U. S. Census Bureau

X-12-ARIMA
Reference Manual

Version 0.2.10

July 26, 2002

Time Series Staff
Statistical Research Division
Room 3000-4
U. S. Census Bureau
Washington, DC 20233-9100
email: x12@census.gov
WWW: http://www.census.gov/srd/www/x12a/

CONTENTS

TABLES

1. Introduction

The X-12-ARIMA seasonal adjustment program is an enhanced version of the X-11 Variant of the Census Method II seasonal adjustment program (Shiskin, Young, and Musgrave 1967). The enhancements include a more self-explanatory and versatile user interface and a variety of new diagnostics to help the user detect and remedy any inadequacies in the seasonal and calendar effect adjustments obtained under the program options selected. The program also includes a variety of new tools to overcome adjustment problems and thereby enlarge the range of economic time series that can be adequately seasonally adjusted. Examples of the use of these tools can be found in Findley and Hood (1999).

The chief source of these new tools is the extensive set of time series model building facilities built into the program for fitting what we call regARIMA models. These are regression models with ARIMA (autoregressive integrated moving average) errors. More precisely, they are models in which the mean function of the time series (or its logs) is described by a linear combination of regressors, and the covariance structure of the series is that of an ARIMA process. If no regressors are used, indicating that the mean is assumed to be zero, the regARIMA model reduces to an ARIMA model. There are built-in regressors for directly estimating various flow and stock trading day effects and holiday effects. There are also regressors for modelling certain kinds of disruptions in the series, or sudden changes in level, whose effects need to be temporarily removed from the data before the X-11 methodology can adequately estimate seasonal adjustments. To address data problems not provided for, there is the capability of incorporating user-defined regression variables into the model fitted. The regARIMA modelling module of X-12-ARIMA was adapted from the regARIMA program developed by the Time Series Staff of Census Bureau's Statistical Research Division.

Whether or not special problems requiring the use of regressors are present in the series to be adjusted, a fundamentally important use of regARIMA models is to extend the series with forecasts (and backcasts) in order to improve the seasonal adjustments of the most recent (and the earliest) data. Doing this mitigates problems inherent in the trend estimation and asymmetric seasonal averaging processes of the type used by the X-11 method near the ends of the series. The provision of this extension was the most important technical improvement offered by Statistics Canada's widely used X-11-ARIMA program. Its benefits, both theoretical and empirical, have been documented in many publications, including Geweke (1978), Dagum (1988) and Bobbitt and Otto (1990) and the articles referenced in these papers.

X-12-ARIMA is available as an executable program for PC microcomputers (386 or higher with a math coprocessor) running DOS (version 3.0 or higher), Sun 4 UNIX workstations, and VAX/VMS computers. FORTRAN source code is available for users to create executable programs on other computer systems. When it is released, the X-12-ARIMA program will be in the public domain, and may be copied or transferred. Computer files containing the current test version of the program (executables for various machines and source code), this documentation, and examples, have been put on the Internet at `http://www.census.gov/srd/www/x12a/`; users can also access these files via via anonymous ftp at `ftp.census.gov` in the directory `pub/ts/x12a` (login as "`anonymous`" and give your full email address as the password.) Information on downloading the relevant files is given in the `ReadMe` file in this directory. Limited program support is available via regular mail and email (the preferred mode of communication) at the addresses given on the title page. If problems are encountered running a particular input file, providing the input and resulting output files will facilitate our identification of the problem.

The seasonal adjustment module uses the `X-11` seasonal adjustment method detailed in Shiskin, Young and Musgrave (1967) and Dagum (1988). The program has all the seasonal adjustment capabilities of the `X-11` and `X-11-ARIMA` programs. The same seasonal and trend moving averages are available, and the program still offers the `X-11` calendar and holiday adjustment routines.

The seasonal adjustment module has also been enhanced by the addition of several new options, including

(a) the sliding spans diagnostic procedures, illustrated in Findley, Monsell, Shulman and Pugh (1990);
(b) the ability to produce the revisions history of a given seasonal adjustment;
(c) a new Henderson trend filter routine which allows the user to choose any odd number for the length of the Henderson filter;
(d) new options for seasonal filters;
(e) several new outlier detection options for the irregular component of the seasonal adjustment;
(f) a table of the trading day factors by type of day;
(g) a pseudo-additive seasonal adjustment mode.

The modelling module of `X-12-ARIMA` is designed for regARIMA model building with seasonal economic time series. To this end, several categories of predefined regression variables are available in `X-12-ARIMA`, including trend constants or overall means, fixed seasonal effects, trading-day effects, holiday effects, pulse effects (additive outliers), level shifts, temporary change outliers, and ramp effects. User-defined regression variables can also be easily read in and included in models. The program is designed around specific capabilities needed for regARIMA modelling, and is not intended as a general purpose statistical package. In particular, `X-12-ARIMA` should be used in conjunction with other (graphics) software capable of producing high resolution plots of time series.

Observations (data) from a time series to be modelled and/or seasonally adjusted using `X-12-ARIMA` should be quantitative, as opposed to binary or categorical. Observations must be equally spaced in time, and missing values are not allowed. `X-12-ARIMA` handles only univariate time series models, i.e., it does not estimate relationships between different time series.

`X-12-ARIMA` uses the standard $(p\ d\ q)(P\ D\ Q)_s$ notation for seasonal ARIMA models. The $(p\ d\ q)$ refers to the orders of the *nonseasonal* autoregressive (AR), differencing, and moving average (MA) operators, respectively. The $(P\ D\ Q)_s$ refers to the *seasonal* autoregressive, differencing, and moving average orders. The s subscript denotes the seasonal period, e.g., $s = 12$ for monthly data. Great flexibility is allowed in the specification of ARIMA structures: any number of AR, MA, and differencing operators may be used; missing lags are allowed in AR and MA operators; and AR and MA parameters can be fixed at user-specified values.

For the user who wishes to fit customized time series models, `X-12-ARIMA` provides capabilities for the three modelling stages of *identification*, *estimation*, and *diagnostic checking*. The specification of a regARIMA model requires specification both of the regression variables to be included in the model and also the type of ARIMA model for the regression errors (i.e., the orders $(p\ d\ q)(P\ D\ Q)_s$). Specification of the regression variables depends on user knowledge about the series being modelled. *Identification* of the ARIMA model for the regression errors follows well-established procedures based on examination of various sample autocorrelation and partial autocorrelation functions produced by `X-12-ARIMA`. Once a regARIMA model has been specified, `X-12-ARIMA` will *estimate* its parameters by maximum likelihood using an iterative generalized least squares (IGLS) algorithm. *Diagnostic checking* involves examination of residuals from the fitted model for signs of model inadequacy. `X-12-ARIMA` produces several standard residual diagnostics for model checking, as well as providing sophisticated methods for detecting additive outliers and level shifts. Finally, `X-12-ARIMA` can produce point forecasts, forecast standard errors, and prediction intervals from the fitted regARIMA model.

In addition to these modelling features, `X-12-ARIMA` has an automatic model selection procedure and an option which uses AICC to determine if user-specified regression variables (such as trading day or Easter regressors) should be included into a particular series. Also, histories can be generated for likelihood statistics (such as AICC, a version of Akaike's AIC that adjusts for the length of the series being modeled) and forecasts to facilitate comparisons between competing models.

The next five sections detail capabilities of the `X-12-ARIMA` program. Section 2 provides an overview of running `X-12-ARIMA` and explains program limits that users can change. Section 3 discusses the general regARIMA model fit by the `X-12-ARIMA` program, summarizes the technical steps involved in regARIMA modelling and forecasting, and relates these steps to capabilities of the program. Section 4 discusses some key points related to model estimation and inference that all users of the modelling features should be aware of, including some estimation problems that may arise and ways to address them. Section 5 provides a general description of the required input file (specification file), and also discusses specification file syntax and related issues. The focus in Sections 2–5 is on giving an overview of the use and capabilities of the `X-12-ARIMA` program. In contrast, Section 6 is intended as the primary reference to be used when constructing specification files for running `X-12-ARIMA`. It gives detailed documentation for each specification statement that can appear in the specification file. These statements function as commands that control the flow of `X-12-ARIMA`'s execution and select among the various program options.

2. Running X-12-ARIMA

Procedures for installing `X-12-ARIMA` are machine-specific; information about this is provided with the program, and is also available in the file `/pub/ts/x12a/ReadMe` on the Internet at `ftp.census.gov`. Having installed the program on a microcomputer running a DOS operating system, a generic statement to run `X-12-ARIMA` is

path\\`x12a` *path*\\`filename`

In this statement *path*\\`filename.spc` is the main input (specification) file used by `X-12-ARIMA`. The program created a file named *path*\\`filename.out` as an output file. The *path* to `X-12-ARIMA` is necessary if the file containing the `X-12-ARIMA` program is not in the current directory; similarly for the *path* to the input file `filename.spc`.

Note that only the filename is specified, not the extension; the program will use the filename provided at runtime to form the filenames for all files generated by the program. For an `X-12-ARIMA` run using the spec file *filename*.spc, the output will be stored in the file *filename*.out, the error messages will be stored in the file *filename*.err, etc. Thus, if the spec file `xuu1.spc` is in a DOS PC's current directory, typing

`x12a xuu1`

and pressing the `<return>` (or `<enter>` key) will cause the program to run and create files `xuu1.out` and `xuu1.err` in the current directory.

Program input and output are both discussed briefly below, and more extensively in the documentation that follows. To run the program under a UNIX operating system, substitute (forward) slashes for the backslashes in the generic statements above. To run `X-12-ARIMA` under other operating systems, specify paths, etc., using the syntax appropriate for the system. For the DOS, UNIX and VAX/VMS operating systems, a quick reference document is also available, giving more detailed instructions on the syntax for running `X-12-ARIMA` in these operating systems.

2.1 Input

To apply `X-12-ARIMA` to any particular time series, a main input file, called a *specification file*, must be created. This ASCII (or "text") file contains a set of specifications or *specs* that `X-12-ARIMA` reads to obtain the information it needs about the time series data, the time series model to be used, the analysis to be performed, and the output desired. `X-12-ARIMA` assumes that the specification file has the extension `.spc`. Thus *path*\\`filename` is sufficient in the above statements. The only input files other than the spec file that `X-12-ARIMA` may need are optional files containing data for the time series being modelled, data for any user-defined regression variables, values for any user-defined prior-adjustment factors, and model types to try with the automatic model selection procedure. The names of these files (including paths) are provided to `X-12-ARIMA` by listing them in appropriate specs in the spec file. The use of such additional input files is optional because the user can alternatively include the data values required in appropriate places in these specs, and a default set of models for the automatic modelling procedure is available. Section 6 explains how to write spec files.

2.2 Output

The usual output is written to the file *path*\`filename.out`. Individual specs control their contribution to this output using optional **print** arguments (discussed in Section 5.1). The **save** argument is used to create certain other output files for further analysis (for example, to save a time series of residuals for plotting using a graphics program). Cautionary note: When **save** is used, the program constructs the name of the file to which the specified output is written using naming conventions discussed in Section 5.1. If a file with this name already exists, it will be overwritten by `X-12-ARIMA`. Users should thus take suitable precautions when saving output. See Section 5.1 for more information.

2.3 Input errors

Input errors are reported as they are discovered by `X-12-ARIMA`, which then prints appropriate error messages. These error messages are also stored in a file named *path*\`filename.err`. When the program can localize the error, the line in the spec file containing the error will be printed out with a caret (`^`) positioned under the error. If the program cannot localize the error, then only the error message will be printed. If the error is fatal, then `ERROR:` will be displayed before the error message, sometimes with suggestions about what to change. For nonfatal errors, `WARNING:` will be printed before the message. `WARNING` messages are also used sometimes to call attention to a situation in which no error has been committed, but some caution is appropriate.

`X-12-ARIMA` first reads the whole spec file, reporting all input errors it finds. This way the user can try to correct more than one input error per run. Frequently, however, the only informative messages are those for the first one or two errors. These errors may result in other errors, especially if input errors occur in the **series** spec. The program will stop if any fatal errors are detected. Warnings will not stop the program, but should alert users to check both the input and output carefully to verify that the desired results are produced.

2.4 Specifying an alternate output filename

As was noted before, for an `X-12-ARIMA` run using the spec file *filename*.spc, the output will be stored in the file *filename*.out, the error messages will be stored in the file *filename*.err, etc. For the purpose of examining the effects of different adjustment and modelling options on a given series, it is sometimes desirable to use a different filename for the output than was was used for the input. The general form for specifing an alternate filename for the output files is

path\`x12a` *path*\`filename` *path*\`outname` (2.1)

This `X-12-ARIMA` run still uses the spec file *filename*.spc, but the output will be stored in the file *outname*.out, the error messages will be stored in the file *outname*.err, etc.

All output files generated by this run will be stored using the path and filename given by the user, not the path and filename of the input specification file.

2.5 Running X-12-ARIMA on more than one series

In a production situation, it is essential to run more than one series in a given X-12-ARIMA run. X-12-ARIMA allows for running multiple series in two modes:

(a) **multi-spec mode**, where there are input specification files for every series specified;
(b) **single spec mode**, where every series will be run with the options from a single input specification file.

Before X-12-ARIMA can be run in either mode, a *metafile* must be created. This is an ASCII file which contains the names of the files to be processed by X-12-ARIMA. Two types of metafiles are used by X-12-ARIMA: input metafiles (for multi-spec mode) and data metafiles (for single spec mode).

