Package: seasonal (via r-universe)

October 21, 2024

Type Package

Title R Interface to X-13-ARIMA-SEATS

Version 1.10.0

Description Easy-to-use interface to X-13-ARIMA-SEATS, the seasonal
adjustment software by the US Census Bureau. It offers full
access to almost all options and outputs of X-13, including
X-11 and SEATS, automatic ARIMA model search, outlier detection
and support for user defined holiday variables, such as Chinese
New Year or Indian Diwali. A graphical user interface can be
used through the 'seasonalview' package. Uses the X-13-binaries
from the 'x13binary' package.

Depends R (>= 2.15)

Imports x13binary

Suggests seasonalview (>= 0.1.3), testthat (>= 3.0.0), covr, knitr,
rmarkdown, spelling

License GPL-3

URL http://www.seasonal.website

BugReports https://github.com/christophsax/seasonal/issues

LazyData true

RoxygenNote 7.3.2

Encoding UTF-8

Config/testthat/edition 3

VignetteBuilder knitr

Language en-US

NeedsCompilation no

Author Christoph Sax [aut, cre]
(<https://orcid.org/0000-0002-7192-7044>), Dirk Eddelbuettel
[ctb] (<https://orcid.org/0000-0001-6419-907X>), Andrea Ranzato
[ctb]

Maintainer Christoph Sax <christoph.sax@gmail.com>

Contents

---

seasonal-package	*seasonal: R interface to X-13ARIMA-SEATS*

Description

seasonal is an easy-to-use interface to X-13-ARIMA-SEATS, the seasonal adjustment software by the US Census Bureau. It offers full access to almost all options and outputs of X-13, including X-11 and SEATS, automatic ARIMA model search, outlier detection and support for user defined holiday variables, such as Chinese New Year or Indian Diwali. A graphical user interface can be used through the **seasonalview** package. Uses the X-13-binaries from the **x13binary** package.

The best way to start is to have a look at the vignette:

```
vignette("seas")
```

Installation

Seasonal depends on the **x13binary** package, which downloads and installs the X-13 binaries. To install both packages, simply type to the R console:

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

A startup message is given if the path to X-13 is specified manually. To suppress the message, use [suppressPackageStartupMessages()](#).

Setting the X-13 path manually

Sometimes, you either cannot or don't want to rely on the binaries provided by **x13binary**:

- because you are on an unsupported system, like Solaris. If you manage to build X-13 on such a system, please let the developers of **x13binary** know.

- because you cannot run executable files in your R library folders, due to corporate IT policy.

- because you are using your own Fortran compilation of X-13ARIMA-SEATS.

Setting the path manually can be done as in previous versions of seasonal. In order to tell seasonal where to find the binary executables of X-13ARIMA- SEATS, the specific environmental variable X13_PATH needs to be set. 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. You can always check your installation with:

```
checkX13()
```

If it works, you may want to set the environmental variable permanently, by adding the Sys.setenv line to one of your `.Rprofile` files. The easiest is to use the one located in your home directory, which can be written directly from R:

```
write('Sys.setenv(X13_PATH = "YOUR_X13_DIRECTORY")', file = "~/.Rprofile", append = TRUE)
```

If the file does not exist (by default), it will be created. Make sure that you get the quotes right: double quotes around your directory, single quotes around the whole Sys.setenv line, such that R understands your string. Check first that the the Sys.setenv line works correctly; once it is written you may have to edit .Rprofile manually. (Or add a second, overwriting line to it.) For other ways to set an environmental variable permanently in R, see [Startup()](#).

Author(s)

Christoph Sax <christoph.sax@gmail.com>

References

Sax C, Eddelbuettel D (2018). "Seasonal Adjustment by X-13ARIMA-SEATS in R." *Journal of Statistical Software*, *87*(11), 1-17. doi:10.18637/jss.v087.i11.

See Also

seas() for the core function and more information on package usage.

