonal factor plots, grouped by month or quarter
seasadjplot	plot of the final seasonally adjusted series
trendplot	plot of the final trend-cycle
irregularplot	plot of the final irregular component
origwsaplot	plot of the original series with the final seasonally adjusted series
ratioplotorig	month-to-month (or quarter-to-quarter) ratio plots of the original series
ratioplotsa	month-to-month (or quarter-to-quarter) ratio plots of the seasonally adjusted series

Name gives the name of each plot for use with the **print** arguments.

Table 6-28: Available Log File Diagnostics for X11

name	*short*	*description of diagnostic*
m1	m1	M1 Quality Control Statistic
m2	m2	M2 Quality Control Statistic
m3	m3	M3 Quality Control Statistic
m4	m4	M4 Quality Control Statistic
m5	m5	M5 Quality Control Statistic
m6	m6	M6 Quality Control Statistic
m7	m7	M7 Quality Control Statistic
m8	m8	M8 Quality Control Statistic
m9	m9	M9 Quality Control Statistic
m10	m10	M10 Quality Control Statistic
m11	m11	M11 Quality Control Statistic
q	q	Overall index of the acceptablility of the seasonal adjustment
q2	q2	Q statistic computed without the M2 Quality Control statistic
movingseasratio	msr	Moving seasonality ratio
icratio	icr	$\bar{I}/\bar{C}$ ratio
fstableb1	fb1	F-test for stable seasonality, performed on the original series
fstabled8	fd8	F-test for stable seasonality, performed on the final SI-ratios
movingseasf	msf	F-test for moving seasonality
idseasonal	ids	Identifiable seasonality test result

Name gives the name of each diagnostic for use with the **savelog** argument.
Short gives a short names for the diagonstics in the **savelog** arguments.

savelog The diagnostics available for output to the log file (see section 2.6) are listed in Table 6-28.

seasonalma Specifies which seasonal moving average (also called seasonal "filter") will be used to estimate the seasonal factors. These seasonal moving averages are **n×m moving averages**, meaning that an n-term simple average is taken of a sequence of consecutive m-term simple averages.

The seasonal filters shown in Table 6-29 can be selected for the entire series, or for a particular month or quarter. If the same moving average is used for all calendar months or quarters, only a single value need be entered. If different seasonal moving averages are desired for some calendar months or quarters, a list of these must be entered, specifying the desired seasonal moving average for each month or quarter. An example for a quarterly series is the following: `seasonalma=(s3x3 s3x9 s3x9 s3x9)`.

Table 6-29: X-12-ARIMA Seasonal Filter Options and Descriptions

name	*description of option*
s3x1	A 3×1 moving average.
s3x3	A 3×3 moving average.
s3x5	A 3×5 moving average.
s3x9	A 3×9 moving average.
s3x15	A 3×15 moving average.
stable	Stable seasonal filter. A single seasonal factor for each calendar month or quarter is generated by calculating the simple average of all the values for each month or quarter (taken after detrending and outlier adjustment).
x11default	A 3×3 moving average is used to calculate the initial seasonal factors in each iteration, and a 3×5 moving average to calculate the final seasonal factors.

If no seasonal moving average is specified, the program will choose the final seasonal filter automatically; this option can also be invoked by setting `seasonalma=msr`. This is done using the moving seasonality ratio procedure of X-11-ARIMA/88, see DETAILS. This is a change from previous versions of X-11 and X-11-ARIMA where, when no seasonal moving average was specified, a 3×3 moving average was used to calculate the initial seasonal factors in each iteration, and a 3×5 moving average to calculate the final seasonal factors. This seasonal filtering sequence can be specified by entering `seasonalma=x11default`.

sigmalim Specifies the lower and upper sigma limits used to downweight extreme irregular values in the internal seasonal adjustment iterations. The **sigmalim** argument has two input values, the lower and upper sigma limits. Valid list values are any real numbers greater than zero with the lower sigma limit less than the upper sigma limit (example: `sigmalim=(1.8 2.8)`). A missing value defaults to 1.5 for the lower sigma limit and 2.5 for the upper sigma limit. For example, the statement `sigmalim=(,3.0)` specifies that the upper sigma limit will be set to 3.0, while the lower sigma limit will remain at the 1.5 default. A comma is necessary if either sigma limit is missing. For an explanation of how X-12-ARIMA uses these sigma limits to derive adjustments for extreme values, see DETAILS.

title Title of the seasonal adjustment, in quotes, for the convenience of the user. This can be a single title or a list of up to 8 titles; an example with two titles is:

```
title=  ("3x9, trading day adjustment"
         "for sales of sporting goods")
```

If a list is provided, each title must be on a separate line of the spec file. This list will be printed on the title page below the series title. There is no default seasonal adjustment title.

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

type When `type=summary`, the program develops estimates of the trend-cycle, irregular, and related diagnostics, along with residual seasonal factors and, optionally, also residual trading day and holiday factors from an input series which is assumed to be either already seasonally adjusted or nonseasonal. These residual factors are not removed. The output series in the final seasonally adjusted series (table D11) is the same as the original series (table A1). When `type=trend`, the program develops estimates for the final trend-cycle and irregular components without attempting to estimate a seasonal component. The input series is assumed to be either already seasonally adjusted or nonseasonal. With this option, estimated trading day and holiday effects as well as permanent prior adjustment factors are removed to form the adjusted series (table D11) as well as for the calculation of the trend (table D12). When a metafile with a composite spec is used to obtain an indirect adjustment of an aggregate, these options are used for components of the aggregate that are not seasonally adjusted. In the default setting, `type=sa`, the program calculates a seasonal decomposition of the series. With all three values of **type**, the final seasonally adjusted series (printed in the D 11 table of the main output file) is used to form the indirect seasonal adjustment of the composite.

x11easter Specifies whether or not an Easter holiday effect is estimated using a non-parametric procedure of Bateman and Mayes from X-11. If `x11easter=yes`, a preliminary seasonal adjustment is performed, and the March and April irregulars from this adjustment are divided into four groups. The trimmed means of these groups are used to estimate the Easter holiday effects. The estimates are removed to obtain the series used to produce the seasonal adjustments. This option is only available for multiplicative adjustments of monthly time series. For more details on the X-11 Easter adjustment procedure, see Appendix A of Monsell (1989) or Chen and Findley (1998). The default setting is `x11easter=no`.

RARELY USED ARGUMENTS

calendarsigma Specifies if the standard errors used for extreme value detection and adjustment are computed separately for each calendar month (quarter), or separately for two complementary sets of calendar months (quarters). If `calendarsigma=all`, the standard errors will be computed separately for each month (quarter). If `calendarsigma=signif`, the standard errors will be computed separately for each month only if Cochran's hypothesis test determines that the irregular component is heteroskedastic by calendar month (quarter). If `calendarsigma=select`, the months (quarters) will be divided into two groups, and the standard error of each group will be computed. For the **select** option, the argument **sigmavec** must be used to define one of the two groups of months (quarters). If **calendarsigma** is not specified, the standard errors will be computed from 5 year spans of irregulars, in the manner described in Dagum(1988).

forcestart Used in conjunction with **force**, this option sets the beginning of the yearly benchmark period over which the seasonally adjusted series will be forced to sum to the total. Unless **forcestart** is used, the year is assumed to be the calendar year for the procedure invoked by setting `force=totals`, but an alternate starting period can be specified for the year (such as the start of a fiscal year) by assigning to **forcestart** the month (either the full name of the month or the abbreviations shown in Section 5.2) or quarter (**q1** for the first quarter, **q2** for the second quarter, etc.) of the beginning of the desired yearly benchmarking period. For example, to specify a fiscal year which starts in October and ends in September, set `forcestart=october` or `forcestart=oct`. To specify a fiscal year which starts in the third quarter of one year and ends in the second quarter of the next, set `forcestart=q3`.

itrendma Specifies the moving average used to form the inital estimate of the trend-cycle. If `itrendma = centered1yr` a centered one year moving average will be used. If `itrendma = cholette2yr` a two year moving average, a modified Leser filter developed by Pierre Cholette of Statistics Canada, will be used (for more details, see Leser(1963) and Cholette(1979)). The default is the one year centered moving average; the two year filter is useful when adjusting series with short cyclical fluctuations or series with sudden changes in the trend level.

keepholiday Determines if holiday effects estimated by the program are to be kept in the final seasonally adjusted series. In the default setting, `keepholiday=no`, holiday adjustment factors derived from the program are removed from the final seasonally adjusted series. If `keepholiday=yes`, holiday adjustment factors derived from the program are kept in the final seasonally adjusted series. The default is used to produce a series adjusted for both seasonal and holiday effects.

print1stpass If `print1stpass=yes`, output from the seasonal adjustment needed to generate the irregular components used for the X-11 Easter and irregular regression adjustment procedures will be printed out. If `print1stpass=no`, this output will be suppressed, and only the tables associated with the X-11 Easter and irregular regression procedures are printed out. The default is `print1stpass=no`. When `print1stpass=yes`, the **print** argument controls which tables are actually printed.

sfshort Controls what seasonal filters are used to obtain the seasonal factors if the series is at most 5 years long. For the default case, `sfshort=no`, a stable seasonal filter will be used to calculate the seasonal factors, regardless of what is entered for the **seasonalma** argument. If `sfshort=yes`, `X-12-ARIMA` will use the seasonal filter given in the **seasonalma** argument wherever possible.

sigmavec Specifies one of the two groups of months (quarters) for whose irregulars a group standard error will be calculated under the `calendarsigma=select` option. The user enters the month(s) (either the full name of the month or the abbreviations shown in Section 5.2) or quarter(s) (**q1** for the first quarter, **q2** for the second quarter, etc.) that comprise one group; all remaining months or quarters comprise the second group. Example: `sigmavec=(jan feb dec)`. **Warning:** This argument can only be specified when `calendarsigma=select`.

spectrumaxis If `spectrumaxis=same`, the spectra for the differenced original series and the differenced seasonally adjusted series modified for extremes will be displayed using the same axis. If `spectrumaxis=diff`, the spectral plots will not be so constrained. The default is `spectrumaxis=diff`.

trendic Specifies the irregular-to-trend variance ratio that will be used to generate the end weights for the Henderson moving average. The procedure is taken from Doherty (1991). If this variable is not specified, the value of **trendic** will depend on the length of the Henderson trend filter. These default values closely reproduce the end weights for the set of Henderson trend filters which originally appeared in `X-11` and `X-11-ARIMA`.

true7term Specifies the end weights used for the seven term Henderson filter. If `true7term = yes`, then the asymmetric ends weights for the 7 term Henderson filter are applied for observations at the end of the series where a central Henderson filter cannot be applied. If `true7term = no`, then central and asymmetric weights from a 5 term Henderson filter are applied, as in previous versions of the `X-11-ARIMA` program released by Statistics Canada. The default is `true7term = no`.