If an error occurs in one of the spec files in a metafile run, the program will print the appropriate error messages. Execution will stop for that series and the program will continue processing the remaining spec files. A listing of all the input files with errors is given in the X-12-ARIMA log file, described in Section 2.7.

2.5.1 Running X-12-ARIMA in multi-spec mode

Before X-12-ARIMA can be run in multi-spec mode, an *input metafile* must first be created. This is an ASCII file which contains the names of the files to be used by X-12-ARIMA. An input metafile can have up to two entries per line: the filename (and path information, if necessary) of the input specification file for a given series, and an optional output filename for the output of that series. If an output filename is not given by the user, then the path and filename of the input specification file will be used to generate the output files. The input specification files are processed in the order in which they appear in the input metafile.

For example, to run the spec files xuu1.spc, xuu2.spc and xuu3.spc, the input metafile should contain the following:

```
xuu1
xuu2
xuu3
```

This assumes that all these spec files are in the current directory. To run these files if they are stored in the c:\export\specs DOS directory, the metafile should read:

```
c:\export\specs\xuu1
c:\export\specs\xuu2
c:\export\specs\xuu3
```

To run X-12-ARIMA with a input metafile, use the following syntax:

```
x12a -m metafile
```

where `metafile.mta` is the metafile and `-m` is a flag which informs `X-12-ARIMA` of the presence of a metafile.

For example, if the metafile defined above is stored in `exports.mta`, type

```
x12a -m exports
```

and press the return key to run the corresponding spec files.

Note that when the name of the input metafile was given in the example above, only the filename was specified, not the extension; `.mta` is the required extension for the input metafile. Path information should be included with the input metafile name, if necessary.

The filenames used by `X-12-ARIMA` to generate output files are taken from the spec files listed in the metafile, not from the metafile itself. The example given above would generate output files named `xuu1.out, xuu2.out` and `xuu3.out` corresponding to the individual spec files given in the metafile `exports.mta`, not a comprehensive output file named `exports.out`. To specify alternate output filenames for the example above, simply add the desired output filenames to each line of the input metafile, e.g.,

```
c:\export\specs\xuu1  c:\export\output\xuu1
c:\export\specs\xuu2  c:\export\output\xuu2
c:\export\specs\xuu3  c:\export\output\xuu3
```

2.5.2 Running `X-12-ARIMA` in single spec mode

To run `X-12-ARIMA` on many series using the same specification commands for each series, it is necessary to create a *data metafile*. A data metafile can have up to two entries per line: the complete filename (and path information, if necessary) of the data file for a given series, and an optional output filename for the output of that series. If an output filename is not given by the user, then the path and filename of the data file will be used to generate the output files. **Note:** In a data metafile, no extension is assumed for the individual data files. The extensions must be specified, along with the path and filename, if the data files are not in the current directory.

The data files are processed in the order in which they appear in the data metafile. The options used to process each data file are provided by a single input specification file identified at runtime. This means that **all** the data files specified in the data metafile must be in the same format. Also, certain formats supported by `X-12-ARIMA` should be avoided; see the description of the **series** spec in Section 6 for more details.

For example, to process the data files xuu1.dat, xuu2.dat and xuu3.dat, the data metafile should contain the following:

```
xuu1.dat
xuu2.dat
xuu3.dat
```

This assumes that all these data files are in the current directory. To run these files if they are stored in the c:\export\data DOS directory, the metafile should read:

```
c:\export\data\xuu1.dat
c:\export\data\xuu2.dat
c:\export\data\xuu3.dat
```

To run `X-12-ARIMA` with a data metafile, use the following syntax:

```
x12a specfile -d metafile
```

where `metafile.dta` is the data metafile, `-d` is a flag which informs `X-12-ARIMA` of the presence of a data metafile, and `specfile.spc` is the single input specification file used for each of the series listed in the data metafile.

For example, if the data metafile with three series used for illustration above is named `exports.dta`, type

```
x12a default -d exports
```

and press the return key to process the corresponding data files using the `default.spc` input specification file.

Note that when the name of the data metafile was given in the example above, only the filename was specified, not the extension; `.dta` is the required extension for the input metafile. Path information should be included with the data metafile name, if necessary.

The filenames used by `X-12-ARIMA` to generate output files are taken from the data files listed in the metafile, not by the metafile itself. The example given above would generate output files named `xuu1.out, xuu2.out` and `xuu3.out` corresponding to the individual data files given in the metafile `exports.dta`, not a comprehensive output file named `exports.out`. To specify alternate output filenames for the example above, simply add the desired output filenames to each line of the data metafile, e.g.,

```
c:\export\data\xuu1.dat  c:\export\output\xuu1
c:\export\data\xuu2.dat  c:\export\output\xuu2
c:\export\data\xuu3.dat  c:\export\output\xuu3
```

2.6 Log Files

Every time `X-12-ARIMA` is run, a **log file** is produced where a summary of modeling and seasonal adjustment diagnostics can be stored for every series or spec file processed. When `X-12-ARIMA` is run in multi-spec or single spec model, as described in the previous section, the log file is stored with the same name and directory as the metafile (for multi-spec mode) or data metafile (single spec mode), with an extension of `.log`. For example

```
x12a -m exports
```

runs each of the spec files stored in exports.mta and stores user-selected diagnostics into the log file `exports.log`.

If only one series is processed, the output directory and filename is used along with the `.log` file extension to form the name of the log file.

Users can specify which diagnostics are stored in the log file by using the **savelog** argument found in the **series**, **composite**, **transform**, **x11**, **x11regression**, **regression**, **automdl**, **estimate**, **check**, **slidingspans**, and **history** specs. The descriptions of the individual specs in Section 6 give more details on which diagnositcs can be stored in the log file.

As mentioned in the previous section, if an error occurs in one of the spec files in a metafile run, a listing of all the input files with errors is given in the log file.

2.7 Flags

In the previous section, the flags **-m** and **-d** were required in the command line to obtain the desired run. There are several other input and output options that are specified on the command line. The general syntax for the command line can be given as

path\\`x12a` *arg1* *arg2* $\cdots$ *argN*

where the arguments given after x12a can be either flags or filenames, depending on the situation.

Table 2-1 gives a summary of the flags available in `X-12-ARIMA`; the remainder of this section will describe what each flag means in more detail. These flags can be specified in any order on the command line. (Some must be followed by appropriate filenames).

Table 2-1: X-12-ARIMA Program Flags

flags	*description of flag*
-c	Sum each of the components of a composite adjustment, but only perform modelling or seasonal adjustment on the total
-d *filename*	Filename (without extension) for data metafile
-g *dirname*	Directory where graphics metafile and related files for input to external graphics programs are stored
-i *filename*	Filename (without extension) for input specification file
-m *filename*	Filename (without extension) for input metafile
-n	(No tables) Only tables specifically requested in the input specification file will be printed out
-o *filename*	Filename (without extension) used for all output files generated during this run of the program
-p	No pagination is used in main output file
-q	Run X-12-ARIMA in quiet-mode (warning messages not sent to the console)
-r	Produce reduced X-12-ARIMA output (as in GiveWin version of X-12-ARIMA)
-s	Store seasonal adjustment diagnostics in a file
-v	Only check input specification file(s) for errors; no other processing
-w	Wide (132 character) format is used in main output file

The **-m** and **-d** flags were described in the previous section. Note that one cannot specify both of these flags in the same run.

The **-i** flag indicates that the next argument is the path and filename of the input specification file. This flag does not need to be specified as long as the input specification file is the first argument; therefore, `x12a test` and `x12a -i test` are equivalent. The **-i** and **-m** flags cannot be specified in the same run.

Similar to **-i**, the **-o** flag indicates that the next argument is the path and filename for the output. The output extensions described earlier (`.out` and `.err`) as well as extensions associated with the **save** command will be appended to this filename. This flag also does not need to be specified as long as the input specification file is the first argument and the output filename is the second argument (as in Equation 2.1). So any of the following commands are equivalent:

```
x12a test test2
x12a -i test -o test2
x12a -o test2 -i test
```

However, `x12a -i test test2` will generate an error, since the first argument is the flag **-i**, not the spec file. The **-o** flags cannot be specified in the same run as the **-m** or **-d** flags. The **-o** and **-m** flags cannot be specified in the same run.

The **-s** flag specifies that certain seasonal adjustment and regARIMA modeling diagnostics that appear in the main output be saved in file(s) separate from the main output. These include tables in the main output file that are not tables of time series. Such tables cannot be stored in the format used for individual time series tables. When the **-s** flag is used, `X-12-ARIMA` automatically stores the most important of these diagnostics in a separate file that can be used to generate diagnostic summaries. This file (called the *seasonal adjustment diagnostics file*) will have the same path and filename as the main output, with the extension **.xdg**. So for

```
x12a test -s
```

the seasonal adjustment diagnostic file will be stored in `test.xdg`, and for

```
x12a test -s -o testout
```

the seasonal adjustment diagnostic file will be stored in `testout.xdg`.

In addition to the seasonal adjustment diagnostics file, the program will also store a number of important regARIMA modeling diagnostics into a *model diagnostics file* when the **-s** flag is used. This file is only produced when a regARIMA model is estimated in the `X-12-ARIMA` run. The model diagnostic file will also have the same path and filename as the main output, with the extension **.mdg**.

So for

```
x12a test -s
```

the model diagnostic file will be stored in `test.mdg`, and for

```
x12a test -s -o testout
```

the model diagnostic file will be stored in `testout.mdg`.

A program is available via the Internet on `ftp.census.gov` that reads the seasonal adjustment diagnostics file and produces a summary of the seasonal adjustment diagnostics. This program is written in the Icon programming language (see Griswold and Griswold (1997)).

The **-g** flag indicates that the next argument is the complete path name of a directory into which output will be stored that is intended as input for a separate graphics program. This output consists of the following files:

(1) files of diagnostic data to be graphed, which are produced by the options specified in the .spc file;
(2) a *graphics metafile* containing the names of these files;
(3) a *seasonal adjustment diagnostics file* containing information about the time series being processed and about the type of seasonal adjustment requested (if any);
(4) a *model diagnostics file* containing information about the regARIMA model fit to the series (if any).

The graphics metafile carries the extension **.gmt**, the seasonal adjustment diagnostics file carries the extension **.xdg**, and (if generated) the model diagnostics file carries the extension **.mdg**; these files carry the filename used for the main program output. For example, if a user enters

```
x12a test -g c:\sagraph
```

the graphics metafile will be stored in `c:\sagraph\test.gmt`, the seasonal adjustment diagnostics file will be stored in `c:\sagraph\test.xdg`, and the model diagnostics file will be stored in `c:\sagraph\test.mdg`. For

```
x12a test -g c:\sagraph -o testout
```

the grapics metafile will be stored in `c:\sagraph\testout.gmt`, the seasonal adjustment diagnostics file will be stored in `c:\sagraph\testout.xdg`, and the model diagnostics file will be stored in `c:\sagraph\testout.mdg`. In both cases, related files needed to generate seasonal adjustment graphics will be also be stored in the `c:\sagraph` subdirectory. (NOTE: The directory entered after the **-g** flag must already have been created and should be different from the directory used for the output files; it can be a subdirectory of the latter.)

Two versions of a program named X-12-Graph that use SAS/GRAPH (see SAS Institute (1990)) to produce graphs from the graphics mode output is distributed with X-12-ARIMA (see Hood (1998a) and Hood (1998b)). For examples of the use of X-12-Graph, see Findley and Hood (1999). For a list of the files stored by X-12-ARIMA in graphics mode, along with the codes used in the graphics metafile to denote these files, see Table 2-2 below.

The seasonal adjustment diagnostics file produced using the **-g** option stores only essential information about the seasonal adjustment run needed for the SAS/GRAPH external graphics procedure, and is a partial version of the file stored when the **-s** option is invoked. If both options are used in the same `X-12-ARIMA` run, the complete version of the seasonal adjustment diagnostics file will be stored in the directory specified by the **-g** option (and not in the directory of the main output file). If a model diagnostics file is also generated, that file will be stored in the the graphics directory as well. A warning message is written to the screen and to the log file telling the user that the seasonal adjusment diagnostics file (and the model diagnostics file, if it is produced) is in the graphics directory.

