

# Journal of Statistical Software

November 2014, Volume 62, Issue 2.

http://www.jstatsoft.org/

# Seasonal Adjustment with the R Packages x12 and x12GUI

Alexander Kowarik

Statistics Austria

Angelika Meraner Statistics Austria

**Matthias Templ** 

TU Vienna Statistics Austria Daniel Schopfhauser

TU Vienna

#### Abstract

The X-12-ARIMA seasonal adjustment program of the US Census Bureau extracts the different components (mainly: seasonal component, trend component, outlier component and irregular component) of a monthly or quarterly time series. It is the state-of-the-art technology for seasonal adjustment used in many statistical offices. It is possible to include a moving holiday effect, a trading day effect and user-defined regressors, and additionally incorporates automatic outlier detection. The procedure makes additive or multiplicative adjustments and creates an output data set containing the adjusted time series and intermediate calculations.

The original output from **X-12-ARIMA** is somehow static and it is not always an easy task for users to extract the required information for further processing. The R package **x12** provides wrapper functions and an abstraction layer for batch processing of **X-12-ARIMA**. It allows summarizing, modifying and storing the output from **X-12-ARIMA** within a well-defined class-oriented implementation. On top of the class-oriented (command line) implementation the graphical user interface allows access to the R package **x12** without requiring too much R knowledge. Users can interactively select additive outliers, level shifts and temporary changes and see the impact immediately.

The provision of the powerful **X-12-ARIMA** seasonal adjustment program available directly from within R, as well as of the new facilities for marking outliers, batch processing and change tracking, makes the package a potent and functional tool.

Keywords: time series, seasonal adjustment, outlier detection, R.

# 1. Introduction

The decomposition of monthly or quarterly time series into trend, seasonal and irregular components is an important part of time series analysis. In seasonal adjustment, the seasonal component of a time series is removed to make it easier to focus on other components and for easier interpretation of the time series.

#### 1.1. Tools in R

Tools for time series analysis are widely available in R (R Core Team 2014). For example, R includes general tools for regular time series and forecasting (see e.g., Pfaff 2008; Hyndman and Khandakar 2008) or for handling irregular time series (Zeileis and Grothendieck 2005). A more comprehensive overview is given in the CRAN Task View on "Time Series Analysis" (Hyndman 2014).

Two functions in R's base package stats can be used to decompose time series. An approach using moving averages before obtaining the seasonal components is implemented by function decompose() (see e.g., Kendall and Alan 1983). Function st1() decomposes a time series based on loess smoothing (see e.g., Cleveland, Cleveland, McRae, and Terpenning 1990).

The function tbats() in the package forecast (Hyndman, Athanasopoulos, Razbash, Schmidt, Zhou, Khan, and Bergmeir 2014) implements the method based on exponential smoothing (see e.g., Livera, Hyndman, and Snyder 2011). After applying one of the three decomposition functions the seasonally adjusted data can be computed by using the function seasadj(). Forecasts can be made using function forecast().

Package **x12** is based on **X-12-ARIMA** (US Census Bureau 2013), see Section 1.3 for a first introduction.

#### 1.2. Specific tools outside R

Two software products which are widely used by statistical offices and which are focused on seasonal adjustment methods are the **X-12-ARIMA** software (see e.g., Time Series Research Staff Statistical Research Division 2011) and **TRAMO/SEATS** that have been developed by the Bank of Spain (see e.g., Maravall 2003).

Especially, **X-12-ARIMA** is a widely used seasonal adjustment software for time series analysis and forecast prediction, additionally the software **Win X-12** (US Census Bureau 2011) provides a basic graphical user interface. Both software packages are developed, distributed and maintained by the United States Census Bureau.

Essentially, the program extracts the different components, i.e., the seasonal component, trend component, outlier component and irregular component of a monthly or quarterly time series. It performs additive or multiplicative adjustments and creates output including the adjusted time series and intermediate calculations. Moving holiday effects and trading day effects can be directly accounted for, models can be selected and revised and various other diagnostic tools are available. In addition tools for automatic outlier detection are available.

The output from **X-12-ARIMA** is, however, somehow static. Users have to extract the required information for further processing manually and it is not possible to modify parameters in an interactive manner.

X-13ARIMA-SEATS (US Census Bureau 2014) is the successor of X-12-ARIMA. The com-

patibility between the two programs is very good. Therefore it is easily possible to use the executables of **X-13ARIMA-SEATS** with the R package **x12**. The main new feature is SEATS (signal extraction in ARIMA time series), which can be used instead of the X11 filters. This functionality is planned to be included in a future version of the R package.

**X-12-ARIMA** is integrated in various other software products, like gretl (Cottrell and Lucchetti 2014, see Baiocchi and Distaso 2003 for a review), EViews (IHS Inc. 2013), SAS/ETS (see e.g., Hood 2000), **Demetra+** (Grudowska 2012) and via the **Excel** add-in **NumXL** (Spider Financial Corp. 2009).

The support for X-12-ARIMA from gretl is quite limited and only the static output is presented as well as some basic graphics are supported. EViews provides a front-end for accessing X-12-ARIMA. Various parameters can be set via point and click and some results from X-12-ARIMA are presented after evaluation. However, the static output is not directly accessible, interactive features are missing and there is a lack of advanced visualization methods. Also SAS has an adaption/re-implementation of X-12-ARIMA methods implemented within its PROC X12 in the SAS add-on ETS for time series analysis. Various graphics are supported by PROC X12. Eurostat and the National Bank of Belgium developed JDemetra+ (Grudowska 2013), an (Java) interface to X-12-ARIMA (and TRAMO/SEATS). It consists of a point and click interface that allows the application of methods and diagnostics. It is the successor of Demetra+. NumXL is a lightweight time series Excel add-in and it provides basic X-12-ARIMA tools. Again, the output is static and implementation of suitable plots is missing.