DETAILS

Modes of seasonal adjustment: In any X-12-ARIMA seasonal adjustment, the original time series (O) is decomposed into three basic components:

Trend-Cycle (C): The long-term and medium-to-long term movements of the series, including consequential turning points.

Seasonal (S): Within-year fluctuations about the trend that recur in a very similar way in the same month or quarter from year to year.

Irregular (I): The residual component that remains after seasonal and trend are removed from the series (and also trading day and holiday effects once these have been identified). It is characterized by movements of very short duration. These can be quite large if there are strikes or other unusual economic events of short duration.

Depending mainly on the nature of the seasonal movements of a given series, several different models are used to describe the way in which the components C, S, and I combine to form the original series O. X-12-ARIMA provides modes of seasonal adjustment appropriate for four different decoposition models. The table below gives the four values of the **mode** arguments and also the corresponding models, both for the original (O) and seasonally adjusted series (SA).

Table 6-30: Modes of Seasonal Adjustment and Their Models

Entry for mode argument	*Name for mode*	*Model for O*	*Model for SA*
mult	Multiplicative	$O = C \times S \times I$	$SA = C \times I$
add	Additive	$O = C + S + I$	$SA = C + I$
pseudoadd	Pseudo-Additive	$O = C \times [S + I - 1]$	$SA = C \times I$
logadd	Log-Additive	$Log(O) = C + S + I$	$SA = exp(C + I)$

The default seasonal adjustment mode is multiplicative. This is because, for most seasonal economic time series, the magnitudes of the seasonal fluctuations appear to increase and decrease proportionally with increases and decreases in the level of the series, in a way that is proportional to the level. A series with this type of seasonality is said to have **multiplicative seasonality**. To estimate the multiplicative components, the program uses a ratio-to-moving average method whose details are given in Shiskin, Young, and Musgrave (1967), Dagum (1988), and Baxter (1994), among others. The pseudo-additive model is considered when some months (or quarters) have extremely small values (due to vacations or climate, for example), and the remaining months appear to have multiplicative seasonality. If the magnitude of the seasonal does not appear to be affected by the level of the series, then the series has **additive seasonality**, and the additive mode is appropriate.

The log-additive mode gives an alternative multiplicative decomposition which can be useful for certain econometric analysis, usually related to time series model considerations. For log-additive seasonal adjustment, the trend component is computed from an additive decomposition of the logged series ($log(O)$), so the additive trend must be exponentiated in order to derive a trend with the same units as the original series. This results in a downwardly biased estimate of trend; this bias is adjusted in `X-12-ARIMA` using a bias-correction procedure described in Thomson and Ozaki (1992).

For multiplicative, pseudo-additive and log-additive seasonal adjustment, the seasonal and irregular components are assumed to be ratios centered about 1. In the main output they are expressed as percentages so that they center about 100. For additive seasonal adjustment, the seasonal and irregular components are in the same units as the original time series and vary about 0.

When a regARIMA model is specified with the **regression** and **arima** specs, trading day, holiday, outlier, and other regression effects defined in the **regression** spec can be derived from the regression coefficients of a regARIMA model and used to adjust the original series prior to seasonal adjustment. In this case, these effects must be the same type as factors generated by the seasonal adjustment procedure, so that combined adjustment factors can be derived and adjustment diagnostics can be generated. If the regARIMA model is fit to a log-transformed series, the regression factors are expressed in the form of ratios, which is the same form as factors generated by the multiplicative or log-additive adjustment modes. Conversely, if the regARIMA model is fit to the original series, the regression factors are measured on the same scale as the original series, which matches the scale of the components generated by the additive adjustment mode. Therefore, users should be careful to ensure that the transformation specified by the **function** or **power** arguments of the **transform** spec is compatable with the seasonal adjustment mode specified by the **mode** argument of the **x11** spec. Furthermore, be aware that the default value for the **mode** argument is multiplicative seasonal adjustment, which conflicts with the default for the **function** and **power** arguments of the **transform** spec, which assume no transformation is done. Currently, you cannot use regression effects to pre-adjust the original series for a pseudo-additive seasonal adjustment.

Multiplicative and pseudo-additive seasonal adjustment give very similar results for most series with multiplicative seasonality, unless the seasonal amplitude of the series is large. If the smallest seasonal factor is 0.7 or less, there will be noticable differences between the multiplicative and pseudo-additive seasonal adjustments. If the smallest seasonal factor is 0.5 or less, this difference is likely to be important. If a multiplicative seasonal adjustment produces many more extreme values (meaning values of less than 100.0 in Table C17, especially 0) in months (or quarters) with small seasonal factors than months with large seasonal factors, then the pseudo-additive seasonal adjustment is likely to be better. For more details on when to use pseudo-additive seasonal adjustment, see Baxter (1994).

For simplicity, this discussion has ignored trading day and holiday effects. When these are estimated, they add additional factors to the decomposition and, depending on how they are defined, adjustment for them can lead to larger differences between the annual totals of the adjusted series and the annual totals of the original time series.

Downweighting of extreme irregulars: Let μ_I be the assumed mean of the irregular component (1.0 for multiplicative seasonal adjustment, 0.0 for additive). Let σ_{X11} denote an estimate of the standard deviation of the irregular component for a month or quarter. If the absolute value of $I_t - \mu_I$ is less than the lower sigma limit multiplied by σ_{X11}, the irregular value I_t receives full weight. If the absolute value of $I_t - \mu_I$ is more than the upper sigma limit multiplied by σ_{X11}, the irregular value receives zero weight, meaning that I_t is replaced by μ_t for seasonal factor calculations. Otherwise, I_t is partially downweighted.

Automatic seasonal filter selection: This procedure is taken from X-11-ARIMA/88, see Dagum(1988). For the first two seasonal adjustment iterations, a 3×3 moving average is used to calculate the initial seasonal factors and a 3×5 moving average is used to calculate the final seasonal factor. In the third and final iteration, a 3×3 moving average is used to calculate the initial seasonal factors, but for the final iteration the program calculates the moving seasonality ratio ($\bar{I}/\bar{S}$, also called the global MSR). Then the program chooses whether to use a 3×3, 3×5, or 3×9 moving average based on the size of the global MSR. For more information on the moving seasonality ratio, see Lothian (1984).

Forecast extension: As mentioned in the introduction, an important use of regARIMA models is to extend the series with forecasts (and backcasts) to improve the seasonal adjustment of the most recent (and the earliest) observations. Therefore, `X-12-ARIMA` will extend the series with one year of forecasts prior to seasonal adjustment whenever a regARIMA model is specified with no **forecast** spec. To specify a seasonal adjustment without forecast extension, set `maxlead = 0` in the **forecast** spec.

Residual seasonal and trading day effects in the adjusted series: A routine searches each of the spectra for peaks at the seasonal and trading day frequencies. A warning message is printed out if visually significant peaks are found, and the plot in which a peak was found is printed out. When the restricted output (the **-n** flag) option is used, the plot is not produced in the main output, but a message is printed suggesting that the user rerun the program without the **-n** flag.

Level shifts and the final Henderson trend: When level shifts are estimate and removed from the series prior to seasonal adjustment, they are put back into the final Henderson trend cycle (Table D12), so that this component will have the level of the observed data. A table of the trend cycle of the level shift adjusted time series can also be obtained by setting `print = trendadjls`.

Easter adjustment: The Easter adjustment options in this spec cannot be used when regARIMA model based holiday are specified in the **regression** spec, or if an Easter adjusment is specified within the **x11regression** spec.

Annual Totals: Forcing the seasonally adjusted totals to be the same as the original series annual totals can degrade the quality of the seasonal adjustment, especially when the seasonal pattern is undergoing change. It is not natural if trading day adjustment is performed because the aggregate trading day effect over a year is variable and moderately different from zero.

Table of SI values with labels for extreme values: Table D 8.B is designed to provide users with direct information about which of the unmodified Seasonal-Irregular values (the detrended series, henceforth called SI values) produced by the X-11 seasonal adjustment program will be modified by extreme value adjustment (as shown by the irregular weights in Table C 17) or are likely to have been affected by regARIMA outliers (either those specified by the user or those identified though the **outlier** spec).

Each SI value that has been identified as an X-11 extreme value is printed with a "*" next to it. SI values at times at which a single regARIMA outlier occurs in the model are printed with a "#" next to them. Extreme SI values at times associated with at least one regARIMA outlier are printed with a "`&`" next to them; SIs at times with more than one regARIMA outlier will have a "`@`" next to them. All observations between (and including) the starting and ending points of a ramp outlier are marked as if they were outliers.

With multiplicative seasonal adjustments, SI values before and after level shift outliers that are most likely to have been affected by the level shift are marked with a "-" character next to the value. The number of observations flagged in this way depends on the magnitude of the level shift outlier (as determined by its regression coefficient estimate) and on the length of the Henderson filter used for the trend that generates the SI ratios, in the manner described in Table 6-31.

Treatment of nonseasonal series: A nonseasonal series can be decomposed into trend-cycle and irregular components using the `type=trend` option. This decomposition is obtained by a simplification of the X-11 seasonal adjustment decomposition procedure that retains only the steps related to the Henderson trends and extreme value detection.

Table 6-31 : Number of Surrounding SI-ratios in Table D 8.B Assumed Affected by a Level Shift

Percent Change in Level (Δ_L)	Length of Henderson Filter 23	13	9	7	5
$\Delta_L \leq 1.1$	0	0	0	0	0
$1.1 < \Delta_L \leq 1.2$	1	1	0	0	0
$1.2 < \Delta_L \leq 1.3$	1	1	1	0	0
$1.3 < \Delta_L \leq 1.5$	2	1	1	0	0
$1.5 < \Delta_L \leq 1.8$	2	1	1	1	0
$1.8 < \Delta_L \leq 1.9$	2	2	1	1	0
$1.9 < \Delta_L \leq 2.0$	3	2	1	1	0
$2.0 < \Delta_L \leq 2.6$	3	2	1	1	1
$2.6 < \Delta_L \leq 2.9$	3	2	2	1	1
$2.9 < \Delta_L \leq 3.6$	4	2	2	1	1
$3.6 < \Delta_L \leq 5.5$	4	3	2	1	1
$5.5 < \Delta_L$	5	3	2	1	1

EXAMPLES

Example 1 Multiplicative seasonal adjustment with all default options (so the program uses the moving seasonality ratio to select the seasonal filter length). The monthly series starts in January, 1976 and is stored in free format in the file `klaatu.dat` in the current directory.

```
Series { File="klaatu.dat" Start = 1976.1 }
X11 {  }
```