The **-n**, **-w**, **-p**, and **-r** flags all control the main output of the program. The **-n** option allows the user to restrict the number of tables appearing in the main output

Table 2-2 : Codes Associated With the `X-12-ARIMA` Graphics Metafile

Code	Description
acf	ACF
acf2	ACF of Squared Residuals
adjori	Prior-Adjusted Original Series
aichst	History of the AICCs
ao	Additive Outliers
caf	Combined Adjustment Factors
cal	Calendar Factors From Irregular Regression
ccal	Combined Calendar Factors From Irregular Regression
cfchst	History of Concurrent Forecasts and Forecast Errors
chol	Combined Holiday Factors
cmpori	Original Composite Series
csahst	History of the Percent Change in the Seasonally Adjusted Series
ctd	Combined Trading Day Factors From Irregular Regression
ctrhst	History of the Percent Change in the Trend Values
fct	Original Series and Forecasts
fcthst	History of the Sum of Squared Forecast Errors
fincal	Combined Calendar Factors
fintst	Final Outlier t-test Statistics
ftr	Transformed Original Series and Forecasts
idacf	ACFs Generated by Identify Spec
idpacf	PACFs Generated by Identify Spec
imori	Original Data Modified for Extremes from Indirect Adjustment
imsa	Seasonally Adjusted Data Modified for Extremes from Indirect Adjustment
imirr	Irregular Component Modified for Extremes from Indirect Adjustment
indirr	Indirect Irregular
indrsi	Indirect Replacement Values for the SI Ratios
indsa	Indirect Seasonally Adjusted Series
indsar	Indirect Seasonally Adjusted with Rounding
indsat	Indirect Seasonally Adj with Forced Annual Totals
indsf	Indirect Seasonal Factors
indsi	Indirect SI Ratios
indtrn	Indirect Trend
irr	Final Irregular Component
irrwt	Irregular Weights
isahst	History of the Indirect Seasonal Adjustment Values
ls	Level Shift Outliers
mdlest	regARIMA Model Estimates
modori	Original Data Modified for Extremes
modsa	Seasonally Adjusted Data Modified for Extremes
modirr	Irregular Component Modified for Extremes
mvadj	Original Series with Missing Values Replaced
oadori	Outlier-Adjusted Original Series
odjcmp	Outlier Adjusted Composite Data
ori	Original Series
otl	Combined Outliers
pacf	PACF
prior	Prior Adjustment Factors
ptd	Prior Trading Day Factors
rgseas	User-Defined Seasonal Regression Factors
rhol	Holiday Factors from regARIMA model

Table 2-2 : Codes Associated With the X-12-ARIMA Graphics Metafile (Continued)

Code	Description
rsi	Replacement SI Ratios
rtd	Trading Day Factors from regARIMA model
sa	Seasonally Adjusted Series
sahst	History of the Seasonally Adjusted Series
sarnd	Seasonally Adjusted Series with Rounding
satot	Seasonally Adjusted Series with Forced Annual Totals
sf	Seasonal Factors
sfhst	History of the Seasonal Factor Values
sfr	Seasonal Factors with User-Defined Regression
si	SI Ratios
siox	SI Ratios, with Labels for Outliers and Extreme Values
spccmp	Spectrum of the Composite Series
spciir	Spectrum of the Indirect Modified Irregular
spcirr	Spectrum of the Modified Irregular
spcisa	Spectrum of the Indirect Seasonally Adjusted Series
spcori	Spectrum of the Original Series
spcrsd	Spectrum of the regARIMA Model Residuals
spcsa	Spectrum of the Seasonally Adjusted Series
tc	Temporary Change Outliers
trn	Final Trend-Cycle Component
trnhst	History of the Trend Values
usrdef	User-Defined Regression Factors
xeastr	Easter Factors
xhol	Holiday Factors From Irregular Regression
xtd	Trading Day Factors From Irregular Regression
xtrm	Final Extreme Value Adjustment Factors

file. The `X-12-ARIMA` program produces a large number of tables in the main output file. While `X-12-ARIMA` is flexible in allowing users to determine which tables are to be printed out, it is sometimes convenient to restrict the output to only a few tables. To facilitate this, the **-n** flag specifies that, as the default, no tables will be written to the main output file. Then only those tables specified by the user in the spec file are written.

The **-w** flag specifies that a wide (132 character) format is used to print out tables in the main output file. The default is an 80 character tabular format. The exact format of the output tables is determined by the magnitude of the series values and by what degree of precision is requested in the **series** spec.

The **-p** flag specifies that page breaks and headers will be suppressed in the main output file. If this option is not specified, then page breaks will be inserted at the beginning of each table of output, along with a title for the run, series name, and page number.

The **-r** flag specifies that output tables and headers will be written in a format that will reduce the amount of output printed out to the main output file. The tables printed out are consolidated, and some blank lines in the printout are suppressed. This output option was first utilized in the version of X-12-ARIMA developed for use with the GiveWin econometrics package (see Doornik and Hendry 2001).

The **-q** flag specifies that X-12-ARIMA will be run in "quiet mode". Warning messages that are normally printed to the console are suppressed, although error messages shall still be printed to the console. All warning messages not printed to the screen will be stored in the error file (see Section 2.3).

The **-c** flag is used only to restrict a composite seasonal adjustment run done with an input metafile (**-m**). In a composite seasonal adjustment, `X-12-ARIMA` usually seasonally adjusts a set of component time series, as well as their composite (also called aggregate), which is usually their sum (for more details, see the description of the **composite** spec in Section 6). An input specification file is needed for each 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.

Finally, the **-v** flag specifies that `X-12-ARIMA` will be run in an input verification mode to enable the user to see if there are errors in one or more input spec files. This allows the user to check the program options for errors without doing the complete `X-12-ARIMA` runs for all the series. The **-v** flag cannot be used with the **-s**, **-c**, **-n**, **-w**, or **-p** flags.

2.8 Program limits

`X-12-ARIMA` contains limits on the maximum length of series, maximum number of regression variables in a model, etc. These limits are set at values believed to be sufficiently large for the great majority of applications, without being so large as to cause memory problems or to significantly lengthen program execution times.

Table 2-3 details those parameter variables in the `model.prm` and `srslen.prm` files corresponding to `X-12-ARIMA` program limits that are subject to user modification.

Table 2-3: X-12-ARIMA Program Limits

parameter variable	*value (limit)*	*description of parameter*
`pobs`	600	maximum length of the series on input. The number, `pobs` + `pfcst` (see below), is the maximum length of input series of user-defined regression variables and user-defined prior adjustment factors—the additional `pfcst` values are allowed to accomodate values of regression variables or adjustment factors in a possible forecast period
`pyrs`	60	maximum number of years in the forecast and backcast extended series
`pfcst`	60	maximum number of forecasts
`pb`	80	maximum number of regression variables in a model (including predefined and user-defined regression variables specified, plus any regression variables generated by automatic outlier detection or an AIC test)
`pureg`	52	maximum number of user-defined regression variables
`porder`	36	maximum lag corresponding to any AR or MA parameter
`pdflg`	3	maximum number of differences in any ARIMA factor (nonseasonal or seasonal)

The limits may be modified if required, but the FORTRAN source code of the program must then be recompiled and relinked to put the new limits into effect. The limits potentially requiring modification for this purpose occur in parameter statements in the files `model.prm` and `srslen.prm`. We suggest keeping a backup copy of the original files, in case problems arise from an attempt to modify program limits.

3. RegARIMA Modeling Capabilities of X-12-ARIMA

Section 3.1 describes the general model handled by the `X-12-ARIMA` program. Sections 3.2 to 3.7 give summary descriptions of the capabilities of `X-12-ARIMA` for the various stages of regARIMA modelling and forecasting: data input and transformation, regression variable specification, ARIMA model identification and specification, model estimation and inference, diagnostic checking including outlier detection, and forecasting. These sections also mention which input specification statements (specs) are used to control the execution of the capabilities discussed. Detailed documentation of the specs is given in Section 6.

When building a regARIMA model, it is strongly recommended that one examine a high resolution plot of the time series. Such a plot gives valuable information about seasonal patterns, potential outliers, stochastic nonstationarity, etc. Additional plots may also be useful for examining the effects of possible transformations on the series, or of applying various differencing operators to the series. Since `X-12-ARIMA` does not possess such plotting capabilities, other software must be used for this purpose.

3.1 General model

ARIMA models, as discussed by Box and Jenkins (1976), are frequently used for seasonal time series. A general multiplicative seasonal ARIMA model for a time series z_t can be written

$$\phi(B)\Phi(B^s)(1-B)^d(1-B^s)^D z_t = \theta(B)\Theta(B^s)a_t \tag{1}$$

where B is the backshift operator ($Bz_t = z_{t-1}$), s is the seasonal period, $\phi(B) = (1 - \phi_1 B - \cdots - \phi_p B^p)$ is the nonseasonal autoregressive (AR) operator, $\Phi(B^s) = (1 - \Phi_1 B^s - \cdots - \Phi_P B^{Ps})$ is the seasonal AR operator, $\theta(B) = (1 - \theta_1 B - \cdots - \theta_q B^q)$ is the nonseasonal moving average (MA) operator, $\Theta(B^s) = (1 - \Theta_1 B^s - \cdots - \Theta_Q B^{Qs})$ is the seasonal MA operator, and the a_ts are i.i.d. with mean zero and variance σ^2 (white noise). The $(1-B)^d(1-B^s)^D$ implies nonseasonal differencing of order d and seasonal differencing of order D. If $d = D = 0$ (no differencing), it is common to replace z_t in (1) by deviations from its mean, that is, by $z_t - \mu$ where $\mu = E[z_t]$.

A useful extension of ARIMA models results from the use of a time-varying mean function modelled via linear regression effects. More explicitly, suppose we write a linear regression equation for a time series y_t as

$$y_t = \sum_i \beta_i x_{it} + z_t \tag{2}$$

where y_t is the (dependent) time series, the x_{it} are regression variables observed concurrently with y_t, the β_i are regression parameters, and $z_t = y_t - \sum \beta_i x_{it}$, the time series of regression errors, is assumed to follow the ARIMA model (1). Modelling z_t as ARIMA addresses the fundamental problem with applying standard regression methodology to time series data, which is that standard regression assumes that the regression errors (z_t in (2)) are uncorrelated over time. In fact, for time series data, the errors in (2) will usually be autocorrelated, and, moreover, will often require differencing. Assuming z_t is uncorrelated in such cases will typically lead to grossly invalid results.

The expressions (1) and (2) taken together define the general regARIMA model allowed by the `X-12-ARIMA` program. Combining (1) and (2), the model can be written in a single equation as

$$\phi(B)\Phi(B^s)(1-B)^d(1-B^s)^D\Big(y_t - \sum_i \beta_i x_{it}\Big) = \theta(B)\Theta(B^s)a_t. \tag{3}$$

The regARIMA model (3) can be thought of either as generalizing the pure ARIMA model (1) to allow for a regression mean function ($\sum \beta_i x_{it}$), or as generalizing the regression model (2) to allow the errors z_t to follow the ARIMA model (1). In any case, notice that the regARIMA model implies that first the regression effects are subtracted from y_t to get the zero mean series z_t, then the error series z_t is differenced to get a stationary series, say w_t, and w_t is then assumed to follow the stationary ARMA model, $\phi(B)\Phi(B^s)w_t = \theta(B)\Theta(B^s)a_t$. Another way to write the regARIMA model (3) is

$$(1-B)^d(1-B^s)^D y_t = \sum_i \beta_i (1-B)^d(1-B^s)^D x_{it} + w_t. \tag{4}$$

where w_t follows the stationary ARMA model just given. Equation (4) emphasizes that the regression variables x_{it} in the regARIMA model, as well as the series y_t, are differenced by the ARIMA model differencing operator $(1-B)^d(1-B^s)^D$.

Notice that the regARIMA model as written in (3) assumes that the regression variables x_{it} affect the dependent series y_t only at concurrent time points, i.e., model (3) does not explicitly provide for lagged regression effects such as $\beta x_{i,t-1}$. Lagged effects can be included by the `X-12-ARIMA` program, however, by reading in appropriate user-defined lagged regression variables.

The `X-12-ARIMA` program provides additional flexibility in the specification of the ARIMA part of a regARIMA model by permitting (*i*) more than two multiplicative ARIMA factors, (*ii*) missing lags within the AR and MA polynomials, (*iii*) the fixing of individual AR and MA parameters at user-specified values when the model is estimated, and (*iv*) inclusion of a *trend constant*, which is a nonzero overall mean for the differenced series ($(1-B)^d(1-B^s)^D y_t$). These features of regARIMA model specification are discussed and illustrated in Section 6.

Detailed discussions of ARIMA modelling are given in the classic book by Box and Jenkins (1976), and also in several other time series texts, such as Abraham and Ledolter (1983) and Vandaele (1983). Bell (1999) discusses regARIMA modelling in particular, with the discussion tailored to use of the `X-12-ARIMA` program.

3.2 Data input and transformation

Observations of the original time series to be analyzed are read into the program with the **series** spec. The data may either be included in the **series** spec or read from a file. The **span** and **modelspan** arguments of the **series** spec are used to restrict the analysis to a span of the data, omitting data from the beginning and/or end of the original time series. The **series** spec is also used to specify the starting date, seasonal period (if appropriate), and title for the time series.

The **transform** spec provides nonlinear transformations of the data, as well as modification by prior-adjustment factors. The nonlinear transformations included are the *Box-Cox* (1964) family of power transformations (such as the logarithm or square root), and the *logistic* transformation (useful for a time series of proportions greater than 0 and less than 1). A predefined prior adjustment may be specified that divides each observation in a monthly series by the corresponding *length of month* (or *length of quarter* for quarterly series) and then rescales it by the average length of month (or quarter). Similarly, leap year adjustment factors for February are also available. Finally, a set of user-defined prior-adjustments may be supplied for division into or subtraction from the original time series. The result of the **series** and **transform** specs is the time series $y_t,\ t = 1, \ldots, n$, used in the regARIMA model (3).

3.3 Regression variable specification

Specification of a regARIMA model requires specification of both the regression variables (the x_{it}'s in (2)) and the ARIMA model (1) for the regression errors z_t. The former is done using the **regression** spec, and the latter using the **arima** spec (discussed in Section 3.4). Choosing which regression variables to include requires user knowledge relevant to the time series being modelled. Several regression variables that are frequently used in modelling seasonal economic time series are built into the `X-12-ARIMA` program, and can be easily included in the model. These are discussed below, and the actual regression variables used are given in Table 3-1 in this section. Specification and use of these variables is described in the documentation of the **regression** spec in Section 6. In addition, users may input data for any other regression variables (called user-defined regression variables) that they wish to include in a model. As part of model estimation (see Section 3.5), `X-12-ARIMA` provides standard t-statistics to assess the statistical significance of individual regression parameters, and χ^2-statistics to assess the significance of groups of regression parameters corresponding to particular effects (such as trading-day effects).