---

 as.data.frame.seas   *Coerce Output to* data.frame

Description

These methods coerce the output to a data.frame. This is useful for further processing. (This is a second attempt to do that; the first experimental attempt in version 1.4 used an argument data.frame in the call to the functions, and is now obsolete. The present approach seems cleaner and is likely to stay, but still consider it as **experimental**.)

Usage

```
## S3 method for class 'seas'
as.data.frame(x, ...)

## S3 method for class 'summary.seas'
as.data.frame(x, ...)
```

Arguments

x	an object of class "seas" or "summary.seas", usually, the result of a call to the functions with the same name.
...	unused.

Details

The data.frames produced by these functions follow the naming conventions from the 'broom' package, but do not depend on it otherwise.

Value

a data.frame without row names.

Examples

```
m <- seas(AirPassengers, x11 = "")

# a data.frame containing data
as.data.frame(m)

# a data.frame containing the summary information on the coefficients
as.data.frame(summary(m))
```

---

checkX13   *Check Installation of X-13ARIMA-SEATS*

Description

Check the installation of the binary executables of X-13ARIMA-SEATS. See [seasonal()](#) for details on how to set X13_PATH manually if you intend to use your own binaries.

Usage

```
checkX13(fail = FALSE, fullcheck = TRUE, htmlcheck = TRUE)
```

Arguments

fail   logical, whether an error should interrupt the process. If FALSE, a message is
              returned.

fullcheck   logical, whether a full test should be performed. Runs Testairline.spc (which
              is shiped with X-13ARIMA-SEATS) to test the working of the binaries. Returns
              a message.

htmlcheck   logical, whether the presence of the the HTML version of X-13 should be checked.

Examples

```
old.path <- Sys.getenv("X13_PATH")
Sys.setenv(X13_PATH = "")  # its broken now
try(checkX13())

# fix it (provided it worked in the first place)
if (old.path == "") {
  Sys.unsetenv("X13_PATH")
} else {
  Sys.setenv(X13_PATH = old.path)
}
try(checkX13())
```

---

cpi   *Consumer Price Index of Switzerland*

Description

Monthly consumer price index of Switzerland. Base year is 1993.

Format

Time series of class `"ts"`.

Source

Federal Statistical Office, Switzerland

Examples

```
data(seasonal)
cpi
```

---

easter   *Dates of Chinese New Year, Indian Diwali and Easter*

Description

Dates of Chinese New Year, Indian Diwali and Easter, suitable for the use in genhol().

Format

Objects of class `"Date"`.

Source

<https://www2.census.gov/software/x-13arima-seats/win-genhol/download/>

<http://www.chinesenewyears.info/chinese-new-year-calendar.php>

Ministry of Statistics and Programme Implementation, with help from Pinaki Mukherjee

Examples

```
data(holiday)
cny
diwali
easter
```

---

exp   *Exports and Imports of China*

Description

Monthly exports and imports of China (July 1983 to December 2013).

Format

Each time series is an object of class "ts".

Details

In 100 mio U.S. Dollar.

Source

China Customs

Examples

```
data(seasonal)
imp
exp
```

---

final   *Time Series of a Seasonal Adjustment Model*

Description

Functions to extract the main time series from a "seas" object. For universal import of X-13ARIMA-SEATS tables, use the [series()](#) function.

Usage

```
final(object)

original(object)

trend(object)

irregular(object)

## S3 method for class 'seas'
residuals(object, ...)
```

Arguments

object   an object of class `"seas"`.

...   not used. For compatibility with the generic.

Details

These functions support R default NA handling. If `na.action = na.exclude` is specified in the call to `seas`, the time series will also contain NAs.

Value

returns a `"ts"` object, depending on the function.

References

Vignette with a more detailed description: <http://www.seasonal.website/seasonal.html>

Comprehensive list of R examples from the X-13ARIMA-SEATS manual: http://www.seasonal.website/examples.html

Official X-13ARIMA-SEATS manual: https://www2.census.gov/software/x-13arima-seats/x13as/windows/documentation/docx13as.pdf

See Also

[seas()](#) for the main function of seasonal.

[series()](#), for universal X-13 output extraction.