Example 2 Multiplicative monthly seasonal adjustment, 3×9 seasonal factors for all months, 23-term Henderson moving average for the trend-cycle. Perform a test (using a version of AIC that adjusts for the length of the series) of the significance of the trading-day regressors in a regression of the irregular component.

```
Series { File="klaatu.dat" Start = 1976.1 }
X11 { SeasonalMA =  s3x9  TrendMA = 23  }
X11regression { variables = td   aictest=td  }
```

Example 3. Quarterly seasonal adjustment, 3×3 seasonal factors for first two quarters, 3×5 seasonal factors for remaining two quarters, 7-term Henderson trend moving average.

```
series {
          file="qhstarts.dat"
          start = 1967.1
          period=4
       }
x11    {
          seasonalma = (s3x3 s3x3 s3x5 s3x5)
          trendma = 7
       }
```

Example 4. A multiplicative monthly seasonal adjustment is to be performed with 3×9 seasonal moving averages for all months using ARIMA forecast extension of length 12 months, if one of the default model types is accepted by the automatic modeling procedure. The fiscal yearly totals for the seasonally adjusted series will be forced to equal the totals of the original series for a fiscal year starting in October.

```
SERIES { TITLE="EXPORTS OF TRUCK PARTS" START =1967.1
         FILE = "X21109.ORI"  }
AUTOMDL{ MODE=FCST }
X11 { SEASONALMA = S3X9    FORCE=TOTALS   FORCESTART = OCT  }
```

Example 5. Seasonal ARIMA model with regression variables used to obtain preadjustments of monthly data. No forecast extension will be done in this example. Specified regression variables are a constant, trading day effects, and two level shifts, one in May, 1972 and one in September, 1976. The ARIMA part of the model is $(0,1,2)(1,1,0)_{12}$. Additively seasonally adjust the series after preadjusting for the outlier, level-shift and trading day effects estimated using the regARIMA model. Use sigma limits set to 2.0 and 3.5 to search for outliers in the irregular component of the seasonal decomposition. Use the **alltables** print level when printing out seasonal adjustment output.

```
SERIES{ TITLE = "EXPORTS OF LEATHER GOODS"  START = 1969.JUL
   DATA = (815 866 926 ... 942) }
REGRESSION{ VARIABLES = (CONST TD LS1972.MAY LS1976.OCT) }
ARIMA{  MODEL=(0 1 2)(1 1 0)  }
ESTIMATE{ }
FORECAST{  MAXLEAD=0  }
X11{   MODE = ADD  PRINT = ALLTABLES  SIGMALIM = (2.0 3.5) }
```

Example 6. The predefined regression effects used are trading day variables and a constant. User-defined regression variables are included to capture the effect of special sales promotions in 1988 and 1990. These variables are read in from the file `special.dat`. The ARIMA part of the model is $(3,1,0)(0,1,1)_{12}$. The seasonal period, 12, is not specified since this is the default. Perform a multiplicative seasonal adjustment on the series after pre-adjusting for the regARIMA trading day and user-defined regression effects and extending the series with 12 forecasts and 12 backcasts. A two-line list of seasonal adjustment titles is specified.

```
series { title = "Unit Auto Sales"  file = "autosal.dat"
         start = 1985.1 }
transform { function = log }
regression { variables = (const td)   user = (sale88 sale90)
             file = "special.dat"     format = "(2f12.2)" }
arima { model = (3 1 0)(0 1 1)12 }
forecast { maxlead=12    maxback=12 }
x11 {  title = ("Unit Auto Sales"
                "Adjusted for special sales in 1988, 1990")
       }
```

Example 7. Read in the data from a file using a predefined X-11 data format. Note that the starting date will be taken from the information provided in the data file and so does not have to be specified. Specify a regARIMA model for the log transformed data with certain outlier terms. Use this model to generate 5 years of forecasts. Perform a multiplicative seasonal adjustment using a 3×9 seasonal moving average for all months.

```
series { title="NORTHEAST ONE FAMILY Housing Starts"
         file="cne1hs.ori"   name="CNE1HS"    format="2R" }
transform {    function=log   }
regression {
    variables=(ao1976.feb ao1978.feb ls1980.feb
               ls1982.nov ao1984.feb)
}
arima {  model=(0 1 2)(0 1 1)  }
forecast {  maxlead=60  }
x11 {  seasonalma=(s3x9)
       title="Adjustment of 1 family housing starts"
}
```

Example 8. The predefined regression effect is a constant. The user-defined regression variables are for strikes in 1980, 1985, and 1991 and are located in the file `strikes.dat`. The ARIMA part of the model is (0,1,1)(0,1,1)12. Since a model is specified in the spec, generate a year of forecasts by default. Seasonally adjust the series after pre-adjusting for the user-defined regression effects. Adjust the series for X-11 trading day and for Easter effects estimated by the Bateman-Mayes procedure before estimating the regARIMA model.

```
series{ title="Automobile Sales"
        file = "carsales.dat"
        start = 1975.1 }
transform{ function = log }
regression{ variables = ( const )
            user = (strike80 strike85 strike90)
            file = "strike.dat"  format = "(3f12.0)"
            }
arima{ model = (0 1 1)(0 1 1)12 }
x11{  title = ("Car Sales in US - Adjust for strikes in 80, 85, 90")
      save=seasonal appendfcst=yes
      x11easter=yes
   }
x11regression { variables = td }
```

X11REGRESSION

DESCRIPTION

An optional spec for use in conjunction with the **x11** spec for series without missing observations. This spec estimates calendar effects by regression modeling of the irregular component with predefined or user-defined regressors. The user can select predefined regression variables with the **variables** argument. The predefined variables are for calendar (trading-day and holiday) variation and additive outliers. A change-of-regime option is available with trading-day regressors. User-defined calendar effect regression variables can be included in the model via the **user** argument. Data for any user-defined variables must be supplied, either in the **data** argument, or in a file named in the **file** argument (not both). The regression model specified can contain both predefined and user-defined regression variables.

USAGE

```
x11regression {variables = (td  or  td1coef  or  tdstock[31]
                            easter[8]  labor[8]
                            thank[1]
                            ao1967.apr )
               user = (temperature precip)
               start = 1955.jan
                   data = (25   0.1   ...)
               or
                   file = "weather.dat"
                   format = "(2f5.1)"
               tdprior = (  1.4   1.4   1.4   1.4   1.4
                            0.0   0.0 )
               aictest = (  easter  user
                            td  or  td1coef  or  tdstock )
               span = (1980.jan,1995.dec)
                   sigma = 2.75
               or
                   critical = 3.5
                   outliermethod = addone
                   outlierspan = (1973.may, 1992.sep)
               usertype = holiday
               print = ( brief +b15 )
               save = ( c16 c18 )
               savelog = aictest
              }
```

ARGUMENTS

aictest Specifies that an AIC-based comparison will be used to determine if a specified regression variable should be included in the user's regARIMA The only entries allowed for this variable are **td**, **tdstock**, **td1coef**, **easter**, and **user**. If a trading day model selection is specified, for example, then AIC values (with a correction for the length of the series, henceforth referred to as AICC) are derived for models with and without the specified trading day variable. By default, the model with smaller AICC is used to generate forecasts, identify outliers, etc. If more than one type of regressor is specified, the AIC-tests are performed sequentially in this order: (a) trading day regressors, (b) easter regressors, (c) user-defined regressors. If there are several variables of the same type (for example, several td regressors), then the **aictest** procedure is applied to them as a group. That is, either all variables of this type will be included in the final model or none. See DETAILS for more information on the testing procedure. If this option is not specified, no automatic AIC-based selection will be performed.

critical Sets the critical value (threshold) against which the absolute values of the outlier t-statistics are compared to detect additive outliers (meaning extreme irregular values). This argument applies unless the **sigma** argument is used, or the only regressor(s) estimated is flow trading day. The assigned value must be a real number greater than 0. Example: `critical=4.0`. The default critical value is determined by the number of observations in the interval searched for outliers (see the **outlierspan** argument below). Table 6-15 gives default critical values for a number of outlier span lengths. Larger (smaller) critical values predispose **x11regression** to treat fewer (more) irregulars as outliers. A large value of **critical** should be used if no protection is wanted against extreme irregular values.

data Assigns values to the user-defined regression variables. The time frame of the values must cover the time frame of the series (or of the span specified by the **span** argument of the **series** spec, if present). It must also cover the time frame of forecasts and backcasts requested in the **forecast** spec. The data values are read in free format. The numerical values given in this argument should be in the order in which the user-defined variables are named in the **user** argument. This assignment should proceed through all the values of the user-defined variables for the first time point, then through all the values for the second time point, etc. If the **data** argument is used, the **file** argument cannot be used.

file Name of the file containing data values for *all* user-defined regression variables. The filename must be enclosed in quotes. If the file is not in the current directory, the path must also be given. As with the **data** argument, the time frame of the data values must cover both the series and any forecasts and backcasts. If the **file** argument is used, the **data** argument cannot be used.

format Indicates the format used when reading the values for the regression variables in the file named in the **file** argument. Three types of input are accepted:

(a) a valid FORTRAN format, which must be enclosed in quotes and must include the initial and terminal parentheses (example: `format="(6f12.0)"`);
(b) "datevalue" format, in which the year, month or quarter, and the associated values for each of the user-defined regression variables for a given observation are given, in this order, in free format on individual lines in the data file. Thus, a line of the data file with three regressors having the values 0, 0, and 1 respectively for July of 1991 would have the form `1991 7 0 0 1`. All the user-defined regressors must be on the same record, and in the order of their appearance in the **user** argument (example: `format="datevalue"`);
(c) the "x12save" format `X-12-ARIMA` uses to save a table. This allows the user to read in a file saved from a previous `X-12-ARIMA` run (example: `format="x12save"`).

If no **format** argument is given the data will be read in free format. In *free format*, all numbers on a line will be read before continuing to the next line, and the numbers must be separated by one or more spaces (not by commas or tabs). **Format** cannot be used with the **data** argument, only with **file**.

outliermethod Determines how the program successively adds detected outliers to the model. The choices are `method = addone` or `method = addall`. See the DETAILS section of the **outlier** spec for a description of these two methods. The default is `method = addone`. This argument cannot be used if the **sigma** argument is used.

outlierspan Specifies start and end dates of the span of the irregular component to be searched for outliers. The start and end dates of the span must both lie within the series, and the start date must precede the end date. A missing value, e.g., `outlierspan = (1976.jan, )`, defaults to the start date or end date of the series, as appropriate. (If there is a **span** argument in the **series** spec, then, in the above remarks, replace the start and end dates of the series by the start and end dates of the span given in the **series** spec.) This argument cannot be used with the **sigma** argument.

print and **save** Table 6-32 shows the tables available for output in the **x11regression** spec.