### 1.3. The R package x12

With package **x12** the methods of **X-12-ARIMA** can now be accessed in R to account for calendar effects and trading day effects, selection and revision of models, diagnostics and outlier detection.

However, the R package x12 (Kowarik and Meraner 2014) is not just another implementation/integration of X-12-ARIMA. The stable version is freely available on the Comprehensive R Archive Network (CRAN, http://CRAN.R-project.org/package=x12) and listed in the CRAN Task Views "Time Series Analysis" and "Official Statistics" (Templ 2014). The latest development version can be obtained via GitHub (https://github.com/alexkowa/x12). The package combines several useful features such as:

- Batch processing: It provides wrapper functions and an abstraction layer for batch processing X-12-ARIMA calculations.
- Multiple approach: Parameter settings can be applied to multiple time series, i.e., various time series can be adjusted with one call using a defined set of parameters for each or for all series.
- Object orientation and output handling: It allows summarizing, modifying and storing the output from X-12-ARIMA in a more advanced and user-friendly manner within a well-defined S4 class (Chambers 2008) oriented implementation.
- **Restoring objects:** An archive of results and parameters is provided, so that it is an easy task for the user to restore or review previous settings.
- **Point and click interface:** On top of the class-oriented (command line) implementation, the graphical user interface (GUI) grants the less experienced R user access to package

**x12** via the R package **x12GUI** (Schopfhauser, Kowarik, and Meraner 2014). By simply clicking on a specific time point in a plot window inside the GUI, users can interactively select additive outliers, level shifts and temporary changes and get feedback immediately.

Note that some few arguments of certain X-12-ARIMA functions are not supported by the x12 package. This belongs only for less commonly used arguments like the ljungboxlimit argument that sets the rejection region for the Ljung-Box Q statistic (Ljung and Box 1978) in the automatic model selection procedure (automdl). However, the decision on supported arguments is subjective and function arguments will be added as soon there is a need for it.

#### 1.4. Outline of the paper

The paper is structured as follows. Section 2 is focused on X-12-ARIMA, Section 3 presents the command line implementation of R package x12 while Section 4 shows the functionality of R package x12GUI. More precisely, in Section 2, basic information on input files necessary for (batch) processing X-12-ARIMA is given and the elementary files created by X-12-ARIMA are described. Furthermore, the class structure of the command line version of R package x12 is explained in Section 3 and methods defined for these classes are presented. In the end of this section, an example shows some features of the command line version by processing four time series at once using different parameter settings. In Section 4, the functionality of the point and click user interface x12GUI is shown in a systematic manner. Section 5 concludes.

# 2. Overview of X-12-ARIMA

#### 2.1. Overview of the seasonal adjustment method used in X-12-ARIMA

The basic assumption of seasonal adjustment with **X-12-ARIMA** is the possibility to decompose a quarterly or monthly series  $Y_t$  into several components, namely the seasonal component  $S_t$ , the trend cycle component  $C_t$ , the trading day component (including additional calendar effects)  $D_t$  and a residual (irregular) component  $I_t$ . The two most commonly used decomposition models of a time series  $Y_t$  at time t are the multiplicative  $Y_t = S_t C_t D_t I_t$  and the additive  $Y_t = S_t + C_t + D_t + I_t$  models.

The decomposition is done with a combination of moving average filters called **X-11** (see e.g., Shiskin, Young, and Musgrave 1967). The most critical part is defining the length of the filters and the weights, which is done for the trend estimation by using Henderson filters (see e.g., Doherty 2001).

For preadjusting the series, an algorithm based on TRAMO (time series regression with ARIMA noise, missing observations and outliers) is used by **X-12-ARIMA** (see e.g., Maravall 2003). The REG-ARIMA model that is used in **X-12-ARIMA** is an extension of the ARIMA (autoregressive integrated moving average) class of models. For REG-ARIMA, the time-varying average of time series is modeled with linear regression whereas the error term follows a seasonal ARIMA model. REG-ARIMA models are used for fore- and backcasting (to be able to use symmetric filters at the beginning and end of the series) and to deal with calendar effects and outliers. There are three types of outliers that can be identified automatically: additive outliers, level shift outliers and temporary change outliers (Akaike 1973b). Ramp outliers, the fourth category of outliers can only be added manually. Corresponding intervention analysis

(see Box, Jenkins, and Reinsel 2008; implemented in **X-12-ARIMA**, see e.g., Findley, Monsell, Bell, Otto, and Chen 1998) is used to handle level shifts in the trends of a time series and to model the interventions from the different outlier types.

#### 2.2. How to obtain X-12-ARIMA

**X-12-ARIMA** and **X-13ARIMA-SEATS** can be downloaded from the homepage of the US Census Bureau. Precompiled versions for Windows and Linux on a x86 platform are available.

For Mac OS X and Linux on non-x86 platforms it is possible to download and compile the source code. For this it is necessary to rename the file makefile.lnx to Makefile and edit its content to be suitable for the relevant build environment (a Fortran compiler is needed).

### 2.3. Overview of basic implementation of X-12-ARIMA

To run **X-12-ARIMA**, an application dependent input file has to be provided by the user, i.e., the so called specification file with the file extension .spc (see e.g., Hood and Monsell 2010). It stores the path name and other general information about the analyzed data, the way it should be processed and the kind of output generated by **X-12-ARIMA**. There are several possible formats to store the input data (like *datevalue* format, free format, **X-11** format and **X12save** format).