The most basic regression variable is the *constant term.* If the ARIMA model does not involve differencing, this is the usual regression intercept, which, if there are no other regression variables in the model, represents the mean of the (stationary) series. If the ARIMA model does involve differencing, `X-12-ARIMA` uses a regression variable such that, when it is differenced according to the ARIMA model (see equation (4)), a column of ones is produced. The corresponding parameter is then called a *trend constant*, since it provides for a polynomial trend of the same degree as the number of differences in the model. For example, with nonseasonal differencing ($d > 0$) but no seasonal differencing ($D = 0$), the (undifferenced) trend constant regression variable is proportional to t^d. Notice that the lower order polynomial terms, t^j for $0 \leq j < d$, are not included among the regression variables because they would be differenced to zero by $(1-B)^d$, hence their coefficients cannot be estimated. With or without the trend constant, the model (3) (or (4)) implicitly allows for these lower order polynomial terms through the differencing. If seasonal differencing is requested ($D > 0$), the nature of the undifferenced trend constant regression variable is more complicated, though the trend constant can be thought of as allowing for a polynomial of degree $d + D$. Without a trend constant, model (3) implicitly allows for a polynomial of degree $d + D - 1$.

Fixed seasonal effects in a monthly series can be modelled using 12 indicator variables, one for each calendar month. Since these 12 variables always add to one, however, they are confounded with an overall level effect. This leads to one of two singularity problems: collinearity with the usual constant term in a model with no differencing; or a singularity in a model with differencing since the 12 variables, when differenced, always sum to 0. One appropriate reparameterization instead uses 11 contrasts in the 12 indicator variables. An alternative reparameterization uses 11 variables taken from the Fourier (trigonometric) series representation of a fixed monthly pattern. The variables used for both of these parameterizations are given in Table 3-1. `X-12-ARIMA` allows either of these options, and also allows specifying the trigonometric terms only for selected frequencies. For quarterly series, or for series with other seasonal periods, `X-12-ARIMA` constructs the appropriate versions of these variables. Notice that these variables cannot be used in a model with seasonal differencing, as they would all be differenced to zero.

Trading-day effects occur when a series is affected by the differing day-of-the-week compositions of the same calendar month in different years. Trading-day effects can be modelled with 7 variables that represent (*no. of Mondays*), ..., (*no. of Sundays*) in month t. Bell and Hillmer (1983) note, however, that a better parameterization of the same effects instead uses 6 contrast variables defined as (*no. of Mondays*) − (*no. of Sundays*), ..., (*no. of Saturdays*) − (*no. of Sundays*), along with a seventh variable for *length of month* (`lom`) or its deseasonalized version, the leap-year regressor (`lpyear`). In `X-12-ARIMA` the 6 contrast variables are called the `tdnolpyear` variables. Instead of using a seventh regressor, a simpler and often better way to handle multiplicative leap-year effects is to rescale the February values Y_t of the original time series before transformation to $\bar{m}_{Feb} Y_t / m_t$, where Y_t is the original time series before transformation, m_t is the length of month t (28 or 29), and $\bar{m}_{Feb} = 28.25$ is the average length of February. (If the regARIMA model includes seasonal effects, these can account for the length of month effect except in Februaries, so the trading day model only has to deal with the leap year effect.) When this is done, only the `tdnolpyear` variables need be included in the model. `X-12-ARIMA` allows explicit choice of either approach, as well as an option (`td`) that makes a default choice of how to handle length-of-month effects—see the documentation of the **regression** spec.

The preceeding paragraph assumes the time series being modelled represents the aggregation of some daily series (typically unobserved) over calendar months. Such series are called monthly *flow* series. If the series instead represents the value of some daily series at the end of the month, called a monthly *stock* series, then different regression variables are appropriate. Trading-day effects in end-of-month stock series can be modelled using 7 indicator variables for the day-of-the-week that the months end on. Since the sum of these variables is always one, this leads to a singularity problem, so 6 contrast variables are used instead. (See Table 3-1.) `X-12-ARIMA` also allows specification of regression variables appropriate for stock series defined as of some other day of the month, e.g., for beginning of the month stock series.

For quarterly flow time series, `X-12-ARIMA` allows the same trading-day options as in the monthly case. Trading-day effects in quarterly series are relatively rare, however, because the calendar composition of quarters does not vary as much over time, on a percentage basis, as that of months does. Trading-day variables are not provided for flow time series with seasonal periods other than monthly or quarterly, or for stock series other than monthly.

`X-12-ARIMA` also provides a simplified model for trading day variation of monthly or quarterly flow series that uses only one regressor, a weekday-weekend contrast variable:

$$T_t = (no.\ of\ Weekdays) - \frac{5}{2}(no.\ of\ Saturdays\ and\ Sundays)$$

The underlying assumption for this model is that all weekdays (Monday through Friday) have identical effects, and Saturday and Sunday have identical effects. In `X-12-ARIMA` this model can be estimated in two ways: by specifying the `td1coef` option if the user wishes the program to make the choice of how to handle length of month effects as with the `td` option mentioned above, or by specifying the `td1nolpyear` option in which case the length of month effects model must be specified by the user, as with `tdnolpyear`.

Holiday effects (in a monthly flow series) arise from holidays whose dates vary over time if (i) the activity measured by the series regularly increases or decreases around the date of the holiday, and (ii) this differentially affects two (or more) months depending on the date the holiday occurs each year. (Effects of holidays with a fixed date, such as Christmas, are indistinguishable from fixed seasonal effects.) *Easter effects* are the most frequently found holiday effects in U.S. economic time series, since the date of Easter varies between March 22 and April 25. *Labor Day* and *Thanksgiving* also are potential, though less common, sources of holiday effects. The basic model used by X-12-ARIMA for Easter and Labor Day effects assumes that the level of activity changes on the $w-$th day before the holiday for a specified w, and remains at the new level until the day before the holiday. For Thanksgiving the model used assumes that the level of activity changes on the day that is a specified number of days before or after Thanksgiving and remains at the new level until December 24. The regression variable constructed for the holiday effect is, for a given month t, the proportion of the affected time period that falls in month t. (Actually, as noted in Table 3-1, these regressors are deseasonalized by subtracting off their long-run monthly means.) Essentially the same Easter effect variable applies also to quarterly flow time series, but Labor Day and Thanksgiving effects are not present in quarterly series. `X-12-ARIMA` does not provide built-in variables for possible holiday effects in stock series.

`X-12-ARIMA` provides four other types of regression variables to deal with abrupt changes in the level of a series of a temporary or permanent nature: *additive outliers* (AOs), *level shifts* (LSs), *temporary changes* (TCs), and *ramps*. AOs affect only one observation in the time series, LSs increase or decrease all observations from a certain time

Table 3-1: Predefined Regression Variables in X-12-ARIMA

[1]*Regression effect*	*Variable definition(s)*
Trend Constant `const`	$(1-B)^{-d}(1-B^s)^{-D}I(t \geq 1)$, where $I(t \geq 1) = \begin{cases} 1 & \text{for } t \geq 1 \\ 0 & \text{for } t < 1 \end{cases}$
[2]**Fixed Seasonal** `seasonal`	$M_{1,t} = \begin{cases} 1 & \text{in January} \\ -1 & \text{in December} \\ 0 & \text{otherwise} \end{cases}, \ldots, \; M_{11,t} = \begin{cases} 1 & \text{in November} \\ -1 & \text{in December} \\ 0 & \text{otherwise} \end{cases}$
[2]**Fixed Seasonal** `sincos[ ]`	$\sin(\omega_j t)$, $\cos(\omega_j t)$, where $\omega_j = 2\pi j/12$, $1 \leq j \leq 6$ (Drop $\sin(\omega_6 t) \equiv 0$)
Trading Day (monthly or quarterly flow) `tdnolpyear`, [3]`td`	$T_{1,t}$ = (no. of Mondays) − (no. of Sundays), …, $T_{6,t}$ = (no. of Saturdays) − (no. of Sundays)
One Coefficient Trading Day (monthly or quarterly flow) `td1nolpyear`, [4]`td1coef`	(no. of weekdays) − $\frac{5}{2}$(no. of Saturdays and Sundays)
Length-of-Month (monthly flow) `lom`	$m_t - \bar{m}$, where m_t = length of month t (in days) and $\bar{m} = 30.4375$ (average length of month)
Length-of-Quarter (quarterly flow) `loq`	$q_t - \bar{q}$, where q_t = length of quarter t (in days) and $\bar{q} = 91.3125$ (average length of quarter)
Leap Year (monthly and quarterly flow) `lpyear`	$LY_t = \begin{cases} 0.75 & \text{in leap year February (first quarter)} \\ -0.25 & \text{in other Februaries (first quarter)} \\ 0 & \text{otherwise} \end{cases}$
Stock Trading Day (monthly stock) `tdstock[`w`]`	$D_{1,t} = \begin{cases} 1 & \tilde{w}^{\text{th}} \text{ day of month } t \text{ is a Monday} \\ -1 & \tilde{w}^{\text{th}} \text{ day of month } t \text{ is a Sunday} \\ 0 & \text{otherwise} \end{cases}, \ldots, \; D_{6,t} = \begin{cases} 1 & \tilde{w}^{\text{th}} \text{ day of month } t \text{ is a Saturday} \\ -1 & \tilde{w}^{\text{th}} \text{ day of month } t \text{ is a Sunday} \\ 0 & \text{otherwise} \end{cases},$ where $\tilde{w}$ is the smaller of w and the length of month t. For end-of-month stock series, set w to 31, i.e., specify `tdstock[31]`.
Statistics Canada Easter (monthly or quarterly flow) `sceaster[`w`]`	If Easter falls before April w, let n_E be the number of the w days on or before Easter falling in March. Then: $E(w,t) = \begin{cases} n_E/w & \text{in March} \\ -n_E/w & \text{in April} \\ 0 & \text{otherwise} \end{cases}.$ If Easter falls on or after April w, then $E(w,t) = 0$. (Note: This variable is 0 except in March and April (or first and second quarter).)

[1]Restrictions, if any, are given in parentheses. Each entry also gives the name used to specify the regression effect in the **variables** argument of the **regression** spec, e.g., `regression {variables=const}`.

[2]The variables shown are for monthly series. Corresponding variables are available for any other seasonal period.

[3]In addition to these 6 variables, the `td` option also includes the `lpyear` regression variable (for untransformed series), or it rescales February values of Y_t to $\bar{m}_{Feb}Y_t/m_t$, where $\bar{m}_{Feb} = 28.25$ (average length of February) (for an original series Y_t that is transformed). Quarterly `td` is handled analogously.

[4]In addition to this variable, the `td1coef` option also includes the `lpyear` regression variable (for untransformed series), or it rescales February values of Y_t to $\bar{m}_{Feb}Y_t/m_t$, where $\bar{m}_{Feb} = 28.25$ (average length of February) (for an original series Y_t that is transformed). Quarterly `td1coef` is handled analogously.

Table 3-1: Predefined Regression Variables in `X-12-ARIMA`

[1]*Regression effect*	*Variable definition(s)*
[5]**Easter Holiday** (monthly or quarterly flow) `easter[`w`]`	$E(w,t) = \frac{1}{w} \times$[no. of the w days before Easter falling in month (or quarter) t]. (Note: This variable is 0 except in February, March, and April (or first and second quarter). It is nonzero in February only for $w > 22$.)
[5]**Labor Day** (monthly flow) `labor[`w`]`	$L(w,t) = \frac{1}{w} \times$[no. of the w days before Labor Day falling in month t]. (Note: This variable is 0 except in August and September.)
[5]**Thanksgiving** (monthly flow) `thank[`w`]`	$ThC(w,t)$ = proportion of days from w days before Thanksgiving through December 24 that fall in month t (negative values of w indicate days after Thanksgiving). (Note: This variable is 0 except in November and December.)
Additive Outlier at t_0 `ao`$date_0$	$AO_t^{(t_0)} = \begin{cases} 1 & \text{for } t = t_0 \\ 0 & \text{for } t \neq t_0 \end{cases}$ ($date_0$ is the date corresponding to time point t_0)
Level Shift at t_0 `ls`$date_0$	$LS_t^{(t_0)} = \begin{cases} -1 & \text{for } t < t_0 \\ 0 & \text{for } t \geq t_0 \end{cases}$
Temporary Change at t_0 `tc`$date_0$	$TC_t^{(t_0)} = \begin{cases} 0 & \text{for } t < t_0 \\ \alpha^{t-t_0} & \text{for } t \geq t_0 \end{cases}$, where α is the rate of decay back to the previous level ($0 < \alpha < 1$).
Ramp, t_0 **to** t_1 `rp`$date_0$`-`$date_1$	$RP_t^{(t_0,t_1)} = \begin{cases} -1 & \text{for } t \leq t_0 \\ (t-t_0)/(t_1-t_0) - 1 & \text{for } t_0 < t < t_1 \\ 0 & \text{for } t \geq t_1 \end{cases}$

[4]Restrictions, if any, are given in parentheses. Each entry also gives the name used to specify the regression effect in the **variables** argument of the **regression** spec, e.g., `regression {variables=const}`.