Table 6-32: Available Output Tables for X11regression

name	*default*	*brief*	*ext*	*description of table*
priortd	+	+	a4	prior trading day weights and factors
extremevalb	·	·	b14	irregulars excluded from the irregular regression, B iteration
x11regb	·	·	·	preliminary irregular regression coefficients and diagnostics
tradingdayb	·	·	b16	preliminary trading day factors and weights
combtradingdayb	·	·	b18	preliminary trading day factors from combined daily weights
holidayb	·	·	bxh	preliminary holiday factors
calendarb	·	·	bxc	preliminary calendar factors
combcalendarb	·	·	bcc	preliminary calendar factors from combined daily weights
extremeval	+	·	c14	irregulars excluded from the irregular regression, C iteration
x11reg	+	·	·	final irregular regression coefficients and diagnostics
tradingday	+	+	c16	final trading day factors and weights
combtradingday	+	+	c18	final trading day factors from combined daily weights
holiday	+	+	xhl	final holiday factors
calendar	+	+	xca	final calendar factors (trading day and holiday)
combcalendar	+	+	xcc	final calendar factors from combined daily weights
xroutlierhdr	+	·	·	options specified for outlier detection including critical value and outlier span
xroutlieriter	·	·	xoi	detailed results for each iteration of outlier detection including outliers detected, outliers deleted, model parameter estimates, and robust and non-robust estimates of the residual standard deviation
xroutliertests	·	·	·	t-statistics for every time point of each outlier detection iteration
xregressionmatrix	·	·	xrm	values of irregular regression variables with associated dates
xregressioncmatrix	·	·	xrc	correlation matrix of irregular regression parameter estimates if used with the **print** argument; covariance matrix of same if used with the **save** argument
xaictest	+	+	·	output from AIC-based tests for trading day and holiday

Name gives the name of each table for use with the **print** and **save** arguments.
Default indicates which tables are printed (+) or not printed (·) by default.
Brief indicates which tables are printed (+) or not printed (·) when the **brief** print level is specified. See section 5.1 for more information on print levels.
Ext gives the file extension used if the table is saved. A dot indicates that the table cannot be saved. The file extensions given can also be used as short names for the tables in the **print** and **save** arguments.

savelog Setting `savelog=aictest` or `savelog=ats` causes the results of the AIC-based selection procedure specified by the **aictest** argument to be output to the log file (see section 2.6 for more information on the log file).

sigma The sigma limit for excluding extreme values of the irregular components before trading day (only) regression is performed. Irregular values larger than this number of standard deviations from the mean (1.0 for multiplicative adjustments, 0.0 for additive adjustments) are excluded as extreme. Each irregular has a standard error determined by its month (or quarter) type. The month types are determined by the month length, by the day of the week on which the month starts. This argument cannot be used when regressors other than flow trading day are present in the model, or when the **critical** argument is used. The assigned value must be a real number greater than 0; the default is 2.5 (which is invoked only when the flow trading day variable(s) are the only regressor estimated). Example: `sigma=3.0`.

span Specifies the span (data interval) of irregular component values to be used to estimate the regression model's coefficients. This argument can be utilized when, for example, the user does not want data early in the series to affect regression estimates used for preadjustment before seasonal adjustment. As with the **modelspan** spec detailed in the **series** spec, the **span** argument has two values, the start and end date of the desired span. A missing value defaults to the corresponding start or end date of the span of the input series. For example, for monthly data, the statement `span=(1968.1, )` causes whatever irregular regression model is specified to be estimated from the time series data starting in January, 1968 and ending at the end date of the analysis span. A comma is necessary if either the start or the end date is missing. The start and end dates of the model span must both lie within the time span of data specified for analysis in the **series** spec, and the start date must precede the end date.

Another end date specification, with the form *0.per*, is available to set the ending date of **span** to always be the most recent occurrence of a specific calendar month (quarter for quarterly data) in the span of data analyzed, where *per* denotes the calendar month (quarter). Thus, if the span of data considered ends in a month other than December, `span=(,0.dec)` will cause the regression coefficients to stay fixed at the values obtained from data ending in December of the next-to-final calendar year of the span.

start The start date for the values of the user-defined regression variables. The default is the start date of the series. Valid values are any date up to the start date of the series (or up to the start date of the span specified by the **span** argument of the **series** spec, if present).

tdprior User-input list of seven daily weights, starting with Monday's weight, which specify a desired X-11 trading day adjustment prior to seasonal adjustment. These weights are adjusted to sum to 7.0 by the program. This option can be used only with multiplicative and log-additive seasonal adjustments. The values must be real numbers greater than or equal to zero. Example: `tdprior=(0.7 0.7 0.7 1.05 1.4 1.4 1.05)`.

user Specifies the list of names of user-defined regression variables. A name is required for each user-defined variable whose coefficients are to be estimated. The names given are used to label estimated coefficients in the program's output. Values for the user-defined variables must be supplied, using either the **data** or the **file** argument (not both). The maximum number of user-defined regression variables is 52. (This limit can be changed—see Section 2.8.)

usertype Assigns a type to the user-defined regression variables. The user-defined regression effects can be defined as a trading day (**td**), stock trading day (**tdstock**), holiday (**holiday**, **easter**, and the US holidays **thanks** and **labor**), additive outlier (**ao**), or other user-defined (**user**) regression effects. A single effect type can be specified for all the user-defined regression variables defined in the **x11regression** spec (`usertype=td`), or each user-defined regression variable can be given its own type (`user-type=(td td td td td td holiday user)`). See DETAILS for more information on assigning types to user-defined regressors.

variables List of predefined regression variables to be included in the model. The values of these variables are calculated by the program, as functions of the calendar in most cases. See DETAILS for a discussion and a table of the available predefined variables.

RARELY USED ARGUMENTS

aicdiff Defines the difference in AICC needed to accept a regressor specified in the **aictest** argument. The default value is `aicdiff=0.0`.

b Specifies initial values or fixed values for irregular component regression parameters in the order in which they appear in the **variables** and **user** arguments. If present, the **b** argument must assign initial values to *all* regression coefficients in the regARIMA model. Initial values are assigned to parameters either by specifying the value in the argument list or by explicitly indicating that it is missing as in the example below. Missing values take on the default value of 0.1. For example, for a model with two regressors, `b=(0.7, )` is equivalent to `b=(0.7,0.1)`, but `b=(0.7)` is not allowed. For a model with three regressors, `b=(0.8,,-0.4)` is equivalent to `b=(0.8,0.1,-0.4)`. To hold a parameter fixed at the specified value, immediately follow the value in the **b** list with an '`f`', e.g., `b=(0.7f, 0.1)`.

centeruser Specifies the removal of the (sample) mean or the seasonal means from the user-defined regression variables. If `centeruser=mean`, the mean of each user-defined regressor is subtracted from the regressor. If `centeruser=seasonal`, means for each calendar month (or quarter) are subtracted from each of the user-defined regressors. If this option is not specified, the user-defined regressors are assumed to already be in an appropriately centered form and are not modified.

eastermeans Specifies whether long term (400 year) monthly means are used to deseasonalize the Easter regressor associated with the variable `easter[w]`, as described in footnote 5 of Table 3-1 (`eastermeans=yes`), or, instead, monthly means calculated from the span of data used for the calculation of the coefficients of the Easter regressors (`eastermeans=no`). The default is `eastermeans=yes`. This argument is ignored if no built-in Easter regressor is included in the regression model, or if the only Easter regressor is `sceaster[w]` (see Details).

forcecal Specifies whether the calendar adjustment factors are to be constrained to have the same value as the product (or sum, if additive seasonal adjustment is used) of the holiday and trading day factors (`forcecal=yes`), or not (`forcecal=no`). The default is `forcecal=no`. This argument is functional only when both holiday and trading day regressors are specified in the **variables** argument of this spec.

noapply List of the types of regression effects defined in the **x11regression** spec whose model-estimated values are not to be adjusted out of the original series or final seasonally adjusted series. Available effects include modelled trading day effects (**td**) and Easter, Labor Day, and Thanksgiving-Christmas holiday effects (**holiday**).

reweight Specifies whether the daily trading day weights are to be reweighted when at least one of the daily weights in the C16 output table is less than zero (`reweight=yes`), or not (`reweight=no`). The default is `reweight=no`. This argument is functional only when trading day regressors are specified in the **variables** argument of this spec. **Note:** the default for previous versions of X-11 and X-11-ARIMA corresponds to `reweight=yes`.

umdata An input array of mean-adjustment values, to be subtracted from the irregular series I_t (or Log I_t) before the coefficients of a model with a user-defined regressor are estimated. This argument, or **umfile**, is used when the mean function for predefined regressors described in DETAILS is incorrect for the model with user-defined regressors. The mean-adjustment function depends on the mode of adjustment. See DETAILS for more information.

The time frame of these values must cover the time frame of the series (or of the span specified by the **span** argument of the **series** spec, if present). It must also cover the time frame of forecasts and backcasts requested in the **forecast** spec. The data values are read in free format. If the **umdata** argument is used, the **umfile** argument cannot be used.

umfile Name of the file containing a series of mean-adjustment values to be subtracted from the irregular series I_t (or Log I_t) before the coefficients of a model with a user-defined regressor are estimated. This replaces the mean function that is subtracted from I_t when only predefined regressors are used, as described in DETAILS. The filename must be enclosed in quotes. If the file is not in the current directory, the path must also be given. As with the **umdata** argument, the time frame of the data values must cover both the series and any forecasts or backcasts. If the **file** argument is used, the **umdata** argument cannot be used.

umformat Denotes the format used when reading the data for the regression variables from the file named in the **umfile** argument. Five types of input are accepted:

(a) a valid FORTRAN format, which must be enclosed in quotes and must include the initial and terminal parentheses (example: `umformat="(6f12.0)"`);
(b) "datevalue" format, in which the year, month or quarter, and the associated value for the mean-adjustment for a given observation are given, in this order, in free format on individual lines in the data file. Thus, a line of the data file with a mean adjustment of 1.01 for July of 1991 would have the form `1991 7 1.01` (example: `umformat="datevalue"`);
(c) the "x12save" format `X-12-ARIMA` uses to save a table. This allows the user to read in a file saved from a previous `X-12-ARIMA` run (example: `umformat="x12save"`).
(d) a two character code which corresponds to a set of data formats used in previous versions of X-11 and X-11-ARIMA (example: `umformat="1r"`);
(e) the format that the `TRAMO` and `SEATS` programs use to read in a series and its descriptors. This enables `X-12-ARIMA` to read in a data file formatted for the `TRAMO` modelling program or the `SEATS` seasonal adjustment program. (example: `umformat="tramo"`).