A very basic input specification file minimal.spc could look like the following:

```
series {
  file = "airpassengers.dat"
  format = "datevalue"
}
```

It states that the data should be read from a file called airpassengers.dat which contains a dataset in the so called *datevalue* format. Calling **X-12-ARIMA** in the terminal would display the following:

C:\work\x12a minimal

```
X-12-ARIMA Seasonal Adjustment Program
Version Number 0.3 Build 188
PSP = 24
Execution began Jul 16, 2012 17.06.12

Reading input spec file from minimal.spc
Storing any program output into minimal.out
Storing any program error messages into minimal.err

WARNING: At least one visually significant seasonal peak has been found in one or more of the estimated spectra.

Execution complete for minimal.spc
C:\work\
```

File	Description
Output file (.out)	Contains the results of the calculations performed by
	X-12-ARIMA as well as additional diagnostics if that
	was specified and clarifications about the settings used.
Error file (.err)	Contains all the error and warning messages as shown
	in the terminal during the execution.
Log file (.log)	Contains additional diagnostics if requested in the
	specification file with the savelog-argument.
Diagnostics summary file (.udg)	Contains the most important diagnostics and will only
	be generated if specifically requested in the terminal.
Other files	Depending on the save argument in the specification
	file, a wide variety of output files can be created con-
	taining diagnostics in tabular or tabular-like format.

Table 1: Files created by **X-12-ARIMA**.

More details on the specification file can be found in the **X-12-ARIMA** reference manual (Time Series Research Staff Statistical Research Division 2011).

In every X-12-ARIMA run, several files are created including the output, error and log files (see e.g., Time Series Research Staff Statistical Research Division 2011; Hood and Monsell 2010). The actual selection of diagnostic output files can be specified by the user in advance.

# 3. The R package x12: Command line interface

#### 3.1. Class structure of R package x12

By using **x12** for R, the handling of specification and output files can be avoided. Parameters and output are easily accessible from within R. The main function is the **x12()** function, which prepares the input data and specification file before calling the **X-12-ARIMA** program. Afterwards it reads the generated output. The package provides classes for different applications which serve as data structure for several **X-12-ARIMA** related input and output tables, as well as methods for manipulating, retrieving, plotting and summarizing parameters or data.

The classes 'x12Single' and 'x12Batch' form the core of the package. An object of class 'x12Single' contains all the information for one time series including the parameters for the X-12-ARIMA run. The concatenation of several objects of class 'x12Single' is an object of class 'x12Batch', which is intended for batch processing. The most important functions are implemented as methods and can be found in Table 3, e.g., summary() and plot().

If the **X-12-ARIMA** parameters are changed iteratively, all the outputs can be compared. Moreover, it is possible to restore **X-12-ARIMA** parameters from a previous run.

The selection of X-12-ARIMA parameters available in the x12 package is stored in an object of class 'x12Parameter' and can be accessed via the setP and getP methods as described in Section 3.2. The parameters are mainly arranged according to X-12-ARIMA standard, i.e., grouped in accordance with the respective sets of specifications. As an example, the parameter arima.model refers to the model argument enclosed in the arima specification

Slot name	Description
ts	Contains the unmodified time series data.
x12Parameter	Contains the X-12-ARIMA parameters selected for the R package
	x12.
x120utput	Contains tables, lists and other information returned from an X-12-
	ARIMA run.
x120ldParameter	Previous X-12-ARIMA parameters for the archive functionality.
x120ldOutput	Previous X-12-ARIMA output for comparisons with previous runs.
tsName	The name of the time series.
firstRun	Logical specifying if $x12()$ was already executed.

Table 2: Slots of the 'x12Single' class.

of an .spc file. A list of all the parameters available can be found in the help file for the 'x12Parameter' class.

To run X-12-ARIMA with the data and settings specified in an object of class 'x12Single', 'x12Batch' or 'ts', the package provides the x12() method. The following short example creates an object of class 'x12Single' and performs an x12() run with it:

```
R> s \leftarrow \text{new}("x12Single", ts = AirPassengers, tsName = "air")}
R> s \leftarrow x12(s)
```

Hidden from the user, the function x12() creates an input specification file and calls X-12-ARIMA with this specification. The output is then imported back into R, saved into the corresponding class and returned.

As mentioned above, objects of class 'x12Single' contain all the information necessary to call X-12-ARIMA using the R function x12().

The output generated by **X-12-ARIMA** will be stored in objects of class 'x120utput', where most slots correspond to the tables returned by the program. These tables can be accessed through the summary() function and several plot functions or directly from the corresponding object, as can be seen in the following example where s is an 'x12Single' object:

R> orgts <- s0x120utput0a1

retrieves the original time series and

R> forecast <- s@x12Output@forecast

retrieves the calculated forecasts.

The same applies to the list of important diagnostic results which are stored in the dg slot of an 'x120utput' object.

#### 3.2. Methods in R package x12

Functions in the **x12** package are implemented in the form of methods. The most important of these are discussed below in more detail while Table 3 provides an overview.

Method	Object class	Description
getP()	'x12Single', 'x12Batch',	Retrieves X-12-ARIMA parameters.
	'x12Parameter'	
setP()	'x12Single', 'x12Batch',	Sets $X-12$ -ARIMA parameters.
	'x12Parameter'	
x12()	'x12Single', 'x12Batch',	Runs $X-12$ -ARIMA and creates diagnos-
	'ts'	tics output.
<pre>summary()</pre>	'x12Single', 'x12Batch',	Creates a diagnostics summary for $x12()$
	'x120utput'	output.
plot()	'x12Single', 'x12Output'	Plots x12() output.
<pre>plotSpec()</pre>	'x12Single',	Creates spectral plots.
	'x120utput', 'spectrum'	
<pre>plotSeasFac()</pre>	'x12Single', 'x12Output'	Creates seasonal factor plots.
<pre>plotRsdAcf()</pre>	'x12Single', 'x12Output'	Creates plots of the (partial) autocorrela-
		tions of the (squared) residuals.
<pre>prev()</pre>	'x12Single', 'x12Batch'	Reverts to previous parameter settings
		and output.
<pre>cleanArchive()</pre>	'x12Single', 'x12Batch'	Resets x120ldParameter() and
		x120ldOutput.
<pre>saveP()</pre>	'x12Single', 'x12Batch',	Saves parameter settings.
	'x12Parameter'	
loadP()	'x12Single', 'x12Batch',	Loads parameter settings.
	'x12Parameter'	

Table 3: Main methods of the R package x12.