Examples

```
m <- seas(AirPassengers)

final(m)
original(m)
irregular(m)
trend(m)

# NA handling
AirPassengersNA <- window(AirPassengers, end = 1962, extend = TRUE)
final(seas(AirPassengersNA, na.action = na.omit))    # no NA in final series
final(seas(AirPassengersNA, na.action = na.exclude)) # NA in final series
final(seas(AirPassengersNA, na.action = na.x13))     # NA filled by x13
# final(seas(AirPassengersNA, na.action = na.fail))    # fails
```

---

fivebestmdl   *Five Best ARIMA Models*

Description

Returns the five best models as chosen by the BIC criterion. It needs the `automdl` spec to be activated (default). If it is not activated, the function tries to re-evaluate the model with the `automdl` spec activated.

Usage

```
fivebestmdl(x)
```

Arguments

x   object of class `"seas"`

References

Vignette with a more detailed description: `http://www.seasonal.website/seasonal.html`

Comprehensive list of R examples from the X-13ARIMA-SEATS manual: `http://www.seasonal.website/examples.html`

Official X-13ARIMA-SEATS manual: `https://www2.census.gov/software/x-13arima-seats/x13as/windows/documentation/docx13as.pdf`

See Also

`seas()` for the main function.

`series()`, for universal X-13 output extraction.

`plot.seas()`, for diagnostical plots.

`out()`, for accessing the full output of X-13ARIMA-SEATS.

Examples

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

---

genhol   *Generate Holiday Regression Variables*

Description

A replacement for the genhol software by the U.S. Census Bureau, a utility that uses the same
procedure as X-12-ARIMA to create regressors for the U. S. holidays of Easter, Labor Day, and
Thanksgiving. **This is a replacement written in R, the U.S. Census Bureau software is not
needed.**

Usage

```
genhol(x, start = 0, end = 0, frequency = 12, center = "none")
```

Arguments

x	a vector of class "Date", containing the occurrences of the holiday. It can be generated with as.Date().
start	integer, shifts the start point of the holiday. Use negative values if start is before the specified date.
end	integer, shifts end point of the holiday. Use negative values if end is before the specified date.
frequency	integer, frequency of the resulting series
center	character string. Either "calendar", "mean" or "none" (default). Centering avoids a bias in the resulting series. Use "calendar" for Easter or Chinese New Year, "mean" for Ramadan. See references: Notes on centering holiday.

Details

The resulting time series can be used as a user defined variable in seas(). Usually, you want the
holiday effect to be removed from the final series, so you need to specify regression.usertype =
"holiday". (The default is to include user defined variables in the final series.)

Value

an object of class "ts" that can be used as a user defined variable in seas().

See Also

seas() for the main function of seasonal.