In the predefined X-11 data formats mentioned in (d), the data is stored in 6 or 12 character fields, with a year and series label associated with each year of data. For a complete list of these formats, see the DETAILS section of the **series** spec. If no **umformat** argument is given the data will be read in free format. **Umformat** cannot be used with the **umdata** argument, only with **umfile**.

umname The name of the series of values stored in the file named in **umfile**. The name must be enclosed in quotes and may contain up to 64 characters. Up to the first 16 characters will be printed as a label for the user-defined mean of the mean-adjustment values. When specified with the predefined formats of the **umformat** argument, the first six (or eight, if `umformat="cs"`) characters of this name are also used with the predefined formats to check that the program is reading the correct series, or to find a particular series in a file where many series are stored.

umprecision The number of decimal digits to be read from the user-defined mean. This option can only be used with the predefined formats of the **umformat** argument. This value must be an integer between 0 and 5, inclusive (for example, `umprecision=5`). The default is zero.

umstart The start date for the mean-adjustment values specified in **umdata** or **umfile**. The default is the start date of the series. Valid values are any date up to the start date of the series (or up to the start date of the span specified by the **span** argument of the **series** spec, if present).

umtrimzero If `umtrimzero=no`, zeroes at the beginning or end of the user mean time series entered via the umfile argument are treated as series values. The default (`umtrimzero=yes`) causes leading and trailing zeros to be ignored. Note that when the **format** argument is set to either `datevalue`, `x12save`, or `tramo`, all values input are treated as series values, regardless of the value of **umtrimzero**.

DETAILS

This spec is used to estimate a calendar effect, or other effect, from the irregular component I_t of a preliminary seasonal adjustment that did not adjust for the effect. The estimation is done by ordinary least squares (OLS) applied to a regression model for the effect. In the simplest cases detailed below, the model has the form

$$I_t - 1.0 = \beta' X_t + e_t,$$

where X_t is a regression vector with variables that describe the basic effect of interest. In other cases, a more complicated linear tranformation of I_t appears on the left of the model. In all cases, t-statistics, chi-square statistics, and AIC's are calculated from the OLS estimates as though the regression errors e_t were independent and had constant variance. Unfortunately, the filtering operations used to produce I_t guarantee that both assumptions about e_t are somewhat incorrect, enough that decisions made for the statistical significance of estimated effects from the statistics just mentioned are often less reliable than decisions made for effects estimated from a regARIMA model using the **regression** spec. That is, **x11regression's** statistics are more likely than **regression's** to suggest that a significant effect is present when it is not.. For effects that are

Table 6-33: Predefined regression variables for X11regression

Variable	Description
td	Estimates monthly (or quarterly) flow trading-day effects by adding the `tdnolpyear` variables (see Table 6-18) to the model. The derivations of February from the average length of 28.25 are handled either by rescaling (for multiplicative adjustments) or by including the `lpyear` regression variable (for additive and log-additive adjustments). `Td` cannot be used with `tdstock[]` or `td1coef`.
td1coef	Estimate monthly (or quarterly) flow trading-day effects by including the `td1nolpyear` variable (see below) in the model, and by handling leap-year effects either by rescaling (for transformed series) or by including the `lpyear` regression variable (for untransformed series). `Td1coef` can only be used for monthly or quarterly series, and cannot be used with `td` or `tdstock[]`.
tdstock[w]	Adds 6 stock trading-day variables to model the effect of the day of the week on a stock series estimated for the w^{th} day of each month. The value w must be supplied and can range from 1 to 31. For any month of length less than the specified w, the `tdstock` variables are measured as of the end of the month. Use `tdstock[31]` for end-of-month stock series. `Tdstock` can be used only with monthly series and cannot be used with `td` or `td1coef`.
easter[w]	Easter holiday regression variable (monthly or quarterly flow data only) which assumes the level of daily activity changes on the $w-$th day before Easter and remains at the new level until the day before Easter. The value w must be supplied and can range from 1 to 25. To estimate complex effects, several of these variables, differing in their choices of w, can be specified.
labor[w]	Labor Day holiday regression variable (monthly flow data only) that assumes the level of daily activity changes on the $w-$th day before Labor Day and remains at the new level until the day before Labor Day. The value w must be supplied and can range from 1 to 25.
thank[w]	Thanksgiving holiday regression variable (monthly flow data only) that assumes the level of daily activity changes on the $w-$th day before or after Thanksgiving and remains at the new level until December 24. The value w must be supplied and can range from -8 to 17. Values of $w < 0$ indicate a number of days after Thanksgiving; values of $w > 0$ indicate a number of days before Thanksgiving.
sceaster[w]	Statistics Canada Easter holiday regression variable (monthly or quarterly flow data only) assumes that the level of daily activity changes on the $(w-1)-$th day and remains at the new level through Easter day. The value w must be supplied and can range from 1 to 24. To estimate complex effects, several of these variables, differing in their choices of w, can be specified.
ao*date*	Adds an additive (point) outlier variable, AO, for the given date or observation number. For series with associated dates, AOs are specified as `ao`*date*. For monthly series the form is `ao`*year.month* (e.g., `ao1985.jul` or `ao1985.7`), for quarterly series it is `ao`*year.quarter* (e.g., `ao1985.1` for an AO in the first quarter of 1985). More than one AO may be specified. All specified outlier dates must occur within the series. (AOs with dates within the series but outside the span specified by the **span** argument of the **series** spec are ignored.)

truly significant, the estimates from the **regression** and **x11regression** specs are usually quite close. When they differ appreciably, those from **regression** are better more often than those from **x11regression**. (The forecast diagnostics of the **history** spec can be used to compare estimated effects for series of sufficient length, see Findley, Monsell, Bell, Otto and Chen (1998) and Soukup and Findley (2000).) Thus use of **x11regression** should normally be reserved for series for which the user is unable to find a regARIMA model with good fit over the data span of interest.

The irregular-component regression models for multiplicative decompositions.

The irregular component is presumed to have no seasonality or trend (beyond a constant level of 1.0, in the case of a multiplicative decomposition). Hence, the regressors that are used in regression models for the irregulars should usually not have a seasonal or trend component. For this reason, most trading day and easter regression functions used in the **regression** spec (see Tables 3-1 and 6-18) are modified for use in the **x11regression** spec (see Table 6-33). The modifications for trading day variables for the various types of seasonal adjustment decompositions are derived in section 1.4 of Findley, Monsell, Bell, Otto and Chen (1998). We will indicate the nature of this modification with a combined monthly flow trading day and holiday regression function of the form

$$\gamma_0 m_t + \sum_{j=1}^{6} \gamma_j (d_{j,t} - d_{7,t}) + \delta' \mathbf{H}_t, \tag{1}$$

where $d_{j,t}$ = no. of weekdays of type j in month t (with $j = 1, \ldots 7$ denoting Monday,. . ., Sunday, respectively), $m_t = \sum_{j=1}^{7} d_{j,t}$ (the length on month t in days), and $\mathbf{H}_t$ denotes a (column) vector of holiday regressors. Because of the definition of the calendar, over most time intervals of interest these variables are periodic, $m_{t+48} = m_{t,}$ $d_{j,t+336} = d_{j,t}$, and $\mathbf{H}_{t+P} = \mathbf{H}_t$ with P depending on the holiday variables included in $\mathbf{H}_t$. (If all proposed corrections to the Gregorian calendar are used, the Easter calendar has a period of $38,000$ years $= 456,000$ months. For this reason it is often more practical to choose P so that approximate periodicity holds, $\mathbf{H}_{t+P} \simeq \mathbf{H}_t$).

If f_t is an approximately periodic function of period $12p$ months, $f_{t+12p} \simeq f_t$, then its (approximate) combined seasonal and level component is given by its calendar month means

$$f_t^* = \frac{1}{p} \sum_{j=1}^{p} f_{t+12j},$$

which is approximately periodic with period 12 months, $f_{t+12}^* \simeq f_t^*$. If seasonal and level effects are removed from f_t by division, the resulting deseasonalized, level-neutral component of f_t is f_t / f_t^*. To apply these ideas to the function (1) above, we note that if p is a multiple of 28, then $d_{j,t}^* = d_{7,t}^*, 1 \leq j \leq 6$, with the result that the seasonal and level component of this calendar effect function is

$$\gamma_0 m_t^* + \delta' \mathbf{H}_t^*,$$

with

$$m_t^* = \begin{cases} m_t & , m_t = 30, 31 \\ 28.25 & , m_t = 28, 29 \end{cases}.$$

Therefore, if a time series contains a trading day and holiday component of the form (1), then its irregular component from multiplicative deseasonalization and detrending can be expected to have a trading day and holiday component close to

$$\frac{\gamma_0 m_t + \sum_{j=1}^{6} \gamma_j (d_{j,t} - d_{7,t}) + \delta' \mathbf{H}_t}{\gamma_0 m_t^* + \delta' \mathbf{H}_t^*} = \frac{\frac{m_t}{m_t^*} + \sum_{j=1}^{6} \alpha_j \left((d_{j,t} - d_{7,t})/m_t^* \right) + \beta' \frac{\mathbf{H}_t}{m_t^*}}{1 + \beta' \frac{\mathbf{H}_t^*}{m_t^*}}. \tag{2}$$

The expression on the right is a nonlinear function of $\alpha_j = \gamma_j/\gamma_0$ and $\beta = \delta/\gamma_0$. However, because trading day effects and holiday effects are usually in the range of a few percent, the approximation

$$\left(1 + \beta' \frac{\mathbf{H}_t^*}{m_t^*}\right)^{-1} \simeq 1 - \beta' \frac{\mathbf{H}_t^*}{m_t^*}$$

can be applied to (2). After multiplying the numerator on the right in (2) by this factor, the terms that involve products of coefficients are generally small enough to be ignored. This yields the following linear approximation to (2),

$$\frac{m_t}{m_t^*} + \sum_{j=1}^{6} \alpha_j (\frac{d_{j,t} - d_{7,t}}{m_t^*}) + \beta' \left(\frac{\mathbf{H}_t - \mathbf{H}_t^*}{m_t^*} \right). \tag{3}$$

In obtaining this approximation, we have also made use of

$$\frac{m_t}{m_t^*} = 1 + \frac{1}{28.25}(m_t - m_t^*),$$

and have treated the term involving the leap year variable $LY_t = m_t - m_t^*$ as one whose product with $\beta' \left(\mathbf{H}_t - \mathbf{H}_t^*\right)/m_t^*$ is negligible. The formula (3) suggests a linear regression model for the irregular component I_t of the form

$$I_t - \frac{m_t}{m_t^*} = \sum_{j=1}^{6} \alpha_j \left(\frac{d_{j,t} - d_{7,t}}{m_t^*} \right) + \beta' \left(\frac{\mathbf{H}_t - \mathbf{H}_t^*}{m_t^*} \right) + \kappa' \mathbf{AO}_t + e_t,$$

where $\mathbf{AO}_t$, denotes a regression vector containing any needed additive outlier variables. Instead of using this model, X-12-ARIMA, for conformity with the X-11 and X-11-ARIMA trading day regression models, obtains the coefficients in (3) from ordinary least squares estimation (OLS) of the rescaled model

$$m_t^* I_t - m_t = \sum_{j=1}^{6} \alpha_j (d_{j,t} - d_{7,t}) + \beta' \left(\mathbf{H}_t - \mathbf{H}_t^*\right) + \kappa' \mathbf{AO}_t + \varepsilon_t \tag{4}$$

whenever `td` is specified in the **variables** argument of **x11regression**, with one or more of the holiday effect specifications `easter[w]`, `labor[w]`, and `thank[w]`. As explained in the footnote of Table 3-1, the regressors associated with these holiday variables also have the deseasonalized form $\mathbf{H}_t - \mathbf{H}_t^*$ when they are estimated from the regression spec. This is done so that seasonal effects occur only in the seasonal part of the model, and only in the seasonal factors of the decomposition. For conformity with X-11-ARIMA/88, the regressors associated with `sceaster[w]` are never deseasonalized. In effect, the entries of $\mathbf{H}_t^*$ in (4) associated with any specified `sceaster[w]` regressors are set to zero.