#### Parameter handling

Setting, changing and retrieving X-12-ARIMA parameters from 'x12Single', 'x12Parameter' or 'x12Batch' objects can be done by means of the getP() and setP() methods. In case of setting parameters for 'x12Batch' objects, an additional element number vector allows the specification of a respective 'x12Single' object.

This can be illustrated with sObject of class 'x12Single' and bObject of class 'x12Batch':

```
R> sObject <- setP(sObject, argumentList)
R> ValueList <- getP(sObject, argumentList)
R> bObject <- setP(bObject, argumentList, elements)</pre>
```

# Plot methods

Beside the access of all parameters and results in an interactive manner, one of the biggest advantages of having **X-12-ARIMA** directly available from within the R environment is the possible usage of plotting functionality. The **x12** package presents the user with its own plot methods for 'x12Single' and 'x12Output' objects but also with more specialized plot functions in the context of seasonal adjustment.

The integrated plot() function is implemented in a flexible manner and can be used for a variety of different objectives. As an example, Figure 1a shows the original time series as

Parameter	Type	Description
original	logical	Specifies if the original time series should be plotted.
sa	logical	Specifies if the seasonally adjusted time series should be plot-
		ted.
trend	logical	Specifies if the trend should be plotted.
log_transform	logical	Specifies if a log transformation should be applied.
showAllout	logical	Specifies if all the outliers should be plotted.
showOut	character	Plots a specific outlier.
${\tt annComp}$	logical	Specifies whether an annual comparison should be performed
		for the outlier defined in showOut.
forecast	logical	Specifies if the forecasts should be plotted.
backcast	logical	Specifies if the backcasts should be plotted.
span	character	Vector of the form c(startyear, startperiod, endyear,
_		endperiod) determining the interval to be plotted.

Table 4: Important parameters of the plot() function for 'x12Single' and 'x12Output'.

well as the seasonally adjusted series, the trend and forecasts with the respective prediction intervals. Figure 3 on the other hand shows the original time series, the trend and outliers detected by **X-12-ARIMA**.

The corresponding help file for the plot function that comes with the package lists all plot parameters. These plot parameters specify, for example, if the trend should be plotted, if transformations should be applied, if outliers should be highlighted and if forecasts or backcasts should be plotted. They also control the appearance of the plot. Table 4 gives an overview of the most important plot parameters.

Function plotRsdAcf() allows the plotting of the (partial) autocorrelations (see e.g., Box et al. 2008) of the (squared) residuals which are included in 'x12Output' and therefore also in 'x12Single' objects. As with most functions that accept both of these types, the 'x12Single' version is merely a wrapper for its consort. This is also valid for the remaining plot functions mentioned below. The most important parameter of the plotRsdAcf() function is which. It expects one of the characters "acf", "pacf" and "acf2" standing for "autocorrelation of the residuals", "partial autocorrelation of the residuals" and "autocorrelation of the squared residuals" respectively. An example is shown in Figure 1b and discussed in Section 3.3.

The plotSeasFac() function graphically represents the seasonal factors (see also Brockwell and Davis 2009), which are contained in the 'x120utput' slots d10 and d8. The SI\_Ratios parameter specifies if the SI ratios (seasonal-irregular, detrended series) should be displayed while the rest of the parameters define the appearance of the plot. See Figure 1d for an example.

The plotSpec() function allows the plotting of spectral plots (see Box and Jenkins 1970; Box et al. 2008). Spectra concerning this function can be found in the 'x120utput' slots sp0 (original series), sp1 (differenced seasonally adjusted series), sp2 (modified irregular series) and spr (regARIMA model residuals) and are of class 'spectrum'. The parameter which defines the type of spectrum to be plotted, i.e., one of the characters "sa", "original", "irregular" and "residuals" has to be selected. Should the plotSpec() function be called for an object of class 'spectrum', an additional frequency parameter has to be passed to the

method. For an example, that is discussed later, see Figure 1c.

#### Diagnostics summary

The x12 package contains an implementation of the generic summary method for 'x120utput', 'x12Single' and 'x12Batch' classes. By calling the summary() function, a diagnostics summary for X-12-ARIMA output is printed which can also be returned in the form of a data frame for further processing. Note that various parameters can be set in the function summary(), see Kowarik and Meraner (2014). For a description of the parameters, see Table 5.

### Archive functionality

With the archive functionality, parameter settings and output stored in the 'x12Parameter' and 'x12Output' slots of 'x12Single' and 'x12Batch' objects can be reverted to a previous state. To make this possible, settings and output of every x12() run are stored in the respective x12OldParameter and x12OldOutput slots before being modified in a new run. Returning to previous settings is easily accomplished by means of the prev() function , i.e., prev(object, n = NULL) for 'x12Single' objects and prev(object, index = NULL, n = NULL) for 'x12Batch' objects, where n corresponds to the chosen index of a previous run. If old settings and output are no longer required, the cleanArchive() function can reset the x12OldParameter and x12OldOutput slots to the empty default state.

#### Saving and loading

Function saveP(object, file) serves the purpose of saving the X-12-ARIMA parameters stored in the x12Parameter slots of 'x12Single' and 'x12Batch' objects to a file for later use.

```
R> s <- new("x12Single", ts = AirPassengers)
R> s <- setP(s, list(arima.model = c(2, 1, 1), arima.smodel = c(2, 1, 1)))
R> saveP(s, file = "xyz1.RData")
They can be retrieved using loadP(object, file).
R> s <- new("x12Single", ts = AirPassengers)
R> s <- loadP(s, file = "xyz1.RData")</pre>
```

#### Experimental functionality

The function <code>readSpc()</code> can be used to read readily available parameter settings from a specification file into R. At the moment it is experimental and incorporates only a limited set of parameters.