[5]The actual variable used for monthly Easter effects is $E(w,t) - \bar{E}(w,t)$, where the $\bar{E}(w,t)$ are the "long-run" monthly means of $E(w,t)$ corresponding to the first 400 year period of the Gregorian calendar, 1583-1982. This provides a close approximation to the average calculated over the much longer period of a complete cycle of the dates of Easter. For more details, see Montes (1998). (These means are nonzero only for February, March, and April). Analogous deseasonalized variables are used for Labor Day and Thanksgiving effects, and for quarterly Easter effects.

point onward by some constant amount, TCs allow for an abrupt increase or decrease in the level of the series that returns to its previous level exponentially rapidly, and ramps allow for a linear increase or decrease in the level of the series over a specified time interval. The specific regression variables used to model these effects are given in Table 3-1. (LS regression variables are defined as -1 and then 0, in preference to an equivalent 0 and then 1 definition, to make the overall level of the regression mean function of any forecasts consistent with the most recent level of the time series. Similar considerations dictate the definition of ramp variables.)

The **regression** spec allows specification of AOs, LSs, TCs, and ramps for cases where prior knowledge suggests such effects at known time points. Often, however, large seasonal movements make it difficult to identify where such changes in level have occured. Determination of the location and nature of potential outliers is the objective of the outlier detection methodology implemented by the **outlier** spec—see Section 3.6 and the **outlier** spec documentation in Section 6. This methodology can be used to detect AOs, TCs, and LSs (not ramps); any that are detected are automatically added to the model as regression variables.

Prespecified AOs, LSs, TCs, and ramps are actually simple forms of *interventions* as discussed by Box and Tiao (1975). While `X-12-ARIMA` does not provide the full range of dynamic intervention effects discussed by Box and Tiao, often a short sequence of suitably chosen AO, LS, TC, and/or ramp variables can produce reasonable approximations to more complex dynamic intervention effects, albeit at the cost of an additional parameter or two. Analogous remarks apply to the relation between regARIMA models containing (user-defined) regression variables that are themselves stochastic time series, and the dynamic transfer function models discussed by Box and Jenkins (1976, chapters 10 and 11). Thus, regARIMA models can often be used to approximate more general dynamic transfer function models, although transfer function models require special treatment when forecasting, since future values of stochastic explanatory variables are generally unknown. (See Box and Jenkins 1976, Section 11.5.)

3.4 Identification and specification of the ARIMA part of the model

The ARIMA part of a regARIMA model is determined by its orders and structure, e.g., $(p\ d\ q)$, $(P\ D\ Q)$, and s for model (1). If no regression variables are included in the model, then determination of the orders for the resulting pure ARIMA model (called ARIMA model identification) can be carried out by following well-established procedures that rely on examination of the sample autocorrelation function (ACF) and sample partial autocorrelation function (PACF) of the time series y_t and its differences. For regARIMA models, a modified approach is needed, since the presence of regression effects can distort the appearance of the ACF and PACF. Typically, the differencing orders can be identified by examining ACFs of the time series y_t and its differences. Then, one should obtain the residuals from a regression of the differenced data on the differenced regression variables. The ACF and PACF of these residuals can then be examined in the usual way to identify the AR and MA orders of the regression error term in the regARIMA model. This approach to regARIMA model identification is discussed and illustrated in Bell and Hillmer (1983) and Bell (1999).

The key spec used to implement this approach to regARIMA model identification is the **identify** spec. For illustration, consider a monthly seasonal time series. The usual ACFs examined to determine the differencing needed are those of y_t, $(1-B)y_t$, $(1-B^{12})y_t$, and $(1-B)(1-B^{12})y_t$. The **identify** spec can produce all these ACFs in a single run. Once the differencing has been determined, another run of `X-12-ARIMA` can be made using the **identify** and **regression** specs together to (i) regress the differenced y_t series on the differenced regression variables, and (ii) produce the ACF and PACF of the regression residuals for use in identifying the AR and MA orders of the model. For example, if one nonseasonal and one seasonal difference are specified ($d=1$ and $D=1$), the **identify** and **regression** specs will fit the model

$$(1-B)(1-B^{12})y_t = \sum_i \beta_i(1-B)(1-B^{12})x_{it} + w_t \tag{5}$$

by ordinary least squares (OLS), and will produce the ACF and PACF of the regression residuals w_t in (5).

An alternative approach that does not require two runs of the `X-12-ARIMA` program can be used if the maximum differencing orders (nonseasonal and seasonal) that may be required are assumed known. For example, suppose that these maximum differencing orders are $d = 1$ and $D = 1$. Then the **identify** and **regression** specs can be used to (*i*) perform OLS regression on (5) to produce parameter estimates $\tilde{\beta}_i$, (*ii*) compute an estimated (undifferenced) regression error series $\tilde{z}_t = y_t - \sum_i \tilde{\beta}_i x_{it}$, and (*iii*) produce ACFs and PACFs of $\tilde{z}_t, (1-B)\tilde{z}_t$, $(1-B^{12})\tilde{z}_t$, and $(1-B)(1-B^{12})\tilde{z}_t$. These ACFs and PACFs can be examined to determine the orders of differencing required, as well as the orders of the AR and MA operators for the model.

There is one exception to the above remarks. If a constant term is specified in the **regression** spec, then it will be included when the OLS regression is done on (5), but not when the regression effects are removed from the data. Thus, actually, $\tilde{z}_t = y_t - \Sigma_{i \geq 2} \tilde{\beta}_i x_{it}$ if $\tilde{\beta}_1 x_{1t}$ is the trend constant term. To explain why this is done, we consider (5). From remarks in Section 3.3, a trend constant variable in model (5) allows for a polynomial of degree 2, though the constant and linear terms (for $t^0 \equiv 1$ and t) are implicitly allowed for through the differencing by $(1-B)(1-B^{12})$. Since the constant and linear coefficients cannot be estimated, the full polynomial effect cannot be subtracted from the undifferenced series y_t. Rather than subtract only the t^2 term of the polynomial, `X-12-ARIMA` ignores the estimated trend constant when creating the undifferenced regression error series $\tilde{z}_t$. Similar remarks apply to the general model (4). The only effect that inclusion of a trend constant has on the computations of the **identify** spec is that its inclusion in (4) will affect the regression estimates $\tilde{\beta}_i$ for $i \geq 2$.

3.5 Model estimation and inference

The **regression** and **arima** specs specify a regARIMA model. The **estimate** spec then estimates the model parameters by exact maximum likelihood, or by a variant known as conditional maximum likelihood (Box, Jenkins and Riensel 1976, pp. 209–212), which is sometimes called conditional least squares. Users may specify maximization of the fully exact likelihood, or of the likelihood conditional for the AR but exact for the MA parameters, or of the likelihood conditional for both the AR and MA parameters. Differences in AR parameter estimation between exact and conditional likelihood maximization are generally small, and there are situations where each approach is appropriate. (See Section 4.) Differences between exact and conditional likelihood for MA parameter estimation are more fundamental, with exact likelihood being the recommended approach. The option of choosing the conditional likelihood for MA parameters is provided in `X-12-ARIMA` mainly for comparison of results with other software, and for occasional use to produce initial estimates for exact maximum likelihood estimation when convergence problems arise. (See Section 4.1.) The default option is exact maximum likelihood estimation for both the AR and MA parameters.

Whichever choice of estimation method is made, the resulting log-likelihood for a pure ARIMA model is reduced to a sum of squares function that is then minimized by a nonlinear least squares routine (MINPACK, discussed by More, Garbow, and Hillstrom 1980). To maximize the likelihood for a full regARIMA model, an iterative generalized least squares (IGLS) algorithm (Otto, Bell and Burman 1987a, 1987b) is used. This algorithm involves two general steps: (*i*) for given values of the AR and MA parameters, the regression parameters that maximize the likelihood are obtained by a generalized least squares (GLS) regression (using the covariance structure of the regression errors, which is determined by their ARIMA model); and (*ii*) for given values, β_i, of the regression parameters, the ARIMA model is fit by maximum likelihood to the time series of regression errors, $z_t = y_t - \sum \beta_i x_{it}$. IGLS iterates between these two general steps until convergence is achieved. (Output options in the **estimate** spec allow for display of intermediate results from the estimation iterations, if desired.) The likelihood function (exact or conditional) is evaluated using an approach derived from those suggested by Box and Jenkins (1976, Chapter 7), Ljung and Box (1979), Hillmer and Tiao (1979), and Wilson (1983). Section 4 discusses certain problems that may arise in model estimation that all users should be aware of.

Statistical inferences about regARIMA model parameters may be made using asymptotic results for maximum likelihood estimation of ARIMA models (Box and Jenkins 1976, chapter 7; Brockwell and Davis 1987, chapter 8) and regARIMA models (Pierce 1971; see also Bell 1999). These results state that, under suitable assumptions, the parameter estimates are approximately normally distributed with means equal to the true parameter values and with a certain covariance matrix that can be estimated. (The "suitable assumptions" include that the true model form is used, that the model's AR operators are all stationary and its MA operators are all invertible, and that the series is sufficiently long for the asymptotic results to apply.) Using these results, `X-12-ARIMA` provides standard errors for the ARMA and regression parameter estimates, and, optionally, correlation (or covariance) matrices for the estimates of both the ARMA and the regression parameters. (The regression parameter estimates are asymptotically uncorrelated with the ARMA parameter estimates.) These results may be used in the usual way to make normal theory inferences about model parameters, including, as mentioned in Section 3.3, use of t-statistics and χ^2-statistics produced by `X-12-ARIMA` to assess the statistical significance of individual regression parameters and of groups of regression parameters corresponding to particular regression effects. Also, since `X-12-ARIMA` prints out the value of the maximized log-likelihood function, various likelihood ratio tests are possible by making multiple runs of the program with different models.

`X-12-ARIMA` uses the maximum likelihood estimate of the residual variance σ^2, which is $\hat{\sigma}^2 = \mathrm{SS}/(n - d - s \cdot D)$, where SS is the residual sum-of-squares and $n - d - s \cdot D$ is the effective number of observations after differencing. (If the likelihood function that is conditional with respect to the AR parameters is used, replace $n-d-s \cdot D$ by $n-p-d-s \cdot P - s \cdot D$.) Notice there is no "degrees of freedom" adjustment for model parameters being estimated. For this reason, if `X-12-ARIMA` is used to fit a pure regression model—a model whose regression errors follow the ARIMA(0 0 0) model—$\hat{\sigma}^2$ will differ from the usual unbiased regression variance estimate. Consequently, the resulting standard errors, t-statistics, and χ^2-statistics for the regression parameter estimates will also differ slightly from those that would be obtained from a standard regression program.

An alternative approach to inference is to use the likelihood-based model selection criteria produced by `X-12-ARIMA`: AIC, AICC (also known as the F-adjusted AIC), Hannan-Quinn, and BIC. For each of these statistics, the model producing the lower value is preferred. One advantage to these criteria over standard t-statistics, χ^2-statistics, and likelihood ratio tests is that they may be used to compare *nonnested* models—models that differ from each other in such a way that one model cannot be obtained simply by removing parameters from another model. (E.g., AR(1) versus MA(1) is a nonnested comparison.) Some caution must be exercised in use of the model selection criteria. Section 4.5 discusses certain situations that arise in regARIMA modelling for which the use of these criteria, as well as standard likelihood ratio tests, is invalid.

3.6 Diagnostic checking including outlier detection

Diagnostic checking of a regARIMA model is performed through various analyses of the residuals from model estimation, the objective being to check if the true residuals (a_t in (3)) appear to be white noise—i.i.d. $N(0, \sigma^2)$. (Note: Normality of the a_ts is not needed for the large sample estimation and inference results; it is most important for validity of prediction intervals produced in forecasting.) The **check** spec is used to produce various diagnostic statistics using the residuals from the fitted model. To check for autocorrelation, `X-12-ARIMA` can produce ACFs and PACFs of the residuals (with standard errors), along with Ljung and Box (1978) summary Q-statistics. `X-12-ARIMA` can also produce basic descriptive statistics of the residuals and a histogram of the standardized residuals. The residuals can be written to a file for further analysis (such as high resolution plotting) by other software.

An important aspect of diagnostic checking of time series models is outlier detection. The **outlier** spec of `X-12-ARIMA` provides for automatic detection of additive outliers (AOs), temporary change outliers (TCs) and level shifts (LSs). These outlier types (referring to AOs, TCs, and LSs as "outliers") and their associated regression variables are defined in Section 3.3. `X-12-ARIMA`'s approach to outlier detection is based on that of Chang and Tiao (1983)—see also Chang, Tiao, and Chen (1988)—with extensions and modifications as discussed in Bell (1983, 1999) and Otto and Bell (1990). The general approach is similar to stepwise (GLS) regression, where the candidate regression variables are AO, LS, and/or TC variables for all time points at which outlier detection is being performed—$3n$ variables for detection of AOs, LSs, and TCs over an entire time series of length n. (Actually, slightly fewer than $3n$ variables are used in this case for reasons discussed in the DETAILS section of the **outlier** spec documentation in Section 6.) In brief, this approach involves computing t-statistics for the significance of each outlier type at each time point, searching through these t-statistics for significant outlier(s), and adding the corresponding AO, LS, or TC regression variable(s) to the model. Overly burdensome computation is avoided by holding the AR and MA parameters fixed as the outlier t-statistics are computed for each time point and outlier type. `X-12-ARIMA` provides two variations on this general theme. The `addone` method provides full model reestimation after each single outlier is added to the model, while the `addall` method reestimates the model only after a set of detected outliers is added. A description of both these methods is given in the documentation of the **outlier** spec in Section 6, with more details in Appendix B of Findley, Monsell, Bell, Otto and Chen (1997).