Obtaining separate trading day and holiday factors. The calendar factors (3) can be approximately factored as the product of holiday factors

$$1 + \beta' \left(\frac{\mathbf{H}_t - \mathbf{H}_t^*}{m_t^*} \right) \tag{5}$$

and trading day factors

$$\frac{m_t}{m_t^*} + \sum_{j=1}^{6} \alpha_j \left(\frac{d_{j,t} - d_{7,t}}{m_t^*} \right) = \frac{\sum_{j=1}^{7} (1+\alpha_j) d_{j,t}}{m_t^*}, \tag{6}$$

(with $\alpha_7 = -\sum_{j=1}^{6} \alpha_j$). The numbers $1+\alpha_j$ are called the *daily weights.* The trading day factor formula (6) can also be written as

$$\frac{m_t}{m_t^*} + \frac{\sum_j^{(5)} \alpha_j}{m_t^*},$$

where $\sum_j^{(5)}$ denotes the sum of the j for which $d_{jt} = 5$. This formula shows that, apart from length of month effects, the trading day effects depend only on the effects of the days that occur five times in the month. When only trading day effects are estimated, the formulas above apply with $\beta = \mathbf{0}$.

If one or more trading day "weights" $1+\alpha_j$ are negative and the option **reweight** has been specified, then, for the trading day factor calculation all $\alpha_j < -1$ are replaced by $\alpha_j' = -1$ and all $\alpha_j \geq -1$ are replaced by $\alpha_j' = (1+\alpha_j)\, w - 1$, where

$$w = 7 \left\{ \sum_{\alpha_i \geq -1} (1+\alpha_i) \right\}^{-1},$$

assuming no $\alpha_j > -1$ have been assigned fixed values using the **b** argument. If there are fixed values, only unfixed $\alpha_j > -1$ are replaced, and in the replacement formula w is defined by

$$w = \left\{ 7 - \sum_{\alpha_i \text{fixed}} (1+\alpha_i) \right\} \left\{ \sum_{\alpha_i \text{notfixed}} (1+\alpha_i) \right\}^{-1},$$

for all $\alpha_i > -1$.

Estimating only holiday effects or stock trading day effects. If only holiday effects, or stock trading day effects, are specified in the **variables** argument of **x11regression**, then X-12-ARIMA estimates these effects by OLS applied to a model of the form

$$I_t - 1 = \beta' \left(\mathbf{H}_t - \mathbf{H}_t^*\right) + \kappa' \mathbf{AO}_t + e_t. \tag{7a}$$

respectively,

$$I_t - 1 = \sum_{j=1}^{6} \alpha_j D_{j,t} + \kappa' \mathbf{AO}_t + e_t. \tag{7b}$$

where the $D_{j,t}$ are the regressors associated with the specified `tdstock[w]` in Table 3-1. These models lead to calendar effect adjustment factors of the form

$$1 + \beta' \left(\mathbf{H}_t - \mathbf{H}_t^*\right). \tag{8a}$$

respectively,

$$1 + \sum_{j=1}^{6} \alpha_j D_{j,t} = 1 - \alpha_{j(t)}. \tag{8b}$$

where $\alpha_{j(t)}$ is the coefficient of the `w`-th day of month t.

Estimating user-defined flow trading day and/or holiday effects. The regression model (4) yields m_t/m_t^* as the component of the mean function for the irregulars I_t that is known independently of the estimated coefficients. This is also the default specification of the known component when user-defined variables are used. If this default is accepted, then the OLS regression model with at least one user-defined trading day or holiday variable has the form

$$m_t^* I_t - m_t = \alpha' \mathbf{TD}_t + \beta' \tilde{\mathbf{H}}_t + \kappa' \mathbf{AO}_t + \varepsilon_t, \tag{9}$$

with $\mathbf{TD}_t$ and $\tilde{\mathbf{H}}_t$ denoting the vectors of trading day and holiday variables specified. User-defined variables are input by way of **file** or **data** arguments. The program does not deseasonalize user-defined variables. They should be input to the progam in an appropriately deseasonalized form. X-12-ARIMA calculates calendar factors

$$\frac{m_t}{m_t^*} + \alpha' \frac{\mathbf{TD}_t}{m_t^*} + \beta' \frac{\tilde{\mathbf{H}}_t}{m_t^*},$$

that are approximately factored into holiday factors and trading day factors in analogy with (5), and (6). If only holiday effects, or stock trading day effects, are estimated, then the default known mean function component is the constant 1.0, and the model and resulting holiday factors are the analogues of (7) and (8).

When the default known mean functions just described are not appropriate, the user can input a mean function μ_t by means of the **umfile** or **umdata** arguments. In this case, the regression model estimated is

$$I_t - \mu_t = \alpha' \mathbf{TD}_t + \beta' \tilde{\mathbf{H}}_t + \kappa' \mathbf{AO}_t + e_t, \tag{10}$$

and only the calendar factors

$$\mu_t + \alpha' \mathbf{TD}_t + \beta' \tilde{\mathbf{H}}_t$$

are produced. The coefficients α,β, estimated from (10) are on a different scale from those obtained from (9), being smaller by roughly the factor

$$\frac{1}{48} \sum_{j=0}^{47} \frac{1}{m_t^*} \simeq .03288.$$

The same approximate scale difference holds for calendar coefficients calculated from **regression** instead of **x11regression**, or from (7) instead of (4).

The irregular-component regression models for other decomposition modes.

We present below the models used with additive, pseudo-additive, and log-additive decompositions for the case of combined flow trading day and holiday effect estimation with predefined regressors. The appropriate modifications to these models for the case of user-defined, stock trading day or holiday regressors are analogous to those described above for multiplicative decompositions.

Additive Decompositions. If `mode=add` in the **x11** spec, calendar effects are estimated by OLS from a model of the form

$$I_t = \alpha_0 LY_t + \sum_{j=1}^{6} \alpha_j (d_{j,t} - d_{7,t}) + \beta' \left(\mathbf{H}_t - \mathbf{H}_t^*\right) + \kappa' \mathbf{AO}_t + e_t.$$

The calendar effect is thus exactly the sum of the trading day effect $\alpha_0 LY_t + \sum_{j=1}^{6} \alpha_j (d_{j,t} - d_{7,t})$ and the holiday effect $\beta' \left(\mathbf{H}_t - \mathbf{H}_t^*\right)$.

Pseudo-Additive Decompositions. If `mode=pseudoadd` in the **x11** spec, then, with $\overline{m} = 30.4375$ and $LY_t = m_t - m_t^*$, calendar effects are estimated by OLS from a model of the form

$$\overline{m}\left(I_t - 1\right) - LY_t = \sum_{j=1}^{6} \alpha_j (d_{j,t} - d_{7,t}) + \beta' \left(\mathbf{H}_t - \mathbf{H}_t^*\right) + \kappa' \mathbf{AO}_t + \varepsilon_t.$$

The calendar effect factors

$$1 + \frac{1}{\overline{m}} LY_t + \sum_{j=1}^{6} \alpha_j \left(\frac{d_{j,t} - d_{7,t}}{\overline{m}}\right) + \beta' \left(\frac{\mathbf{H}_t - \mathbf{H}_t^*}{\overline{m}}\right)$$

can be approximately factored as

$$\left\{1+\frac{1}{m}LY_t+\sum_{j=1}^{6}\alpha_j\left(\frac{d_{j,t}-d_{7,t}}{m}\right)\right\}\left\{1+\beta'\left(\frac{\mathbf{H}_t-\mathbf{H}_t^*}{m}\right)\right\}$$

to obtain trading day and holiday factors.

Log-Additive Decompositions. If `mode=logadd` in the **x11** spec, calendar effects are estimated by OLS from a model of the form

$$m_t^*(\log I_t+1)-m_t=\sum_{j=1}^{6}\alpha_j(d_{j,t}-d_{7,t})+\beta'\left(\mathbf{H}_t-\mathbf{H}_t^*\right)+\kappa'\mathbf{AO}_t+\varepsilon_t. \quad (11)$$

These can be exactly factored into trading day and holiday factors,

$$\exp\left\{-1+\frac{m_t}{m_t^*}+\sum_{j=1}^{6}\alpha_j\left(\frac{d_{j,t}-d_{7,t}}{m_t^*}\right)\right\}\exp\left\{\beta'\left(\frac{\mathbf{H}_t-\mathbf{H}_t^*}{m_t^*}\right)\right\}.$$

Two other useful forms for the trading day factors appear in the identity

$$\exp\left\{-1+\frac{m_t}{m_t^*}\right\}\exp\left\{\sum_{j=1}^{6}\alpha_j\left(\frac{d_{j,t}-d_{7,t}}{m_t^*}\right)\right\}$$

$$\exp\left\{-1+\sum_{j=1}^{6}(1+\alpha_j)\frac{d_{j,t}}{m_t^*}\right\}, \quad (12)$$

which emphasizes the leap year factors $\exp\{-1+m_t/m_t^*\}\simeq m_t/m_t^*$ and the daily weights $(1+\alpha_j)$.

When tdprior is used.