To make assessments of the stability of the adjustment process, especially at the beginning and the end of the time series, the function crossVal() can be used to perform a kind of cross validation.

#### Files in the local file system

It should be noted, that the R package x12 writes files to the local file system. At first the .spc file will be generated and then, when X-12-ARIMA calling on it, several additional files

Parameter	Type	Description
fullSummary	logical	Convenience parameter for adding all the diagnostics below
		to the summary.
spectra.detail	logical	Specifies if more detail on the spectra should be added.
almostout	logical	Specifies if "almost" outliers should be added.
rsd.autocorr	character	Vector specifying the type of autocorrelation of the residuals that should be returned, i.e., the autocorrelations and/or partial autocorrelations of the residuals and/or the autocorrelations of the squared residuals ("acf", "pacf", "acf2").
quality.stat	logical	Specifies whether the second Q statistic (Lothian and Morry 1978), i.e., the Q statistic computed w/o the M2 quality control statistic, and the M statistics (Lothian and Morry 1978) for monitoring and quality assessment should be returned as well.
likelihood.stat	logical	If TRUE, the likelihood statistics AIC (Akaike 1973a), AICC (Hurvich and Tsai 1989), BIC (Schwarz 1978) and HQ (Hannan and Quinn 1979) are returned as well as the estimated maximum value of the log likelihood function of the model for the untransformed data.
aape	logical	Specifies whether the average absolute percentage error for forecasts should be returned.
id.rsdseas	logical	Specifies whether the presence/absence of residual seasonality should be indicated.
slidingspans	logical	Defines whether the diagnostics output of the slidingspans analysis should be returned.
history	logical	Defines whether the diagnostics output of the (revision) history analysis should be returned.
identify	logical	Defines whether the (partial) autocorrelations of the residuals generated by the "identify" specification should be returned.
oldOutput	integer	Specifying the number of previous x12() runs stored in the x120ld0utput slot of an 'x12Single' class or an 'x12Batch' class object that should be included in the summary.
print	logical	Specifies if the summary should be printed or only the data frame returned.

Table 5: Parameters of the summary() function for 'x12Single', 'x12Output' and 'x12Batch' objects.

will be created and read into R. If several R processes with the same working directory are running, there is a chance of possible problems due to this implementation. Therefore it is advised to use different working directories for R processes running at the same time and using package  $\mathbf{x}12$ .

# 3.3. Example

The following example briefly illustrates the functionality of R package x12, i.e., an 'x12Batch'

object is created for which several **X-12-ARIMA** parameters are set and the findings are shown by means of a textual summary and several plot functions.

First, the **x12** package and the data are loaded within R and the path for the **X-12-ARIMA** binaries is set.

```
R> library("x12")
R> data("AirPassengers", package = "datasets")
R> x12path(file.path("..", "x12a.exe"))
```

Secondly, a new 'x12Batch' object is generated consisting of four identical time series.

```
R> xb <- new("x12Batch", list(AirPassengers, AirPassengers,
+ AirPassengers, AirPassengers))</pre>
```

The same X-12-ARIMA parameters are set for all four elements.

```
R> xb <- setP(xb, list(estimate = TRUE, outlier.types = "all"))</pre>
```

Different parameters are set for each time series.

```
R> xb \leftarrow setP(xb, list(outlier.critical = list(LS = 3.5, TC = 2.5)), 1)
R> xb \leftarrow setP(xb, list(automdl = FALSE), 2:4)
R> xb \leftarrow setP(xb, list(arima.model = c(0, 1, 1), + arima.smodel = c(0, 1, 1)), 2)
R> xb \leftarrow setP(xb, list(arima.model = c(0, 1, 1), + arima.smodel = c(1, 1, 1)), 3)
R> xb \leftarrow setP(xb, list(arima.model = c(1, 1, 1), + arima.smodel = c(1, 1, 1)), 4)
```

Subsequently, x12() is run on this 'x12Batch' object and the summary is shown for all four series.

```
R> xb <- x12(xb)
R> summary(xb)
```

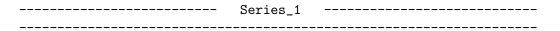
An 'x12Single' object s is extracted from the 'x12Batch' object xb, e.g., the first series.

```
R> s <- xb@x12List[[1]]</pre>
```

The diagnostics summary is called for this series and returned as a data frame.

```
R> sum.out <- summary(s)</pre>
```

It summarizes the output and settings, see:



Time Series

```
Frequency: 12
```

Span: 1st month, 1949 to 12th month, 1960

#### Model Definition

ARIMA Model: (0 1 1)(0 1 1) (Automatic Model Choice)

Transformation: Automatic selection : Log(y)

Regression Model: Automatically Identified Outliers

#### Outlier Detection

Critical |t| for outliers:
accrit1 accrit2 | lscrit | tccrit | "3.89" | "\*" | "3.5" | "2.5"
Total Number of Outliers: 6

Automatically Identified Outliers: 6

#### Regression Model

variable coef stderr tval
1 autooutlier\_TC1951.May 0.078 0.023 3.321
2 autooutlier\_TC1951.Jun -0.099 0.024 -4.174
3 autooutlier\_TC1952.Mar -0.083 0.023 -3.610
4 autooutlier\_LS1953.Jun -0.090 0.023 -3.851
5 autooutlier\_TC1954.Feb -0.075 0.023 -3.243
6 autooutlier\_A01960.Mar -0.104 0.024 -4.270

# Seasonal Adjustment

Identifiable Seasonality: yes
Seasonal Peaks: rsd
Trading Day Peaks: rsd
Overall Index of Quality of SA
(Acceptance Region from 0 to 1)
Q: 0.2
Number of M statistics outside the limits: 0
SA decomposition: multiplicative

Seasonal moving average used for the final iteration:

3x3 (Based on the size of the global moving seasonality ratio (msr))

Moving average used to estimate the final trend-cycle:

9-term Henderson filter

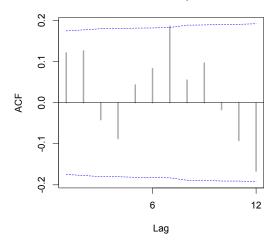
The plot functions mentioned in Section 3.2 can be used as shown below. The results can be viewed in Figures 1a, 1b, 1c and 1d.

```
R> plot(s, trend = TRUE, sa = TRUE, forecast = TRUE)
R> plotRsdAcf(s, which = "acf2")
```

# **Time Series with Forecasts** 700 9 200 400 300 200 8 1950.1 1954.1 1958.1 1962.1 Date Original Seasonally Adjusted Trend

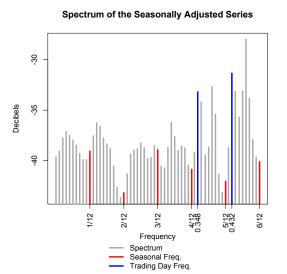
# (a) Output of the plot() method showing trend and forecasts with prediction intervals as well as the seasonally adjusted series.

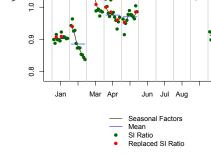
#### **Autocorrelations of the Squared Residuals**



(b) Output of the plotRsdAcf() function from the R package x12, showing the autocorrelations of the squared residuals from the regARIMA model.

Seasonal Factors by period and SI Ratios





- (c) Output of the plotSpec() function, showing the spectrum of the seasonally adjusted series.
- (d) Output of the seasonal factor plot (plotSeasFac()).

Figure 1: Plots obtained with different plot functions called in the example using the AirPassengers data.

.3

7.

R> plotSpec(s)
R> plotSeasFac(s)

Marking and highlighting outliers that have influence on the regression model works as follows:

 $R > s \leftarrow setP(s, list(regression.variables = c("TC1951.5", "TC1954.2",$ 

```
+ "A01960.Mar")))
R> s <- x12(s)
R> plot(s, trend = TRUE, showAllout = TRUE)
```

The corresponding plot showing all automatically detected and manually selected outliers can be found in Figure 3, i.e., additive outliers, level shifts and trend changes.

# 4. The R package x12GUI: Graphical user interface

#### 4.1. Overview

The **GTK**+ toolkit (**GTK**+ Development Team 2012) is used for constructing the graphical user interface. It is implemented in R via the **RGtk2** package (Lawrence and Temple Lang 2014) which supports **GTK**+ versions 2.0 and higher (see Lawrence and Temple Lang 2010 for more detail).

R graphics are embedded in the GUI with the **cairoDevice** package (Lawrence 2014), an R wrapper for the vector based, anti-aliased 2D graphics library **Cairo** (**Cairo** Graphics 2014). **GTK+** and **Cairo** are both licensed under the GNU Lesser General Public License (LGPL).

Handling the **x12** R package via the command line interface can be a challenging task for users without prior knowledge of R. In this context, the R package **x12GUI** provides a graphical user interface (GUI). The GUI offers the following advantages:

- No prior knowledge of R or the R package x12 is required.
- Convenient management of the displayed X-12-ARIMA parameters.
- Dynamic editing, i.e., adding and highlighting of outliers from within the plot window.
- Comparable summaries are provided via point and click.
- Interactive manipulation of the plots.
- On the fly feedback and results.

Basically the whole **x12** functionality can be controlled with the GUI. The function **x12GUI()** can be called on an object of class 'ts', 'x12Single' and 'x12Batch' and it returns the modified version of the object.

```
R> library("x12GUI")
R> x12path(file.path("..", "x12a.exe"))
R> data("AirPassengersX12Batch", package = "x12")
R> x12batch_changed <- x12GUI(AirPassengersX12Batch)</pre>
```

The interface works with a copy of the original data. Thus it does not change the respective object itself but returns the modified version.

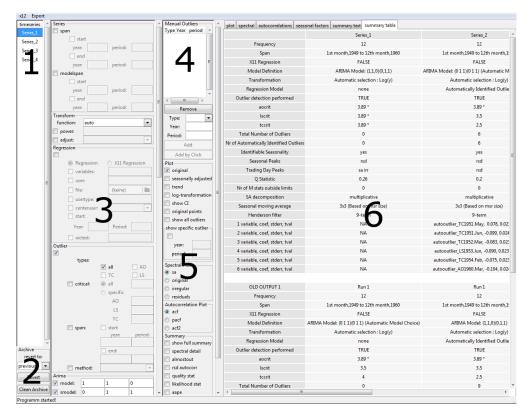


Figure 2: Screenshot of the **x12** GUI after startup.

# 4.2. User interaction within the graphical user interface

Figure 2 shows the  $\mathbf{x}12$  GUI after startup. It consists of six major areas as well as a menu and a status bar. The left half is considered the input region while the right half displays the output.

A description of the six areas tagged in Figure 2 is given in the following:

- (1) Time series selection table lists the time series belonging to either the 'x12Single' or 'x12Batch' object passed to the x12 GUI. One or more of them can be selected for parameter manipulation tasks. Deselecting everything is equivalent to selecting everything. Plots can only be viewed for one object at a time as do the textual summaries.
- (2) Archive panel gives the user access to the archive features of the x12 package like prev() and clearArchive() (see Section 3.2). The list contains all previous runs the object can be reverted to.
- (3) X-12-ARIMA parameter list contains components for editing the X-12-ARIMA parameters. If more than one time series were selected in the time series selection table and some parameter settings differ for these series, these parameters are indicated with an asterisk or something similar.
- (4) Manual outlier editing provides the means to manually add and remove outliers from the regression model. Formally, they are values of the regression.variables argument

Menu item	Description
X12 Update	Initiates an X-12-ARIMA run with the current settings (CTRL+U).
saveParameter	Uses the saveP() function from the x12 package to save the X-12-
	ARIMA parameters to a file. The file location can be specified in a
	dialog window.
loadParameter	Uses the loadP() function from the x12 package to load X-12-ARIMA
	parameters from a file. These parameters can then be assigned to the
	right time series corresponding to an 'x12Single' object.
load and save	Uses standard R functionality to save the current 'x12Batch' object
	to an .RData file or load an .RData file as the current object.
x12path	Displays the currently used path to the binaries of <b>X-12-ARIMA</b> and
	provides the possibility to change it.
Export plot	Exports the currently visible plot to a file in .pdf or .png format.
Export summary	Exports the table version of the summary as a .csv file or to the
	clipboard.

