

seasonal: R interface to X-13ARIMA-SEATS

Christoph Sax*

August 5, 2014

1 Introduction

seasonal is an easy-to-use and full-featured R-interface to X-13ARIMA-SEATS, the newest seasonal adjustment software developed by the [United States Census Bureau](#). X-13ARIMA-SEATS combines and extends the capabilities of the older X-12ARIMA (developed by the Census Bureau) and TRAMO-SEATS (developed by the Bank of Spain).

If you are new to seasonal adjustment or X-13ARIMA-SEATS, the automated procedures of *seasonal* allow you to quickly produce good seasonal adjustments of time series. Start with the [Installation](#) and [Getting started](#) section and skip the rest. Alternatively, `demo(seas)` gives an overview of the package functionality.

If you are familiar with X-13ARIMA-SEATS, you may benefit from the flexible input and output structure of *seasonal*. The package allows you to use (almost) all commands of X-13ARIMA-SEATS, and it can import (almost) all output generated by X-13ARIMA-SEATS. The only exception is the ‘composite’ spec, which is not supported. Read the [Input](#) and [Output](#) sections and have a look at the [wiki](#), where the examples from the official X-13ARIMA-SEATS [manual](#) are reproduced in R.

2 Installation

To install the latest development version with support for the **HTML version of X-13** and a new **Shiny-based graphical user interface** directly from Github, type to the R console:

```
install.packages("devtools")
devtools::install_github('seasonal', 'christophsax')
```

The stable version (0.6) is available from CRAN:

```
install.packages("seasonal")
```

seasonal does not include the binary executables of X-13ARIMA-SEATS. They can be obtained precompiled from [here](#) (Windows: `x13ashtmlall.zip`). A guide on how to build it from source for Ubuntu or Mac OS-X can be found [here](#). My own compilation for Mac OS-X can be obtained [upon request](#).

Download the file, unzip it and copy `x13ashtml.exe` (or `x13ashtml`, on a Unix system) to the desired location in your file system. Next, you need to tell *seasonal* where to find the binary executables of X-13ARIMA-SEATS, by setting the specific environmental variable `X13_PATH`. This may be done during your active session in R:

```
Sys.setenv(X13_PATH = "YOUR_X13_DIRECTORY")
```

Exchange `YOUR_X13_DIRECTORY` with the path to your installation of X-13ARIMA-SEATS. Note that the Windows path `C:\something\somewhere` has to be entered UNIX-like `C:/something/somewhere` or `C:\\something\\somewhere`. You can always check your installation with:

*State Secretariat of Economic Affairs, Switzerland

```
checkX13()
```

If you want to set the environmental variable permanently, you may do so by adding it to the `Renviron.site` file, which is located in the `etc` subdirectory of your R home directory (use `R.home()` in R to reveal the home directory). `Renviron.site` does not exist by default; if not, you have to create a file named `Renviron.site` with your favorite text editor (on Windows, be careful with the ‘show extensions for known file types’ option, the extension `.site` may be hidden). Add the following line to the file (without quotes!):

```
X13_PATH = YOUR_PATH_TO_X13
```

Alternatively, use the system terminal (on Windows, it’s called command prompt; also, the `cd` command requires `\` instead of `/`):

```
cd YOUR_R_HOME_DIRECTORY/etc
echo X13_PATH = YOUR_PATH_TO_X13 >> Renviron.site
```

There are other ways to set an environmental variable permanently in R, see `?Startup`.

3 Getting started

`seas` is the core function of the *seasonal* package. By default, `seas` calls the automatic procedures of X-13ARIMA-SEATS to perform a seasonal adjustment that works well in most circumstances:

```
m <- seas(AirPassengers)
```

The first argument of `seas` has to be a time series of class `"ts"`. The function returns an object of class `"seas"` that contains all necessary information on the adjustment.

There are several functions and methods for `"seas"` objects: The `final` function returns the adjusted series, the `plot` method shows a plot with the unadjusted and the adjusted series. The `summary` method allows you to display an overview of the model:

```
final(m)
plot(m)
summary(m)
```

By default, `seas` calls the SEATS adjustment procedure. If you prefer the X11 adjustment procedure, use the following option (see the [Input](#) section for details on how to use arbitrary options with X-13):

```
seas(AirPassengers, x11 = "")
```

A default call to `seas` also invokes the following automatic procedures of X-13ARIMA-SEATS:

- Transformation selection (log / no log)
- Detection of trading day and Easter effects
- Outlier detection
- ARIMA model search

Alternatively, all inputs may be entered manually, as in the following example:

```
seas(x = AirPassengers,
     regression.variables = c("td1coef", "easter[1]", "ao1951.May"),
     arima.model = "(0 1 1)(0 1 1)",
     regression.aictest = NULL,
     outlier = NULL,
     transform.function = "log")
```

The `static` command returns the manual call of a model. The call above can be easily generated from the automatic model:

```
static(m)
static(m, coef = TRUE) # also fixes the coefficients
```

If you have *Shiny* installed, the `inspect` command offers a way to analyze and modify a seasonal adjustment procedure (see the section below for details):

```
inspect(m)
```