Any of the coefficients in the models above can be assigned fixed values by an appropriate specification of the **b** argument. Sometimes users have prior information that suggests values for the seven daily weights associated with the trading factors (6) or (12) of multiplicative, respectively, log-additive adjustment. When "prior" daily weights $1+\alpha_j^{(p)}, 1\leq j\leq 7$ are assigned values by means of the **tdprior** argument, the series is preadjusted by

$$\frac{\sum_{j=1}^{7}(1+\alpha_j^{(p)})d_{j,t}}{m_t^*}=\frac{m_t}{m_t^*}\left\{1+\sum_{j=1}^{6}\alpha_j^{(p)}\left(\frac{d_{j,t}-d_{7,t}}{m_t}\right)\right\} \quad (13)$$

when `mode=mult` in **x11,** or by

$$\exp\left\{-1+\frac{m_t}{m_t^*}\right\}\exp\left\{\sum_{j=1}^{6}\alpha_j^{(p)}\left(\frac{d_{j,t}-d_{7,t}}{m_t^*}\right)\right\} \quad (14)$$

when `mode=logadd`. One advantage of using **tdprior** instead of **b** is that the user can also invoke **aictest** to automatically test whether significant trading day effects still occur in the irregular component of the preadjusted series and to calculate adjustment factors for removing any remaining effects. However, the fact that prior adjustment by (13) or (14) removes the leap year effect m_t/m_t^*, respectively $\exp\{-1 + m_t/m_t^*\}$, makes it necessary, when **td** is specified in the **variables** argument, to modify the models (4) and (12) used by **x11regression** for estimating remaining effects. When `mode=mult`, the model

$$m_t I_t - m_t = \sum_{j=1}^{6} \alpha_j (d_{j,t} - d_{7,t}) + \beta' \left(\mathbf{H}_t - \mathbf{H}_t^*\right) + \kappa' \mathbf{AO}_t + \varepsilon_t,$$

is used in place of (4), and, when `mode=logadd`,

$$m_t^* \log I_t = \sum_{j=1}^{6} \alpha_j (d_{j,t} - d_{7,t}) + \beta' \left(\mathbf{H}_t - \mathbf{H}_t^*\right) + \kappa' \mathbf{AO}_t + \varepsilon_t.$$

instead of (12). The first model yields the calendar factors

$$1 + \sum_{j=1}^{6} \alpha_j \left(\frac{d_{j,t} - d_{7,t}}{m_t}\right) + \beta' \left(\frac{\mathbf{H}_t - \mathbf{H}_t^*}{m_t}\right),$$

from which combined calendar factors are formed by multiplication with (13). The result is approximately

$$\frac{m_t}{m_t^*} \left\{ \sum_{j=1}^{7} (1 + \alpha_j^{(p)} + \alpha_j) \frac{d_{j,t}}{m_t} + \beta' \left(\frac{\mathbf{H}_t - \mathbf{H}_t^*}{m_t}\right) \right\}. \tag{15}$$

The second model yields the calendar factors

$$\exp\left\{ \sum_{j=1}^{6} \alpha_j \left(\frac{d_{j,t} - d_{7,t}}{m_t^*}\right) + \beta' \left(\frac{\mathbf{H}_t - \mathbf{H}_t^*}{m_t^*}\right) \right\},$$

and multiplication by (14) yields the combined factors

$$\exp\left\{ -1 + \sum_{j=1}^{6} (1 + \alpha_j^{(p)} + \alpha_j) \frac{d_{j,t}}{m_t^*} \right\}. \tag{16}$$

The formulas (15) and (16) show that a statistically significant t-statistic for a coefficient α_j can be interpreted as meaning that the prior weight $1 + \alpha_j^{(p)}$ needs significant revision.

This completes the discussion of the irregular component regression models and their factors.

The regression model specified by **x11regression** is estimated from the series of irregulars of the B and C iterations of the calculations of the **x11** spec. If the spec file also includes the **arima**, **automdl**, or **regression** spec, then the effects estimated via **x11regression** are obtained first, and they are removed from the data used for the estimations, or the forecast and backcast extensions, specified by these other specs. The series resulting from the calculations of these other specs is then decomposed by a second execution of the **x11** spec calculations in order to obtain the seasonal, trend, calendar-effect, and irregular components output by the program. Similarly, if the **x11** spec requests the Bateman-Mayes Easter-effect adjustment, this adjustment is calculated from a series that has been preadjusted for the effects estimated by **x11regression**.

If forecasting is performed, `X-12-ARIMA` creates data values for the selected predefined regression variables for the entire forecast period. If there are any user-defined regression variables, then data values must also be supplied for them for the entire forecast period. In addition to the limit of 52 user-defined regression variables, there is an overall limit of 80 regression variables in the model. (These limits can be changed—see Section 2.8.) The latter limit is on the sum of the number of predefined and user-defined regression variables, plus the number of regression variables generated from automatic outlier detection. The maximum length of the series of user-defined regression variables, not including the forecast period, is 600. (This limit can also be changed—see Section 2.8.)

Trading day and/or holiday adjustments must be obtained either from regARIMA or from irregular regression models, but not from both. If these effects are estimated in both the **regression** and **x11regression** spec, then the **noapply** option must be used to ensure that only one set of factors is used in the adjustment.

The effect of the argument **aictest** can be to delete a regressor set named in the **variables** argument from this list, or to add a regressor set to the model specified by the **variables** argument. The effect of a positive value of **aicdiff** is to make it more difficult for the **aictest** procedure to include in the model the variable being tested. Let Δ_{AIC} denote the value associated with the **aicdiff** argument, which by default is zero. Let $AICC^{with}$ (and $AICC^{without}$) denote the AICC value of the model with (or without) a set of regressors specified in the **aictest** argument. If this set is not named in the **variables** list, it will be added to the regression model if

$$AICC^{with} + \Delta_{AIC} < AICC^{without}.$$

If this set is named in the variables list, it will be retained in the regARIMA model only if this inequality holds.

In the second case, if `aictest = (tdstock)`, then the end-of-month stock variables, specified by `tdstock[31]`, are the variables added, because 31 is the default value for w in $tdstock[w]$.

There are more possibilities if `aictest = (easter)` and no Easter effect regressors appear in the **variables** argument. Then three additional models are considered, namely the models obtained by augmenting the specified regARIMA model with the regressor $easter[w]$ for $w = 1, 8, 15$ respectively. The Easter regressor whose model has the smallest AICC is retained if its AICC is smaller than the model with no Easter regressors by at least the amount $\Delta_{AIC} = 0$; otherwise, the model without Easter regressors is selected.

When trading day regressors appear in both the **aictest** and **variables** arguments, the type of regressors specified must be identical. The sample day for stock trading day variables and the date specified for change-of-regime regressors should **not** be included in the **aictest** argument; they will be assumed from the entry in the **variables** argument. For example, if `variables=(tdstock[15] ao1995.jan)`, then the entry for **aictest** should be `tdstock`.

If a trading day (**td** or **tdstock**) or holiday (**holiday**, or the specific US holidays **easter**, **thanks**, and **labor**) regressor type is assigned in to a user-defined variable with the **usertype** argument, the factor derived from the user-defined regression variables of that type will be combined with the regression factor from variables of the same type specified in the **x11regression** spec. The resulting factor will be adjusted out of the series for the seasonal adjustment factor calculations determined by the **x11** spec unless the type name appears in the **noapply** argument.

If **x11regression** is used in a spec file without an **x11** spec, then the irregular component used for the regression is that obtained from the default specification `x11{}`.

The two choices for the argument **eastermeans** yield noticeably different holiday factors. But the choice usually has negligible effects on the combined seasonal and holiday factors, because the seasonal factors change to compensate for the differences between the choices.

The monthly means used to obtain deseasonalized Easter regressors under `eastermeans=yes` were generated from frequencies of the date of Easter for the first 400 year period of the Gregorian calendar, 1583-1982. These frequencies are given in Montes (1997); the algorithm used to compute the date of Easter for the Gregorian calendar is given in Duffet-Smith (1981).

For a nonseasonal time series, an adjustment for trading day and holiday effects estimated by means of this spec can be obtained by setting `type=trend` in the **x11** spec.

When the `b=()` argument is used to fix coefficients, AIC and the other model selection statistics may become invalid, see the Details section of **estimate**.

EXAMPLES

The following examples show complete spec files.

Example 1 Multiplicative seasonal adjustment with all default options (so the program uses the moving seasonality ratio to select the seasonal filter length). The monthly series starts in January, 1976 and is stored in free format in the file `klaatu.dat` in the current directory. A trading day adjustment is done using a regression on the irregular component.

```
Series { File = "westus.dat"
         Start = 1976.1
       }
X11 {  }
X11Regression { Variables = td
              }
```

Example 2 Same as Example 1, only an AIC-based test will be performed to see if trading day and Easter regressors should be included in the regression on the irregular component.

```
Series { File = "westus.dat"
         Start = 1976.1
       }
X11 {  }
X11Regression { Variables = td
                Aictest = (td easter)
              }
```

Example 3 User-defined holiday regressors for the period both before and after Easter are included in the irregular regression along with trading day regressors. AO outlier identification will be performed during the irregular regression procedure.

```
series {
   file = "ukclothes.dat"
   start = 1985.Jan
}
x11 {  }
x11regression{
   variables = td            critical = 4.0
   user = (easter1 easter2)  file = "ukeaster.dat"
   usertype = holiday        start = 1980.Jan
}
```

Example 4 Perform a default seasonal adjustment. The trading day regressors in the **x11regression** spec will be fixed to their initial values; the Easter holiday regressor will be estimated.

```
series{
   format = '2R'
   title = 'MIDWEST ONE FAMILY Housing Starts'
   name = 'CMW1HS'
   file = 'cmw1hs.ori'
   span = (1964.01,1989.03)
}
x11{   }
x11regression{
        variables = (td easter[8])
        b = (  0.4453f  0.8550f -0.3012f  0.2717f
              -0.1705f  0.0983f -0.0082)
}
```

Example 5 Prior trading day weights are provided with this spec file. The trading day weights calculated during the irregular regression will be combined with these weights for a combined trading day component.

```
 series {
    file = "nzstarts.dat"  start = 1980.Jan
 }
 x11 {  }
 x11regression{
    variables = td
    tdprior = (1.4 1.4 1.4 1.4 1.4 0.0 0.0)
}
```

Example 6 Use an irregular component regression to estimate the trading day effect (with change-of-regime in January of 1990). The Bateman-Mayes procedure is specified to estimate the Easter holiday effect. These effects will be estimated first, and the seasonal adjustment specifed in the **x11** spec will be performed on the trading day and holiday adjusted series.

```
series{
    title = 'Motor Home Sales'
    start = 1967.1
    span = (1972.1, )
    name = 'SB0562'
    file = 'C:\final.x12\T0B05601.TXT'
    format = '2L'
   }
X11REGRESSION {
    variables = (td/1990.1/   labor[10]  thank[10] )
     }
x11{
    seasonalma = x11default
    sigmalim = (1.8 2.8)
    appendfcst = YES
    save = (D11 D16)
    x11easter = yes
   }
```

Example 7 The predefined regression effects are trading day variables and a constant. The user-defined regression variables are for strikes in 1980, 1985, and 1991 and are located in the file `strikes.dat`. The ARIMA part of the model is (0,1,1)(0,1,1)12. Since a model is specified in the spec, generate a year of forecasts by default. The seasonal period, 12, is not indicated since this is the default. Seasonally adjust the series after pre-adjusting for the user-defined regression effects. Adjust the series for X-11 trading day (using prior factors estimated elsewhere) and for Easter effects, and remove the holiday effects from the final seasonally adjusted series. A two-line seasonal adjustment title is specified.

```
  series{ title = "Automobile Sales"
          file = "carsales.dat"
          start = 1975.Jan }
  transform{ function = log }
  regression{ variables = (const)
              user = (strike80 strike85 strike90)
              file = "strike.dat"
              format = "(3f12.0)" }
  arima{ model = (0 1 1)(0 1 1)12 }
  x11{  title = ("Car Sales in US"
                 "Adjust for strikes in 80, 85, 90")
        save = seasonal appendfcst = yes
        x11easter = yes      }
  x11regression{   variables = td   }
```

References

Abraham, B., and G.E.P. Box (1978), "Deterministic and Forecast-Adaptive Time-Dependent Models," *Applied Statistics*, 27, 120-130.