Table 6: Illustration of the menu items.

in the form of "TypeYear.Period" characters but the GUI distinguishes them from the remaining variables. They can be added "by click" as well, referring to the possibility of flagging an outlier in the plot (see Figure 3). This interactive use provides a powerful tool for practitioners who have to seasonally adjust their time series.

- (5) Plot and summary parameter list controls the parameter settings of plots and summaries. Changes here have an immediate effect on the output area (6).
- (6) Output region contains all possible output, i.e., the plots and the diagnostics summaries introduced in Section 3.2. As an additional **x12GUI** feature, outliers can be added to the regression model by right clicking on the respective data point in the time series depicted in the plot window. Each plot can also be saved to a file as .pdf.

The menu also includes possibilities to save and load parameters as well as to export objects, plots and summaries.

The seasonal adjustment becomes interactive by the dynamic selection of outliers, the straightforward changing of parameters within the GUI and the possibility to load any results that have been fitted.

#### 5. Conclusion

The analysis of time dependent data plays a substantial role in application fields like econometrics, natural sciences, environmetrics and official statistics. Improving the quality of the findings and reducing the expenditure of time for adjusting time series seems very worthwhile.

The X-12-ARIMA seasonal adjustment software of the US Census Bureau is profoundly capable in this regard but it comes with the disadvantage of the slightly complicated and inconvenient usage. The input specification files, for example, require thorough knowledge from the user and the output generated is rather difficult to administer and to modify.

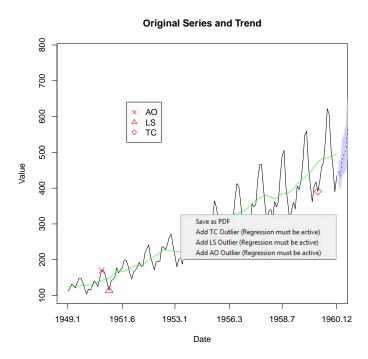


Figure 3: Context menu for manually selecting outliers from within the time series plot. (The regression has to be activated for this series). This plot is displayed directly in the GUI – here we provide the snapshot of the corresponding area that displays the plot.

Using the **x12** R package overcomes these difficulties. It offers the possibility to employ R for preprocessing time series data, for managing **X-12-ARIMA** parameters and output and for presenting diagnostics in a more approachable and accessible manner. In practice, often several thousand of time series have to be adjusted on a regular basis (monthly, quarterly or yearly). Processing time series in an interactive manner to find suitable sets of parameters as well as the application of these parameter settings to a large number of time series is straightforward using the R package **x12**.

In addition to that, the graphical user interface – provided by the  $\mathbf{x12GUI}$  package – enables the manual selection of outliers directly from within the time series plot. Without this GUI, this would normally require several steps in the R  $\mathbf{x12}$  package and even more steps using the original software tool  $\mathbf{X-12-ARIMA}$ . Furthermore, very little prior knowledge of R or the respective packages is required for using the GUI.

# References

Akaike H (1973a). "Information Theory and an Extension of the Maximum Likelihood Principle." In BN Petrov, F Cáski (eds.), Second International Symposium on Information Theory, pp. 267–281. Akademiai Kaidó, Budapest. Reprinted in Breakthroughs in Statistics.

Akaike H (1973b). "Outlier Selection for RegARIMA Models." In BN Petrov, F Cáski (eds.),

- Proceedings of the American Statistical Association, pp. 1–6. American Statistical Association, Alexandria. CD Paper no. 438.
- Baiocchi G, Distaso W (2003). "gretl: Econometric Software for the GNU Generation." *Journal of Applied Econometrics*, **18**(1), 105–110.
- Box GEP, Jenkins GM (1970). Time Series Analysis. Holden-Day, San Francisco.
- Box GEP, Jenkins GM, Reinsel GC (2008). *Time Series Analysis: Forecasting and Control*. Wiley Series in Probability and Statistics. John Wiley & Sons.
- Brockwell PJ, Davis RA (2009). *Time Series: Theory and Methods*. Springer Series in Statistics. Springer-Verlag.
- Cairo Graphics (2014). Cairo Vector Graphics Library, Version 1.14.0. URL http://www.cairographics.org/.
- Chambers JM (2008). Software for Data Analysis: Programming with R. Springer-Verlag, New York.
- Cleveland RB, Cleveland WS, McRae JE, Terpenning I (1990). "STL: A Seasonal-Trend Decomposition Procedure Based on Loess." *Journal of Official Statistics*, **6**(1), 3–73.
- Cottrell A, Lucchetti R (2014). gretl User's Guide Gnu Regression, Econometrics and Time-Series Library. Version 1.9.92, URL http://gretl.sourceforge.net/.
- Doherty M (2001). "Applications: The Surrogate Henderson Filters in **X-11**." Australian & New Zealand Journal of Statistics, **43**(4), 385–392.
- Findley DF, Monsell BC, Bell WR, Otto MC, Chen BC (1998). "New Capabilities and Methods of the **X-12-ARIMA** Seasonal-Adjustment Program." *Journal of Business & Economic Statistics*, **16**(2), 127–152.
- Grudowska S (2012). *Demetra+ User Manual*. URL http://www.cros-portal.eu/content/demetra-user-manual/.
- Grudowska S (2013). **JDemetra+** User Manual. URL http://www.cros-portal.eu/sites/default/files//JDemetra%2B%20User%20Manual.pdf.
- GTK+ Development Team (2012). "GTK+: The Gimp Toolkit." URL http://www.gtk.org/.
- Hannan EJ, Quinn BG (1979). "The Determination of the Order of Autoregression." *Journal* of the Royal Statistical Society B, **41**(2), 190–195.
- Hood CC (2000). "SAS Programs to Get the Most from X-12-ARIMA's Modeling and Seasonal Adjustment Diagnostics." In *Proceedings of the 25th Annual SAS Users Group International Conference (SUGI 25)*.
- Hood CCH, Monsell B (2010). Getting Started with X-12-ARIMA, Using the Command Prompt on Your PC. URL http://www.catherinechhood.net/papers/gsx12input.pdf.
- Hurvich CM, Tsai CL (1989). "Regression and Time Series Model Selection in Small Samples." Biometrika, **76**(2), 297–307.