4 Input

In *seasonal*, it is possible to use almost the complete syntax of X-13ARIMA-SEATS. This is done via the `...` argument in the `seas` function. The X-13ARIMA-SEATS syntax uses *specs* and *arguments*, with each spec optionally containing some arguments. These spec-argument combinations can be added to `seas` by separating the spec and the argument by a dot (`.`). For example, in order to set the ‘variables’ argument of the ‘regression’ spec equal to `td` and `ao1999.jan`, the input to `seas` looks like this:

```
m <- seas(AirPassengers, regression.variables = c("td", "ao1955.jan"))
```

Note that R vectors may be used as an input. If a spec is added without any arguments, the spec should be set equal to an empty string (or, alternatively, to an empty list). Several defaults of `seas` are empty strings, such as the default `seats = ""`. See the help page (`?seas`) for more details on the defaults. Note the difference between `""` (meaning the spec is enabled but has no arguments) and `NULL` (meaning the spec is disabled).

It is possible to manipulate almost all inputs to X-13ARIMA-SEATS in this way. For instance, example 1 in section 7.1 from the [manual](#),

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

translates to R in the following way:

```
seas(AirPassengers,
     x11 = ""),
     arima.model = "(0 1 1)"
)
```

`seas` takes care of the ‘series’ spec, and no input beside the time series has to be provided. As `seas` uses the SEATS procedure by default, the use of X11 has to be specified manually. When the ‘x11’ spec is added as an input (like above), the mutually exclusive and default ‘seats’ spec is automatically disabled. With `arima.model`, an additional spec-argument is added to the input of X-13ARIMA-SEATS. As the spec cannot be used in the same call as the ‘automdl’ spec, the latter is automatically disabled. The best way to learn about the relationship between the syntax of X-13ARIMA-SEATS and *seasonal* is to study the comprehensive list of examples in the [wiki](#).

There are some mutually exclusive specs in X-13ARIMA-SEATS. If more than one mutually exclusive spec is included in `seas`, specs are overwritten according the following priority rules:

- Model selection
 1. `arima`
 2. `pickmdl`
 3. `automdl` (default)
- Adjustment procedure
 1. `x11`
 2. `seats` (default)

5 Output

seasonal has a flexible mechanism to read data from X-13ARIMA-SEATS. With the `series` function, it is possible to import almost all output that can be generated by X-13ARIMA-SEATS. For example, the following command returns the forecasts of the ARIMA model as a "ts" time series:

```
m <- seas(AirPassengers)
series(m, "forecast.forecasts")
```

Because the `forecast.save = "forecasts"` argument has not been specified in the model call, `series` re-evaluates the call with the 'forecast' spec enabled. It is also possible to return more than one output table at the same time:

```
series(m, c("forecast.forecasts", "d1"))
```

You can use either the unique short names of X-13 (such as `d1`), or the the long names (such as `forecasts`). Because the long table names are not unique, they need to be combined with the spec name (`forecast`). See `?series` for a complete list of options.

Note that re-evaluation doubles the overall computation time. If you want to speed it up, you have to be explicit about the output in the model call:

```
m <- seas(AirPassengers, forecast.save = "forecasts")
series(m, "forecast.forecasts")
```

Some specs, like 'slidingspans' and 'history', are time consuming. Re-evaluation allows you to separate these specs from the basic model call:

```
m <- seas(AirPassengers)
series(m, "history.saestimates")
series(m, "slidingspans.sfspans")
```

If you are using the HTML version of X-13, the `out` function shows the content of the main output in the browser:

```
out(m)
```

6 Graphs

There are several graphical tools to analyze a `seas` model. The main plot function draws the seasonally adjusted and unadjusted series, as well as the outliers. Optionally, it also draws the trend of the seasonal decomposition:

```
m <- seas(AirPassengers, regression.aictest = c("td", "easter"))
plot(m)
plot(m, outliers = FALSE)
plot(m, trend = TRUE)
```

The `monthplot` function allows for a monthwise plot (or quarterwise, with the same function name) of the seasonal and the SI component:

```
monthplot(m)
monthplot(m, choice = "irregular")
```

Also, many standard R function can be used to analyze a "seas" model:

```
pacf(resid(m))
spectrum(diff(resid(m)))
plot(density(resid(m)))
qqnorm(resid(m))
```

The `identify` method can be used to select or deselect outliers by point and click. Click several times to loop through different outlier types.

```
identify(m)
```

7 Inspect tool

The `inspect` function is a graphical tool for choosing a seasonal adjustment model. Since seasonal 0.62, it uses *Shiny* and can now be used without RStudio. To install the latest version of Shiny, type:

```
install.packages("shiny")
```

The goal of `inspect` is to summarize all relevant options, plots and statistics that should be usually considered. `inspect` uses a `"seas"` object as its only argument:

```
inspect(m)
```

The `inspect` function opens an interactive window that allows for the manipulation of a number of arguments. It offers several views to analyze the series graphically. With each change, the adjustment process and the visualizations are recalculated. Summary statistics are shown in the first tab. The last tab offers access to all series that can be produced with X-13. The views in `inspect` are also customizable, see the examples in `?inspect`.

8 License

seasonal is free and open source, licensed under GPL-3. It has been developed for the use at the Swiss State Secretariat of Economic Affairs. Its development is not connected to the development of X-13ARIMA-SEATS ([license](#)).

Please report bugs and suggestions on [Github](#) or send me an [e-mail](#). Thank you!