During outlier detection a robust estimate of the residual standard deviation, $1.48 \times$ the median absolute deviation of the residuals (Hampel et al. 1986, p. 105), is used. Because outlier detection involves searching over all (or a specified set of) time points for the most significant outliers, the usual normal distribution critical values (e.g., 2.0) are too low for judging significance in outlier detection. The default critical value is determined by the number of observations in the interval searched for outliers (see Table 6-15), but this can be changed by the user.

When a model contains two or more level shifts, including those obtained from outlier detection as well as any level shifts specified in the **regression** spec, `X-12-ARIMA` will optionally produce t-statistics for testing null hypotheses that each run of two, three, etc. successive level shifts actually cancels to form a *temporary level shift*. Two successive level shifts cancel to form a temporary level shift if the effect of one offsets the effect of the other, which implies that the sum of the two corresponding regression parameters is zero. Similarly, three successive level shifts cancel to a temporary level shift if the sum of their three regression parameters is zero, etc. (There is a user-specified limit on the number of successive level shifts in the runs tested.) The t-statistics produced are the sums of the estimated parameters for each run of successive level shifts divided by the appropriate standard error. An insignificant temporary level shift t-statistic (say, one less than 2 in magnitude) fails to reject the null hypothesis that the corresponding level shifts cancel to form a temporary level shift. These tests are provided primarily as diagnostics to help users assess the impacts of level shifts in a model. Of course, if one or more of these t-statistics are significant, the user may wish to respecify the model with the relevant level shift regression variables replaced by appropriate temporary level shift variables. (These can be specified as user-defined regression variables, or can be obtained by fixing the coefficient of the level shift regressors so that they sum to zero.) The choice between using level shifts (which correspond to permanent changes in the level of a series) versus temporary level shifts could be important for forecasting a series with level shifts near the end of the data.

3.7 Forecasting

For a given regARIMA model with parameters estimated by the `X-12-ARIMA` program, the **forecast** spec will use the model to compute point forecasts, and associated forecast standard errors and prediction intervals. The point forecasts are minimum mean squared error (MMSE) linear predictions of future y_ts based on the present and past y_ts assuming that the true model is used—which means we assume the regARIMA model form is correct, that the correct regression variables have been included, that no additive outliers or level shifts will occur in the forecast period, that the specified ARIMA orders are correct, and that the parameter values used (typically estimated parameters) are equal to the true values. These are standard assumptions, though obviously unrealistic in practical applications. What is more realistically hoped is that the regARIMA model will be a close enough approximation to the true, unknown model for the results to be approximately valid. Two sets of forecast standard errors are produced. One assumes that all parameters are known. The other allows for additional forecast error that comes from estimating the regression parameters, while still assuming that the AR and MA parameters are known. For a reasonably long time series, Box and Jenkins (1976, pp. 267–269) observe that the contribution to forecast error of the error in estimating the AR and MA parameters is generally small, thus providing a justification for ignoring this source of error when computing the forecast standard errors.

If the series has been transformed, then forecasting results are first obtained in the transformed scale, and then transformed back to the original scale. For example, if one specifies a model of form (3) for $y_t = \log(Y_t)$, where Y_t is the original time series, then y_t is forecasted first, and the resulting point forecasts and prediction interval limits are exponentiated to produce point and interval forecasts in the original (Y_t) scale. The resulting point forecasts are MMSE for $y_t = \log(Y_t)$, but not for Y_t under the "standard" assumptions mentioned above. Analogous procedures are followed for other transformations allowed by `X-12-ARIMA`. If any prior adjustments are made, these will also be inverted in the process of transforming the point forecasts and prediction interval limits back to the original scale.

If there are any user-defined regression variables in the model, `X-12-ARIMA` requires that the user supply data for these variables for the forecast period. For the predefined regression variables in `X-12-ARIMA`, the program will generate the future values required. If user-defined prior adjustment factors are specified, values for these should also be supplied for the forecast period.

4. Points Related to regARIMA Model Estimation

While the IGLS algorithm and nonlinear least squares routine used by the `X-12-ARIMA` program are quite reliable at finding maximum likelihood estimates for regARIMA models, problems in estimation occasionally do occur. Some problems that can arise in model estimation are discussed below, along with possible solutions. This is followed by important cautions regarding the use of the model selection criteria produced by the `X-12-ARIMA` program.

4.1 Initial values for parameters and dealing with convergence problems

Users may supply initial values for AR and MA parameters that are then used to start the iterative likelihood maximization. This is rarely necessary, however, and is not generally recommended. The default choice of initial parameter values in `X-12-ARIMA` is 0.1 for all AR and MA parameters. (Initial values are not needed for the regression parameters, which are determined in the GLS regressions.) This default choice of initial values appears to be adequate in the great majority of cases. Supplying better initial values (as might be obtained, e.g., by first fitting the model using conditional likelihood) does not seem to speed up convergence enough to make obtaining the initial estimates generally worth the effort. A possible exception to this occurs if initial estimates that are likely to be extremely accurate are already available, such as when one is reestimating a model with a small amount of new data added to a time series. However, the main reason for specifying initial parameter values is to deal with convergence problems that may arise in difficult estimation situations.

When `X-12-ARIMA`'s iterative estimation scheme fails to converge, several remedies are available. If the program stopped short of convergence because it reached the maximum number of iterations (indicated by a warning message to this effect and the printing of parameter values at the last iteration), then rerunning the program with initial parameter values set at the values obtained at the last iteration may produce convergence. An easier, though computationally slower, alternative is to simply increase the number of iterations allowed and rerun the program. If the program crashed before converging or reaching the maximum number of iterations, then it may help to first fit the model by conditional likelihood, and then use the resulting parameter estimates as initial values for exact maximum likelihood estimation. On the other hand, it has been our experience that convergence problems are often due to the use of a model that is complicated (e.g., high order), or poorly conditioned. In such cases, the appropriate action is to examine the results and specify a simpler model. Sections 4.2 through 4.4 discuss some particular situations that can lead to estimation problems and that suggest specific model modifications.

4.2 Invertibility (of MA operators)

An MA polynomial, $\theta(B) = 1 - \theta_1 B - \cdots - \theta_q B^q$, is invertible if all the roots, $G_1, \ldots, G_q$, of $\theta(B) = 0$ lie outside the unit circle ($|G_j| > 1$ for all j). As shown in Brockwell and Davis (1991, pp. 123–125), for any invertible MA operator in an ARIMA model there are one or more corresponding noninvertible MA operators that produce the same autocovariance structure, and hence the same unconditional likelihood function. Although the data thus cannot discriminate between the invertible and corresponding noninvertible models, the preferred choice is the invertible model. This is essential for forecasting—grossly incorrect forecasting results can be obtained with noninvertible models. There is one important exception. MA polynomials with roots on the unit circle ($|G_j| = 1$), the boundary of the invertibility region, do not cause problems for forecasting when handled appropriately (by exact maximum likelihood for MA models).

Estimation in `X-12-ARIMA` enforces invertibility constraints on the MA parameters in the iterative nonlinear maximization of the likelihood function. Strictly speaking, then, models estimated by `X-12-ARIMA` are invertible. If the maximum likelihood estimates (MLEs) for a given model are actually on the boundary of the invertibility region, i.e., the model at the MLEs contains an MA operator with zeroes exactly on the unit circle, then `X-12-ARIMA`'s nonlinear search will approach the boundary of the invertibility region from within, and will generally get as close to the boundary as the convergence tolerance dictates or the maximum number of iterations allows. `X-12-ARIMA` can thus effectively produce estimated models on the boundary of the invertibility region. Convergence of the estimation iterations in such cases can be slow, since finding the maximum of the likelihood function on the boundary of the constrained parameter space is a difficult optimization problem. More importantly, convergence of the estimation to the invertibility boundary often indicates that the model is poorly conditioned, and should alert users to examine the results (and possibly detailed output of the estimation iterations) for signs of this. Section 4.4 discusses the most important causes of poor conditioning—cancellation of factors and overdifferencing—and the appropriate remedies.

Estimation seems most likely to produce a noninvertible model when the model contains a seasonal difference and a seasonal MA polynomial, e.g., $1 - \Theta B^s$ when the MLE of Θ is 1. As such models are commonly used for seasonal economic time series, users should be alert to this possibility and be aware of the appropriate action to take as discussed in Section 4.4.

4.3 Stationarity (of AR operators)

An AR polynomial, $\phi(B) = 1 - \phi_1 B - \cdots - \phi_p B^p$, is stationary if all roots of $\phi(B) = 0$ lie outside the unit circle; otherwise, it is nonstationary. (More accurately, the series $w_t = (1 - B)^d (1 - B^s)^D z_t$ following the model $\phi(B)\Phi(B^s) w_t = \theta(B)\Theta(B^s) a_t$ (derived from equation (1)) is stationary if and only if the zeroes of all the AR polynomials lie outside the unit circle.) The exact (for AR) likelihood function assumes all AR operators are stationary. Hence, the exact (for AR) likelihood can be evaluated, and estimation and other analyses (e.g., forecasting) performed, only if the AR parameters satisfy stationarity constraints. Thus, when the exact (for AR) likelihood function is used, `X-12-ARIMA` enforces stationarity constraints on the estimation. Unless cancellation of factors is present (see the next section), it is unlikely for X-12-ARIMA's nonlinear estimation to approach the boundary of the stationary region, since the log-likelihood approaches $-\infty$ as this boundary is approached.

If the likelihood is defined conditionally with respect to the AR parameters, stationarity is neither assumed nor enforced by X-12-ARIMA. Model estimation, forecasting, etc., are not compromised by parameter values outside the stationary region in this case. Inference results, however, are affected, as noted in Section 3.5. Special techniques (as in, e.g., Fuller 1976, Section 8.5) are required for inference about AR parameters outside the stationary region.

4.4 Cancellation (of AR and MA factors) and overdifferencing

Cancellation of AR and MA factors is possible when a model with a mixed ARMA structure is estimated. A model as in (1) or (3) is said to have a mixed ARMA structure if either $p > 0$ and $q > 0$, or $P > 0$ and $Q > 0$. (Technically, a model with $p > 0$ and $Q > 0$, or with $P > 0$ and $q > 0$, is also mixed, but such mixed models are unlikely to lead to cancellation problems.) The simplest example of cancellation occurs with the ARMA(1,1) model, $(1-\phi B)z_t = (1-\theta B)a_t$, when $\phi = \theta$. Cancelling the $(1-\phi B)$ factor on both sides of the model $(1-\phi B)z_t = (1-\phi B)a_t$ leaves the simplified model, $z_t = a_t$. Because of this, the likelihood function will be nearly constant along the line $\phi = \theta$. This can lead to difficulties with convergence of the nonlinear estimation if the MLEs for the ARMA(1,1) model approximately satisfy $\hat{\phi} = \hat{\theta}$. Analogous problems occur in more complicated mixed models when an AR polynomial and an MA polynomial have a common zero (e.g., the ARIMA (2,1,2)(0,1,1) model that is used as a candidate model for the **automdl** spec). For a fuller discussion of this topic, see Box and Jenkins (1976, pp. 248-250).

If the X-12-ARIMA program has difficulty in converging when estimating a mixed model, cancellation of AR and MA factors may be responsible. In any case, possible cancellation can be checked by computing zeroes of the AR and MA polynomials (setting `print=roots` in the **estimate** spec), and examining these for zeroes common to an AR and an MA polynomial. If a common zero (or zeroes) is found, then the model should be simplified by cancelling the common factor(s) (reducing the order of the corresponding AR and MA polynomials), and the model should be reestimated. Cancellation need not be exact, but may be indicated by zeroes of an AR and an MA polynomial that are approximately the same.

It is also possible for estimated MA polynomials to have factors that cancel with differencing operators. This occurs when a model has a nonseasonal difference and an estimated nonseasonal MA polynomial contains a $(1 - B)$ factor, or the model has a seasonal difference and an estimated seasonal MA polynomial contains a $(1-B^s)$ factor. For example, the model $(1-B)(1-B^s)z_t = (1-\theta B)(1-\Theta B^s)a_t$ involves such cancellation if either θ or Θ is estimated to be one. Such cancellation is called "overdifferencing", since it implies that the series was differenced more times than necessary to achieve stationarity. When overdifferencing occurs the corresponding difference and MA factor may be cancelled to simplify the model, but the user must then also add to the model regression term(s) to account for the deterministic function of time that was previously annihilated by the cancelled differencing operator. This means that if a nonseasonal difference is cancelled with a $(1-\theta B)$ MA factor with $\hat{\theta}=1$, then the simplified model should include a trend constant (or overall mean, if the model had only this one difference). If a seasonal difference is cancelled with a $(1-\Theta B^s)$ seasonal MA factor with $\hat{\Theta}=1$, then the simplified model should include both a trend constant (or overall mean) and fixed seasonal effects. Overdifferencing is discussed by Abraham and Box (1978) and Bell (1987).

If estimation converges to an overdifferenced model, modifying the model by removing the differencing operator and MA factor that cancel as well as including the appropriate regression terms, and then reestimating the model, is somewhat optional, because this cancellation does not necessarily lead to problems with model estimation and other results (assuming use of the likelihood function that is exact for the MA parameters). In particular, forecasting results should be the same for both the overdifferenced model and the corresponding modified model, and regression and ARMA parameter estimates and standard errors under the two models should be approximately the same. (However, log-likelihood values and the corresponding model selection criteria will be different for the two models—see the next section.) This contrasts with the situation regarding cancellation of AR and MA factors. Since cancellation of AR and MA factors is more likely to lead to convergence problems in estimation, common AR and MA factors should always be removed from the model, and the model reestimated.