- Hyndman RJ (2014). "CRAN Task View: Time Series Analysis." Version 2014-08-26, URL http://CRAN.R-project.org/view=TimeSeries.
- Hyndman RJ, Athanasopoulos G, Razbash S, Schmidt D, Zhou Z, Khan Y, Bergmeir C (2014). *forecast:* Forecasting Functions for Time Series and Linear Models. R package version 5.6, URL http://CRAN.R-project.org/package=forecast.
- Hyndman RJ, Khandakar Y (2008). "Automatic Time Series Forecasting: The **forecast** Package for R." *Journal of Statistical Software*, **27**(3), 1–22. URL http://www.jstatsoft.org/v27/i03/.
- IHS Inc (2013). EViews 8 for Windows. Irvine. URL http://www.eviews.com/.
- Kendall MG, Alan S (1983). The Advanced Theory of Statistics, Volume 3. Griffin.
- Kowarik A, Meraner A (2014). x12: X12 Wrapper Function and Structure for Batch Processing. R package version 1.6.0, URL http://CRAN.R-project.org/package=x12.
- Lawrence M (2014). cairoDevice: Cairo-Based Cross-Platform Antialiased Graphics Device Driver. R package version 2.20, URL http://CRAN.R-project.org/package=cairoDevice.
- Lawrence M, Temple Lang D (2010). "RGtk2: A Graphical User Interface Toolkit for R." Journal of Statistical Software, 37, 52. URL http://www.jstatsoft.org/v37/i08/.
- Lawrence M, Temple Lang D (2014). "RGtk2: R Bindings for GTK 2.8.0 and Above." R package version 2.20.31, URL http://CRAN.R-project.org/package=RGtk2.
- Livera AMD, Hyndman RJ, Snyder RD (2011). "Forecasting Time Series with Complex Seasonal Patterns Using Exponential Smoothing." *Journal of the American Statistical Association*, **106**(496), 1513–1527.
- Ljung GM, Box GEP (1978). "On a Measure of Lack of Fit in Time Series Models." *Biometrika*, **65**(2), 297–303.
- Lothian J, Morry M (1978). A Set of Quality Control Statistics for the **X-11-Arima** Seasonal Adjustment Method. Research Papers. Seasonal Adjustment and Time Series Analysis Staff, Statistics Canada.
- Maravall A (2003). Notes on Programs TRAMO and SEATS: Part I. URL http://www.bde.es/f/webbde/SES/servicio/software/tramo/Part\_I\_Intro\_TS.pdf.
- Pfaff B (2008). "VAR, SVAR and SVEC Models: Implementation within R Package vars." Journal of Statistical Software, 27(4), 1–32. URL http://www.jstatsoft.org/v27/i04/.
- R Core Team (2014). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.
- Schopfhauser D, Kowarik A, Meraner A (2014). x12GUI: X12 Graphical User Interface. R package version 0.13.0, URL http://CRAN.R-project.org/package=x12GUI.
- Schwarz G (1978). "Estimating the Dimension of a Model." The Annals of Statistics,  $\mathbf{6}(2)$ , 461-464.

- Shiskin J, Young AH, Musgrave JC (1967). "The X-ll Variant of the Census Method II Seasonal Adjustment Program." *Technical Report 15*.
- Spider Financial Corp (2009). *NumXL*: *Numerical Analysis for Excel*. URL http://www.spiderfinancial.com/products/numxl/.
- Templ M (2014). "CRAN Task View: Official Statistics & Survey Methodology." Version 2014-08-18, URL http://CRAN.R-project.org/view=OfficialStatistics.
- Time Series Research Staff Statistical Research Division (2011). *X-12-ARIMA* Reference Manual. US Census Bureau. URL http://www.census.gov/ts/x12a/v03/x12adocV03.pdf.
- US Census Bureau (2011). Win X-12 Windows Interface for X-12-ARIMA. URL https://www.census.gov/srd/www/winx12/.
- US Census Bureau (2013). **X-12-ARIMA** Seasonal Adjustment Program. URL https://www.census.gov/srd/www/x12a/.
- US Census Bureau (2014). **X-13ARIMA-SEATS** Seasonal Adjustment Program. URL https://www.census.gov/srd/www/x13as/.
- Zeileis A, Grothendieck G (2005). "zoo: S3 Infrastructure for Regular and Irregular Time Series." Journal of Statistical Software, 14(6), 1–27. URL http://www.jstatsoft.org/v14/i06/.

#### **Affiliation:**

Alexander Kowarik Methods Unit Statistics Austria 1110 Vienna, Austria

E-mail: alexander.kowarik@statistik.gv.at

Journal of Statistical Software published by the American Statistical Association Volume 62, Issue 2

 Volume 62, Issue 2
 Submitted: 2013-07-30

 November 2014
 Accepted: 2014-07-02

http://www.jstatsoft.org/

http://www.amstat.org/