Examples

```
data(holiday)  # dates of Chinese New Year, Indian Diwali and Easter

### use of genhol

# 10 day before Easter day to one day after, quarterly data:
genhol(easter, start = -10, end = 1, frequency = 4)
genhol(easter, frequency = 2)  # easter is always in the first half-year

# centering for overall mean or monthly calendar means
genhol(easter, center = "mean")
genhol(easter, center = "calendar")

### replicating X-13's built-in Easter adjustment

# built-in
m1 <- 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", x11 = "")
summary(m1)

# user defined variable
ea1 <- genhol(easter, start = -1, end = -1, center = "calendar")

# regression.usertype = "holiday" ensures that the effect is removed from
# the final series.
m2 <- seas(x = AirPassengers,
           regression.variables = c("td1coef", "ao1951.May"),
           xreg = ea1, regression.usertype = "holiday",
           arima.model = "(0 1 1)(0 1 1)", regression.aictest = NULL,
           outlier = NULL, transform.function = "log", x11 = "")
summary(m2)

all.equal(final(m2), final(m1), tolerance = 1e-06)


# with genhol, its possible to do sligtly better, by adjusting the length
# of easter from Friday to Monday:

ea2 <- genhol(easter, start = -2, end = +1, center = "calendar")
m3 <- seas(x = AirPassengers,
           regression.variables = c("td1coef", "ao1951.May"),
           xreg = ea2, regression.usertype = "holiday",
           arima.model = "(0 1 1)(0 1 1)", regression.aictest = NULL,
           outlier = NULL, transform.function = "log", x11 = "")
summary(m3)


### Chinese New Year
```

```
data(seasonal)
data(holiday)  # dates of Chinese New Year, Indian Diwali and Easter

# de facto holiday length: http://en.wikipedia.org/wiki/Chinese_New_Year
cny.ts <- genhol(cny, start = 0, end = 6, center = "calendar")

m1 <- seas(x = imp, xreg = cny.ts, regression.usertype = "holiday", x11 = "",
           regression.variables = c("td1coef", "ls1985.Jan", "ls2008.Nov"),
           arima.model = "(0 1 2)(0 1 1)", regression.aictest = NULL,
           outlier = NULL, transform.function = "log")
summary(m1)

# compare to identical no-CNY model
m2 <- seas(x = imp, x11 = "",
           regression.variables = c("td1coef", "ls1985.Jan", "ls2008.Nov"),
           arima.model = "(0 1 2)(0 1 1)", regression.aictest = NULL,
           outlier = NULL, transform.function = "log")
summary(m2)

ts.plot(final(m1), final(m2), col = c("red", "black"))

# modeling complex holiday effects in Chinese imports
# - positive pre-CNY effect
# - negative post-CNY effect
pre_cny <- genhol(cny, start = -6, end = -1, frequency = 12, center = "calendar")
post_cny <- genhol(cny, start = 0, end = 6, frequency = 12, center = "calendar")
m3 <- seas(x = imp, x11 = "",
           xreg = cbind(pre_cny, post_cny), regression.usertype = "holiday",
           x11 = list())
summary(m3)


### Indian Diwali (thanks to Pinaki Mukherjee)

# adjusting Indian industrial production
m4 <- seas(iip,
x11 = "",
xreg = genhol(diwali, start = 0, end = 0, center = "calendar"),
regression.usertype = "holiday"
)
summary(m4)

# without specification of 'regression.usertype', Diwali effects are added
# back to the final series
m5 <- seas(iip,
x11 = "",
xreg = genhol(diwali, start = 0, end = 0, center = "calendar")
)

ts.plot(final(m4), final(m5), col = c("red", "black"))

# plot the Diwali factor in Indian industrial production
```

```
plot(series(m4, "regression.holiday"))


### Using genhol to replicate the regARIMA estimation in R

# easter regressor
ea <- genhol(easter, start = -1, end = -1, center = "calendar")
ea <- window(ea, start = start(AirPassengers), end = end(AirPassengers))

# estimating ARIMA model in R base
arima(log(AirPassengers), order = c(0,1,1), seasonal = c(0,1,1), xreg = ea)

summary(seas(AirPassengers, regression.variables = c("easter[1]"),
             regression.aictest = NULL))

# Note that R defines the ARIMA model with negative signs before the MA term,
# X-13 with a positive sign.
```

---

identify.seas   *Manually Identify Outliers*

Description

Select or deselect outliers by point and click. To quit and return the call, press ESC. Click several times to loop through different outlier types.

Usage

```
## S3 method for class 'seas'
identify(x, type = c("ao", "tc", "ls"), ...)
```

Arguments

x	an object of class "seas".
type	character vector, types of outlier to loop through.
...	unused, for compatibility with the generic function.

Value

an object of class "seas", containing the static call of the selected model.

Examples

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

---

iip   *Industrial Production of India*

Description

Industrial Production of India (IIP).

Format

Time series of class `"ts"`.

Details

Index value. IIP is used for measuring the performance overall industrial sector of the Indian economy. IIP is compiled by using data from 16 source agencies.