4.5 Use of model selection criteria

The `X-12-ARIMA` program provides the following model selection criteria: AIC (Akaike 1973, see also Findley 1985, 1999), AICC (Hurvich and Tsai 1989), a criterion due to Hannan and Quinn (1979), and BIC (Schwarz 1978). Suppose the number of estimated parameters in the model, including the white noise variance, is n_p. If after applying the model's differencing and seasonal differencing operations, there are N data, and if the estimated maximum value of the exact log likelihood function of the model is denoted L_N, then the formulas for these criteria are:

$$\begin{aligned}
AIC_N &= -2L_N + 2n_p \\
AICC_N &= -2L_N + 2n_p\left\{\frac{N}{N-\frac{n_p+1}{N}}\right\} \\
HannanQuinn_N &= -2L_N + 2n_p \log\log N \\
BIC_N &= -2L_N + n_p \log N.
\end{aligned}$$

For each criterion, among competing models for a given times series, the model with the smallest criterion value is model preferred by the criterion. Some important points about the use of such criteria need to be kept in mind in making model comparisons. The first is that such comparisons are valid only between models that have the same differencing operators. One implication of this is that the model selection criteria cannot be used to determine the order of differencing required for a series. The second point is that for the model comparisons to be valid, the model estimations should be performed by exact maximum likelihood with respect to both the AR and MA parameters (the program default). For this reason `X-12-ARIMA` does not print the model selection criteria whenever a conditional likelihood function is used.

These model selection criteria should not be used to compare regARIMA models with different outlier regressors (AO, LS, TC, or ramp regressors). In particular they should not be used to decide which outliers to include in the model. There are theoretical and practical reasons. The basic theoretical problem is that, in the derivation of these criteria, it is assumed that the model coefficients can be estimated with unlimited precision given enough future data and this is not the case for the outlier regression coefficients.

However, if you are comparing models with the same outlier variables, empirical experience and heuristic calculations suggest that the outlier regression estimation will affect the maximum likelihood values of both models in about the same way (meaning their effects will almost cancel out when you take the differences of the log likelihoods). For example, using an AO regressor at time t has almost the same effect as treating the observation at time t as missing (see Gomez, Maravall, and Pena, 1999) so you are, in essence, treating the same observations as missing in both models. But if you have different AO regressors in the two models, you are, in effect, estimating the models from different data sets, so there is no justification for comparing their log likelihoods or other statistics calculated from the log likelihoods. (This is also the reason that the model selection criteria require seasonal and nonseasonal differencings to be the same when two models are compared. The likelihoods are calculated from the differenced data, and different differencings lead to different data.)

In practice, if you use the model selection criteria to compare models with different outlier regressors, the model with the larger number of these will usually be selected, but its forecasts may be worse. Also, some of the outliers in the selected model may no longer be identified as outliers after another year or two of data are available, so the model identification will be unstable.

Transformations and prior adjustments affect the model selection criteria. For example, if the **transform** spec includes the **function** argument with a value other than `none`, or the **power** argument with a value of $\lambda \neq 1$, then the transformation affects the likelihood function defined in terms of the original data, and hence it affects the model selection criteria. For this reason `X-12-ARIMA` computes a "transformation adjustment" to calculate the log-likelihood function as determined by the original (untransformed) data. The transformation adjusted log-likelihood is used to produce the model selection criteria. This way, for a given series, models that involve different transformations and/or prior-adjustment factors can be compared. Notice that this also implies that if a time series is transformed *before* input to `X-12-ARIMA` (e.g., the logarithm of the original time series is read into `X-12-ARIMA`), then different model selection criteria will be produced than if the transformation is performed within `X-12-ARIMA`. The same comment applies to use of prior-adjustment factors. Therefore, the model selection criteria should be used only to compare different models for the same input series. This also rules out comparing model selection criteria obtained from different segments of a series using the **span** argument of the **series** spec.

5. The Specification File and Its Syntax

The main input to `X-12-ARIMA` comes from a special input file called a specification file. This file contains a set of specifications or **specs** that give `X-12-ARIMA` various information about the data and the desired seasonal adjustment options and output, the time series model to be used, if any, etc. The different **specs** are:

- **series** a required spec except when composite adjustment is done. It specifies the time series data, start date, seasonal period, span to use in the analysis, and series title,
- **composite** specifies that both a direct and an indirect adjustment of a composite series be performed; it is used instead of the **series** spec,
- **transform** specifies a transformation and/or prior adjustment of the data,
- **x11** specifies seasonal adjustment options, including mode of adjustment, seasonal and trend filters, an Easter holiday adjustment option, and some seasonal adjustment diagnostics,
- **x11regression** specifies irregular regression options, including which regressors are used and what type of extreme value adjustments will be made to robustify the regression on the irregular component,
- **automdl** specifies automatic model selection procedure,
- **arima** specifies the ARIMA part of the regARIMA model,
- **regression** specifies regression variables used to form the regression part of the regARIMA model, and to determine the regression effects removed by the **identify** spec,
- **estimate** requests estimation or likelihood evaluation of the model specified by the **regression** and **arima** specs, and also specifies estimation options,
- **check** produces statistics useful for diagnostic checking of the estimated model,
- **forecast** specifies forecasting with the estimated model,
- **outlier** specifies automatic detection of additive outliers and/or level shifts using the estimated model. There is an optional test for temporary level shifts,
- **identify** produces autocorrelations and partial autocorrelations for specified orders of differencing of the data with regression effects (specified by the **regression** spec) removed for ARIMA model identification,
- **slidingspans** specifies that a sliding spans analysis of seasonal adjustment stability be performed,
- **history** requests the calculation of a historical record of seasonal adjustment revisions and/or regARIMA model performance statistics.

Each spec is defined in the spec file by its name, which is followed by braces {} containing ***arguments*** and their assigned ***values***. The arguments and their value assignments take the form *argument = value*, or, if multiple values are required, *argument = (value*$_1$*, value*$_2$*, ...)*. There are various types of values: titles, variable names, keywords, numerical values, and dates. These are defined and illustrated in the documentation of the individual specs in Section 6. Because of their occurrence in several specs, detailed discussions of the **print** and **save** arguments (Section 5.1), and **date** argument values (Section 5.2) are given below.

There are no required arguments for any spec other than either **series** or **composite** (see below). Most arguments have default values; these are given in the documentation of each spec. Default values for all arguments are used if no arguments are specified.

Typically, not all specs are included in any one spec file. In fact, for most `X-12-ARIMA` runs (any that is not a composite run) there is only one required spec in the specification file—the **series** spec. This spec must include either the **data** or **file** argument. (The only exception is when a data metafile is used; see Section 2.5.2 for more details.) Thus, `X-12-ARIMA` will accept the minimal spec file `series {data=(` *data values* `)}`. However, this spec file produces no useful output. For seasonal adjustment runs, the **x11** spec is needed, unless one or more of the **x11regression**, **slidingspans**, or **history** (with the **estimates** argument set to perform a seasonal adjustment history) specs are present. In this case, `X-12-ARIMA` behaves exactly as if the **x11** spec were present with default arguments, which is equivalent to including `x11{}` in the spec file. For model identification runs, the **identify** spec is needed. For model estimation, the **arima** and/or **regression** specs, and the **estimate** spec are ordinarily included. If the **estimate** spec is absent, but one or more of the **outlier**, **automdl**, **check**, **forecast**, **x11**, **slidingspans** and **history** specs is present, this forces estimation of the specified model. In this case, `X-12-ARIMA` behaves exactly as if the **estimate** spec were present with default arguments, which is equivalent to including `estimate{}` in the spec file. If the **arima** spec is absent, estimation proceeds with the default ARIMA(0 0 0) model (white noise). This is equivalent to including `arima{}` in the spec file.

The order of the specification statements in the spec file (with one exception), and the order of arguments within the braces of any spec do not matter. The only requirement is that **series** or **composite** must be the first spec. The spec file is free format, and blank spaces, tabs, and blank lines may be used as desired to make the spec file more readable. Comments can also be included. The use of comments and other general rules governing input syntax are discussed in Section 5.3. **Important:** *There must be a carriage return at the end of the last line of the spec file, otherwise, this line will not be read. This is a FORTRAN requirement.*

A very simple spec file producing a default X-11 run is given in Example 5.1. The spectrum diagnostics in the output file of this run indicated the presence of a trading day component, and a message saying this was written in the output. A regARIMA model can be used to both estimate the trading effect and to extend the series by forecasts prior to seasonal adjustment. Examples 5.2 and 5.3 illustrate spec files that might be used to identify the ARIMA part of the model before the final seasonal and trading day adjustment is achieved in Example 5.4. Alternatively, the X-11 trading day adjustment procedures described in the part of Chapter 6 dealing with the X-11 spec could be used.

It is customary to make at least two runs of `X-12-ARIMA` when modelling a time series. The first run is usually done to permit identification of the ARIMA part of the model; the second run is done to estimate and check the regARIMA model, and possibly to use it in forecasting the series. The spec file for the first run requires the **series** and **identify** specs, and may also include the **transform** and **regression** specs. The spec file for the second run includes the **series**, **arima**, and **estimate** specs; possibly the **transform** and **regression** specs; and the **outlier**, **check**, and **forecast** specs as desired. The two runs of `X-12-ARIMA` require two different spec files, or, more conveniently,

Example 5.1 : X-12-ARIMA spec file for a default X-11 run

```
series{title = "Monthly Retail Sales of Household Appliance Stores"
       data = ( 530  529  526  532  568  785  543  510  554  523  540  599
                574  619  619  600  652  877  597  540  594  572  592  590
                632  644  621  604  613  828  578  533  582  605  660  677
                682  684  700  705  747 1065  692  654  719  690  706  759
                769  730  740  765  791 1114  695  680  788  778  780  805
                852  823  831  836  913 1265  726  711  823  780  844  870
                865  915  920  935 1030 1361  859  852  954  895  993 1109
               1094 1173 1120 1159 1189 1539 1022  987 1024 1005 1054 1098
               1191 1191 1161 1201 1294 1782 1154 1059 1178 1126 1120 1233
               1260 1311 1302 1365 1395 1899 1123 1087 1210 1157 1159 1260
               1357 1265 1231 1287 1452 2186 1309 1242 1388 1400 1397 1527
               1654 1650 1555 1560 1836 2762 1541 1480 1619 1455 1510 1698
               1651 1749 1783 1863 2074 3051 1836 1690 1856 1796 1904 1927
               1978 2055 1976 2204 2423 3502 1977 1767 1935 1900 2073 2143
               2299 2247 2162 2274 2529 3731 2184 1901 2058 1974 2018 2091
               2239 2253 2157 2190 2397 3659 2170 2086 2297 2251 2311 2520)
       start = 1972.jul}
x11{}
```

the spec file from the first run can be modified for use in the second run. If diagnostic checking suggests changes need to be made to the estimated model, then the spec file can be modified again to change the model for a third run of X-12-ARIMA.

Example 5.2 : X-12-ARIMA spec file for regARIMA model identification

```
series{title = "Monthly Retail Sales of Household Appliance Stores"
       data = ( 530  529  526  532  568  785  543  510  554  523  540  599
                574  619  619  600  652  877  597  540  594  572  592  590
                  .
                  .
                  .
               2239 2253 2157 2190 2397 3659 2170 2086 2297 2251 2311 2520)
       start = 1972.jul}
transform{function = log}
regression{variables = td}     # Comment:  Series has trading-day effects
identify{diff=(0, 1) sdiff = (0, 1)}
```

The contents of a typical spec file for the model identification run might follow the same format as Example 5.2. This spec file includes the **series**, **transform**, **regression**, and **identify** specs. It provides X-12-ARIMA with the data given in the **series** spec, takes the logarithm of the series (**transform** spec), and specifies regression variables (**regression** spec) known or suspected to affect the series. Here, `variables = td` includes the six trading-day contrast variables (`td6`) in the model and also adjusts the series for leap year effects. (See Section 3.3 and the documentation of the **regression** spec in Section 6.) The **identify** spec performs a regression of the differenced transformed series (also adjusted for length-of-month effects) on the differenced regression variables (the six trading-day variables). The regression uses the highest order of seasonal and nonseasonal differencing specified, $(1-B)(1-B^{12})$. The **identify** spec then computes a regression residual series for the undifferenced data from which it produces tables and line printer plots of the sample autocorrelation and partial autocorrelation functions for all combinations of seasonal and nonseasonal differencing specified (here, four sets of ACFs and PACFs).

After studying the output from the first run and identifying the ARIMA part of the model as, for example, $(0\ 1\ 1)(0\ 1\ 1)_{12}$, the **identify** spec is commented out and the **arima** and **estimate** specs are added to the spec file. The resulting spec file is given in Example 5.3 (the data are not reproduced in full).