Abraham, B., and J. Ledolter (1983), *Statistical Methods for Forecasting*, New York: John Wiley and Sons.

Akaike, H. (1973), "Information Theory and an Extension of the Likelihood Principle," in the *Second International Symposium on Information Theory*, eds. B.N. Petrov and F. Czaki, Budapest: Akademia Kiado, 267-287.

Baxter, M.A. (1994) "A Guide to Seasonal Adjustment of Monthly Data With X-11 (Third Edition), " Central Statistical Office, United Kingdom.

Bell, W.R. (1983), "A Computer Program for Detecting Outliers in Time Series," *Proceedings of the American Statistical Association, Business and Economic Statistics Section*, 634-639.

Bell, W.R. (1987), "A Note on Overdifferencing and the Equivalence of Seasonal Time Series Models With Monthly Means and Models With (0,1,1)12 Seasonal Parts When $\Theta = 1$," *Journal of Business and Economic Statistics*, 5, 383-387.

Bell, W.R. (1999), "An Overview of regARIMA Modeling," forthcoming Research Report, Statistical Research Division, U. S. Census Bureau.

Bell, W.R., and S.C. Hillmer (1983), "Modeling Time Series with Calendar Variation," *Journal of the American Statistical Association*, 78, 526-534.

Bobbitt, L. and Otto, M. C. (1990), "Effects of Forcasts on the Revisions of Seasonally Adjusted Values Using the X-11 Seasonal Adjustment Procedure," *Proceedings of the American Statistical Association, Business and Economic Statistics Section*, 449-453.

Box, G.E.P., and D.R. Cox (1964), "An Analysis of Transformations," *Journal of the Royal Statistical Society*, B, 26, 211-252.

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

Box, G.E.P., and G.C. Tiao (1975), "Intervention Analysis with Applications to Economic and Environmental Problems," *Journal of the American Statistical Association*, 70, 70-79.

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

Chang, I., and G.C. Tiao (1983), "Estimation of Time Series Parameters in the Presence of Outliers," Technical Report No. 8, Statistics Research Center, University of Chicago.

Chang, I., G.C. Tiao, and C. Chen (1988), "Estimation of Time Series Parameters in the Presence of Outliers," *Technometrics*, 30, 193-204.

Chen, B., and D.F. Findley (1998) "Comparison of X-11 and regARIMA Easter Holiday Adjustments," Statistical Research Division, U. S. Census Bureau, manuscript in preparation.

Cholette, P.A. (1978) "A Comparison and Assessment of Various Adjustment Methods of Sub-Annual Series to Yearly Benchmarks," Research Paper, Seasonal Adjustment and Time Series Staff, Statistics Canada.

Cholette, P.A. (1979) "A Comparison of Various Trend Cycle Estimators," Research Paper, Seasonal Adjustment and Time Series Staff, Statistics Canada.

Dagum, E.B. (1988) "X-11-ARIMA/88 Seasonal Adjustment Method - Foundations and Users Manual, " Statistics Canada.

Doherty, M. (1991) "Surrogate Henderson Filters in X-11," Technical Report, Statistics New Zealand, Wellington, New Zealand.

Doornik, J. A., and D.F. Hendry (2001) *GiveWin: An Interface to Empirical Modelling, 3rd edition*, London: Timberlake Consultants Press.

Duffet-Smith, P (1981) "Practical Astronomy With Your Calculator, 2nd Edition," Cambridge University Press.

Findley, D.F. (1985) "On the Unbiasedness Property of AIC for Exact or Approximating Linear Stochastic Time Series Models," *Journal of Time Series Analysis*, **6**, 229–252.

Findley, D.F. (1999) "Akaike's Information Criterion II" in the *Encyclopedia of Statistical Science, Update Volume 3*, eds. S. Kotz, C. B. Read, and D. L. Banks. New York: John Wiley and Sons, 2–6.

Findley, D.F. and C.C. Hood (1999). "X-12-ARIMA and Its Application to Some Italian Indicator Series," to appear in *Seasonal Adjustment Procedures – Experiences and Perspectives*, Istituto Nazionale di Statistica (ISTAT), Rome, 231-251.

Findley, D.F., B.C. Monsell, H.B. Shulman, and M.G. Pugh (1990), "Sliding Spans Diagnostics for Seasonal and Related Adjustments," *Journal of the American Statistical Association*, 85, 345-355.

Findley, D.F., B.C. Monsell, W.R. Bell, M.C. Otto, B.C. Chen (1998), "New Capabilities and Methods of the X-12-ARIMA Seasonal Adjustment Program," *Journal of Business and Economic Statistics*, 16, 127-176 (with Discussion).

Fuller, W.A. (1976), *Introduction to Statistical Time Series*, New York: John Wiley.

Geweke, J. (1978), "Revision of Seasonally Adjusted Time Series," SSRI Report No. 7822, University of Wisconsin, Department of Statistics.

Gomez, V., A. Maravall, and D. Peña (1999). "Missing Observations in ARIMA Models: Skipping Approach versus Additive Outlier Approach," `Journal of Econometrics`, 88, 341-363.

Griswold, R.E., and M.T. Griswold (1997) *The ICON Programming Language, 3rd Edition*, San Jose: Peer-to-Peer Communications.

Hampel, F.R., E.M. Ronchetti, P.J. Rousseeuw, and W.A. Stahel (1986), *Robust Statistics: The Approach Based on Influence Functions*, New York: John Wiley.

Hannan, E.J., and B.G. Quinn (1979), "The Determination of the Order of an Autoregression," *Journal of the Royal Statistical Society*, B, 41, 190-195.

Hillmer, S.C., and G.C. Tiao (1979), "Likelihood Function of Stationary Multiple Autoregressive Moving Average Models," *Journal of the American Statistical Association*, 74, 652-660.

Hood, C.C. (1998a) "X-12-Graph: A SAS/GRAPH Program for X-12-ARIMA Output, User's Guide for X-12-Graph Interactive for PC/Windows, Version 1.0", U. S. Census Bureau, Washington, DC.

Hood, C.C. (1998b) "X-12-Graph: A SAS/GRAPH Program for X-12-ARIMA Output, User's Guide for X-12-Graph Batch for PC/Windows, Version 1.0", U. S. Census Bureau, Washington, DC.

Huot, G. (1975) "Quadratic Minimization Adjustment of Monthly or Quarterly Series to Annual Totals," Research Paper, Seasonal Adjustment and Time Series Staff, Statistics Canada.

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

Leser, C.E.V. (1963), "Estimation of Quasi-Linear Trend and Seasonal Variation," *Journal of the American Statistical Association*, 58, 1033-1043.

Ljung, G.M. (1993), "On Outlier Detection in Time Series," *Journal of the Royal Statistical Society*, B, 55, 559-567.

Ljung, G.M., and G.E.P. Box (1978), "On a Measure of Lack of Fit in Time Series Models," *Biometrika*, 65, 297-304.

Ljung, G.M., and G.E.P. Box (1979), "The Likelihood Function of Stationary Autoregressive-Moving Average Models," *Biometrika*, 66, 265-270.

Lothian, J. and M. Morry (1978), "A Test of Quality Control Statistics for the X-11-ARIMA Seasonal Adjustment Program," Research Paper, Seasonal Adjustment and Time Series Staff, Statistics Canada.

Monsell, B.C. (1989) "Supplement to Technical Paper No. 15 - The Uses and Features of X-11.2 and X-11Q.2," Statistical Research Division, U. S. Census Bureau.

Montes, M. J. (1997), "Frequency of the Date of Easter over one 400 year Gregorian Cycle" [Online]. Available: `http://www.smart.net/~mmontes/freq2.html` [1999, December 8].

Montes, M. J. (1998), "Calculation of the Ecclesiastical Calendar" [Online]. Available: `http://www.smart.net/~mmontes/ec-cal.html` [1999, December 8].

More, J.J., B.S. Garbow, and K.E. Hillstrom (1980), User Guide for MINPACK-1, Report ANL-80-74, Argonne National Laboratory, Argonne, Illinois.

Otto, M. C., Bell, William R. and Burman, J. P. (1987a), "An Iterative GLS Approach to Maximum Likelihood Estimation of Regression Models with ARIMA Errors," Proceedings of the American Statistical Association, Business and Economic Statistics Section, pp. 632-637.

Otto, M. C., Bell, William R. and Burman, J. P. (1987b), "An Iterative GLS Approach to Maximum Likelihood Estimation of Regression Models with ARIMA Errors," SRD Research Report No. 87/34, Bureau of the Census.

Otto, M.C., and W.R. Bell (1990), "Two Issues in Time Series Outlier Detection Using Indicator Variables," *Proceedings of the American Statistical Association, Business and Economic Statistics Section*, 182-187.

Otto, M.C., and W.R. Bell (1993), "Detecting Temporary Changes in Level in Time Series," *Proceedings of the American Statistical Association, Business and Economic Statistics Section*, 170-174.

Pierce, D.A. (1971), "Least Squares Estimation in the Regression Model With Autoregressive-Moving Average Errors," *Biometrika*, 58, 299-312.

SAS Institute Inc. (1990), *SAS/GRAPH Software: Reference, Version 6, First Edition, Volume 1*, Cary, NC: SAS Institute.

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

Shiskin, J., A.H. Young, and J.C. Musgrave (1967) "The X-11 Variant of the Census Method II Seasonal Adjustment Program," Technical Paper No. 15, U.S. Department of Commerce, U. S. Census Bureau.

Soukup, R.J. and Findley, D. F. (2000) "Detection and Modeling of Trading Day Effects," to appear as an SRD Technical Report, U. S. Census Bureau.

Thomson, P., and T. Ozaki (1992) "Transformation and Seasonal Adjustment," submitted for publication.

Vandaele, W. (1983), *Applied Time Series and Box-Jenkins Models*, New York: Academic Press.

Wilson, G.T. (1983), "The Estimation for Time Series Models, Part I. Yet Another Algorithm for the Exact Likelihood of ARMA Models," Technical Report No. 2528, Mathematics Research Center, University of Wisconsin-Madison.