Example 5.3 : X-12-ARIMA spec file for regARIMA model estimation

```
series{title = "Monthly Retail Sales of Household Appliance Stores"
       data = ( 530  529  526  532  568  785  543  510  554  523  540  599
                574  619  619  600  652  877  597  540  594  572  592  590
                 .
                 .
                 .
               2239 2253 2157 2190 2397 3659 2170 2086 2297 2251 2311 2520)
       start = 1972.jul}
transform{function = log}
regression{variables = td}     # Comment:  Series has trading-day effects
# identify{diff=(0, 1) sdiff = (0, 1)}
arima{model = (0,1,1)(0,1,1)}
estimate{print = iterations}
```

This spec file includes the **series**, **transform**, **regression**, **arima**, and **estimate** specs. It specifies (**regression** and **arima** specs) and fits (**estimate** spec) the following model:

$$(1-B)(1-B^{12})\Big(y_t - \sum_{i=1}^{6} \beta_i T_{it}\Big) = (1-\theta B)(1-\Theta B^{12})a_t,$$

where the T_{it} are the six trading-day regression variables. The series y_t being modelled consists of the logarithms of the original data adjusted for leap-year effects. If diagnostic checking of residuals, outlier detection, or forecasting were desired, the appropriate specs would need to be added to the spec file.

Assuming this is a satisfactory model, a seasonal adjustment utilizing forecast extension can be performed by adding the **x11** and **forecast** to the input specification file. Such a spec file appears in Example 5.4 (the data are not reproduced in full).

The spec file now generates seasonal adjustments from 3×9 seasonal filters (**x11**) for the trading day pre-adjusted series. The pre-adjusted series is extended by 60 forecasts (**forecast**) prior to seasonal adjustment. The main output file will contain some

Example 5.4 : X-12-ARIMA spec file for seasonal adjustment

```
series{title = "Monthly Retail Sales of Household Appliance Stores"
       data = ( 530  529  526  532  568  785  543  510  554  523  540  599
                574  619  619  600  652  877  597  540  594  572  592  590
                 .
                 .
                 .
               2239 2253 2157 2190 2397 3659 2170 2086 2297 2251 2311 2520)
       start = 1972.jul}
transform{function = log}
regression{variables = td}     # Comment:  Series has trading-day effects
# identify{diff=(0, 1) sdiff = (0, 1)}
arima{model = (0,1,1)(0,1,1)}
estimate{print = iterations}
forecast{maxlead = 60}
x11{seasonalma = s3x9}
```

diagnostics concerning the quality of the seasonal adjustment. Additional diagnostics can be specified by including the appropriate specs described in Chapter 6.

5.1 Print and save

Control of the output from `X-12-ARIMA` is achieved within individual specs by using the **print** and **save** arguments. The **print** argument controls the given spec's output to the main output file, while the **save** argument allows certain output tables to be written to files. For ease of reference we refer to all the individual parts of the output subject to control through **print** and **save** as "tables", even though some of this output (e.g., line printer plots of an ACF) is not in a form that is ordinarily thought of as a table. The tables subject to control through **print** and **save** are listed with their default print status and file extensions (for savable tables) under the documentation of the **print** and **save** arguments for each spec. Tables output to files using **save** are written in a format with high numerical precision and with minimal or no labelling information to facilitate their use for further analysis utilizing other software. Saved tables are also given a consistent format—a single tab separates fields.

Default output from a spec is written to the main output file if the **print** argument is absent, or if `print=default` or `print=()` appears in the spec. To stop a spec from writing output to the main output file, set `print=none`. (Note: A few specs write some minor labelling information to the screen even with `print=none`.) To have all the available output tables and plots for a spec written to the main output file, set `print=all`. To have all the available output tables (no plots) for a spec written to the main output file, set `print=alltables`. To have a small subset of the available output tables for a spec written to the screen, set `print=brief`. Individual tables may be added to the `default`, `brief`, and `none` print levels by including their names as print argument values. These may (but need not) be preceded by a `+`. For example, in the **estimate** spec, `print=(+iterations +residuals)`, which is equivalent to `print = (default +iterations +residuals)`, requests printing of results from the estimation iterations and the residuals from the estimated model, in addition to the default output. Using `print=(none estimates)` requests printing of only the parameter estimates. Individual tables may be suppressed from the `default` and `all` print levels by including their names preceded by a `-` as print argument values, e.g., `print=brief -acf` or `print=(all - iterations)`.

If the user wishes to save any output tables to files, these must be specifically listed in the **save** arguments of the appropriate specs, e.g., `save=(mdl estimates)` in the **estimate** spec. Those tables that are savable may be specified in the **print** and **save** arguments using either a "long" name, the name listed in the spec's description, or a "short" 3-letter name, which is the same as the file extension used if the table is saved. For example, the optional table `regcmatrix` in the **estimate** spec can also be specified as `rcm`. The keywords `none`, `all`, `alltables`, `default`, and `brief` defined above are not available for use in the **save** argument. Also, names of tables to be saved should not be preceded with a `+` or `-`. Not all tables are savable, and not all specs produce savable tables.

The **save** argument writes the specified tables to individual files. A saved file will be placed in the same directory as the output, and will be given the filename of the main output file, but with a distinct 3-letter extension. If a file with this name already exists, it will be overwritten. The extensions used are listed under the documentation of the **print** and **save** arguments for each spec. For example, suppose `X-12-ARIMA` is run (on a DOS machine) from the directory `C:\TSERIES` using as input a spec file stored in `SALES.SPC` in that directory. If the **estimate** spec contains `save = (mdl estimates)`, the resulting saved tables of the model and parameter estimates will be written to the files `C:\TSERIES\SALES.MDL` and `C:\TSERIES\SALES.EST`. If files with these names already exist, they will be overwritten. Although the extensions used by `X-12-ARIMA` have been chosen to avoid obvious conflicts (examples of extensions not used are `.dat`, `.exe`, `.com`, `.for`, `.spc`), users should still exercise caution to prevent unintended overwriting of files by `X-12-ARIMA` saves. A list of the files saved, with an `*` indicating those overwriting existing files, appears at the beginning of `X-12-ARIMA`'s output. If there are errors in the spec file or the program terminates prematurely for other reasons, some or all of the saved files may not be written.

5.2 Dates

Date arguments occur in several specs, and their values are always specified in the same format. Dates for monthly data are written *year.month*; this format generalizes to other seasonal periods (e.g., *year.quarter*). It is necessary to include all four digits when specifying a year. Thus, `67` means the year AD (or CE) 67, not AD 1967.

For monthly data the months can be denoted by either the integers 1–12 or by three-letter month abbreviations (`jan`, `feb`, `mar`, `apr`, `may`, `jun`, `jul`, `aug`, `sep`, `oct`, `nov`, and `dec`). Thus, `1967.12` and `1967.dec` are equivalent. For quarterly data or data with other seasonal periods, only integers are allowed, e.g., `1967.1` and `1967.4`.

Dates are used to define the starting time point of a series, and when defining a subset (**span**) of a time series to analyze. They are also used when defining outlier regression variables. For example, to specify regression variables for an additive outlier in April of 1978 and a level shift beginning in September of 1982, we use the following:

```
regression  {  variables=(ao1978.apr ls1982.sep)  }
```

The seasonality of the dates used must match the seasonality specified for the data in the **series** spec, e.g., `ao1972.jan` is valid for monthly data but is not permitted for quarterly data.

5.3 General rules of input syntax

Allowable input characters

The allowable input characters, excepting characters that appear within quotes, are letters, numbers, spaces, tabs, newline characters, and the following: `= . , { } ( ) [ ] + - #`. The program will ignore any other ASCII characters in the spec file, but will flag them and generate a warning message. The following additional characters are allowed within quotes: `! % ' * / : ; < > ? @ \ _ | ~ ^`. Also, double quotes are allowed within statements delimited by single quotes and vice-versa.

Braces, parentheses, and brackets

The { }, ( ), and [ ] characters serve different functions and cannot be used interchangeably. { } is used to contain arguments in a spec, ( ) is used to contain a list of multiple values for an argument, and [ ] is used (*i*) to contain values used in defining certain special arguments, such as the duration of an Easter holiday regression variable, e.g., `regression` {`variables = (td Easter[14])`}, and (*ii*) to enclose the lags present in an ARIMA model with missing lags, e.g., `arima` {`model = (0 1 [1,3])`}.

Case sensitivity

Spec names, arguments, dates, keywords (such as `none` and `all`), and predefined regression variable names (such as `td` and `seasonal`) are not case sensitive. Thus, `TD` and `td` are the same to `X-12-ARIMA`; both are recognized by the **variables** argument of the **regression** and **x11regression** specs.

Comments

Anything on a line after the `#` character, unless the `#` character is in quotes, is taken to be a comment. If parts of a spec are commented out, what remains must still have balanced parentheses, brackets, and braces.

Equals sign

The equals sign, `=`, is used when assigning values to arguments, e.g., `print = none`, or `title = "Monthly Retail Sales of Household Appliance Stores"`.

Line length in the spec file

Lines in the spec file are limited to 132 characters—any characters appearing beyond column 132 are ignored. In particular, note that if a data set with lines exceeding 132 characters is placed in a spec file this will result in data truncation on input. The 132 characters per line limitation does not apply, however, to data read from a separate file (not the spec file) using the **file** argument. (The latter would be governed by FORTRAN input line length restrictions, which may be system specific.)

Multiple argument values

Multiple argument values must be enclosed together in parentheses, e.g., `variables=(td seasonal const)`. If an argument accepts only a single value or it accepts multiple values but only one value is given, then parentheses are optional. For example, the following are all valid; `variables=td`, `variables=(td)`, `variables = (td seasonal)`, `start=1967.4`, and `start=(1967.4)`.

Null list

A null list of arguments is allowed, e.g., `outlier{ }`. Any implied arguments in the null list then take on their default values.

Numerical values

Numerical values can be specified in free format, including the use of exponential notation (e.g., 400, 400.0, 400., and 4.e+2 all denote the same real value). Integer notation must be used when an integer is required (e.g., 2, not 2.0 or 2.e+0).

Ordering

The only restriction on the ordering of *specs* is that either **series** or **composite** must be the first spec. There is no restriction on the ordering of *arguments* within specs. The ordering of multiple *values* given to arguments matters for certain obvious cases, such as observations in **data** arguments (**series**, **transform**, and **regression** specs), the ARIMA model specification in the **model** argument (**arima** spec), and dates in **span** arguments (**series** and **outlier** specs).

Separators

Blank spaces, tabs, and blank lines may be used as separators as desired. Within a list of multiple argument values, single commas may also be used as separators, e.g., `data=(0, 1, 2, 3, 4, 5)`. Commas *must* be used to indicate missing argument values that are to be replaced by default values (for arguments that require a specific number of values). For example, the **span** argument requires two values. In the statement `span=(1967.4, )`, the presence of the comma after 1967.4 indicates that the second **span** argument value is missing, so it takes on its default value (the date of the last observation).

Titles and filenames

A title, such as the name of a time series, must consist of at least one allowable input character (see above), even if blank, and must be enclosed in either single or double quotes (`'title'` or `"title"`). Lower and upper case of characters is preserved within titles. When the `#` character appears within quotes, it is considered part of the title and does not denote the start of a comment. Titles must be completed on one line and contain no more than 79 characters. Filenames, including the path, must follow the same rules as titles.

6. Documentation for Individual Specs

The following pages provide detailed documentation on all of the specs, with discussions of the available arguments and their default values. Each spec's documentation also includes several examples illustrating its use. For the **series** and **transform** specs the examples are intended only to illustrate the capabilities of these specs. They do not show complete spec files in the sense that if these examples were used as input to `X-12-ARIMA`, they would produce no useful output. For the remaining specs (**composite**, **x11**, **identify**, **regression**, **arima**, **estimate**, **outlier**, **check**, **forecast**, **x11regression**, **slidingspans**, and **history**) the examples all show complete spec files that could be used, except that data sets (e.g., for the input series appearing in the **series** spec, or for a user-defined regression variable in the **regression** spec) are often abbreviated using the ... notation. Readers will notice that the examples for a given spec tend to vary, not only in content, but also in format. This is done deliberately to illustrate and emphasize the flexibility the user has in formatting the spec file.

The next few paragraphs will give the reader a summary of what specs to include in the input file when doing general tasks (such as a simple seasonal adjustment or modeling run). Except in certain default situations, arguments must be specified within each spec to accomplish these tasks. Information about these arguments can be formed within the sections of this chapter devoted to the individual specs.

For the reader who wants the shortest path to a seasonal adjustment, the essential specs are **series** and **x11**. These will yield a default X-11 seasonal adjustment. If it is not clear whether the seasonal adjustment should be additive or multiplicative then the **transform** spec should be added. If an elementary approach to trading day and moving holiday effect estimation and adjustment is desired, then add **x11regression**. The **sliding spans** and **history** specs provide diagnostics for the stability of the adjustment when the span of data used to calculate the adjustment changes.

For the reader wanting the shortest path to modeling a time series, the essential specs are **series**, **automdl**, and possibly **transform**. Add the **forecast** spec if forecasting is desired, add **outlier** if there are problematic data values or data movements, and add **regression** if trading day or holiday components may be present in the series. The **arima** spec replaces **automdl** if custom rather than automatic modeling is desired. It is supported by **identify**. The **check** spec provides standard model-fit diagnostics. The **history** spec provides forecasting diagnostics for comparing two models, and **estimate** offers estimation options and the ability to reuse stored models.

Time series models (obtained via **automdl/arima** and **transformation**) can improve seasonal adjustment by extending the data with forecasts (via **forecast**), by providing a way of dealing with disruptions to the level of the series (via **outlier**) and by providing estimates of trading day and holiday effects (via **regression**) that are sometimes better than those obtained from **x11regression**.

The **composite** spec is required to obtain the indirect adjustment of an aggregate series from adjustments of its components and to compare this adjustment with its direct adjustment. For indirect adjustment the **composite** spec replaces the **series** spec.