Source

Central Statistics Office of the Ministry of Statistics and Programme Implementation, with help from Pinaki Mukherjee

Examples

```
data(seasonal)
iip
```

---

import.spc   *Import X-13* `.spc` *Files*

Description

Utility function to import `.spc` files from X-13. It generates a list of calls to `seas` (and `import.ts`) that can be run in R. Evaluating these calls should perform the same X-13 procedure as the original `.spc` file. The `print` method displays the calls in a way that they can be copy-pasted into an R script.

Usage

```
import.spc(file, text = NULL)

## S3 method for class 'import.spc'
print(x, ...)
```

Arguments

`file`	character, path to the X-13 `.spc` file
`text`	character, alternatively, the content of a `.spc` file as a character string.
`x`	object of class `import.spc`
`...`	further arguments, not used

Value

returns an object of class `import.spc`, which is a list with the following (optional) objects of class `call`:

`x`	the call to retrieve the data for the input series
`xtrans`	the call to retrieve the data for the `xtrans` series (if required by the call)
`xreg`	the call to retrieve the data for the `xreg` series (if required by the call)
`seas`	the call to `seas()`

See Also

`import.ts()`, for importing X-13 data files.

`seas()` for the main function of seasonal.

Examples

```
# importing the orginal X-13 example file
import.spc(text =
'
  series{
    title="International Airline Passengers Data from Box and Jenkins"
    start=1949.01
    data=(
    112 118 132 129 121 135 148 148 136 119 104 118
    115 126 141 135 125 149 170 170 158 133 114 140
    145 150 178 163 172 178 199 199 184 162 146 166
    171 180 193 181 183 218 230 242 209 191 172 194
    196 196 236 235 229 243 264 272 237 211 180 201
    204 188 235 227 234 264 302 293 259 229 203 229
    242 233 267 269 270 315 364 347 312 274 237 278
    284 277 317 313 318 374 413 405 355 306 271 306
    315 301 356 348 355 422 465 467 404 347 305 336
    340 318 362 348 363 435 491 505 404 359 310 337
    360 342 406 396 420 472 548 559 463 407 362 405
    417 391 419 461 472 535 622 606 508 461 390 432)
    span=(1952.01, )
  }
  spectrum{
    savelog=peaks
  }
  transform{
    function=auto
```

```
      savelog=autotransform
    }
    regression{
      aictest=(td easter)
      savelog=aictest
    }
    automdl{
      savelog=automodel
    }
    outlier{ }
    x11{}
'
)



### reading .spc with multiple user regression and transformation series

# running a complex seas call and save output in a temporary directory
tdir <- tempdir()
seas(x = AirPassengers, xreg = cbind(a = genhol(cny, start = 1, end = 4,
    center = "calendar"), b = genhol(cny, start = -3, end = 0,
    center = "calendar")), xtrans = cbind(sqrt(AirPassengers), AirPassengers^3),
    transform.function = "log", transform.type = "temporary",
    regression.aictest = "td", regression.usertype = "holiday", dir = tdir,
    out = TRUE)

# importing the .spc file from the temporary location
ll <- import.spc(file.path(tdir, "iofile.spc"))

# ll is list containing four calls:
# - 'll$x', 'll$xreg' and 'll$xtrans': calls to import.ts(), which read the
#   series from the X-13 data files
# - 'll$seas': a call to seas() which performs the seasonal adjustment in R
str(ll)

# to replicate the original X-13 operation, run all four calls in a series.
# You can either copy/paste and run the print() output:
ll

# or use eval() to evaluate the call(s). To evaluate the first call and
# import the x variable:
eval(ll$x)

# to run all four calls in 'll', use lapply() and eval():
ee <- lapply(ll, eval, envir = globalenv())
ee$seas  # the 'seas' object, produced by the final call to seas()
```

---

import.ts   *Import Time Series from X-13 Data Files*

Description

Utility function to read time series from X-13 data files. A call to `import.ts` is constructed and included in the output of `import.spc()`.

Usage

```
import.ts(
  file,
  format = "datevalue",
  start = NULL,
  frequency = NULL,
  name = NULL
)
```

Arguments

file	character, name of the X-13 file which the data are to be read from
format	a valid X-13 file format as described in 7.15 of the X-13 manual: `"datevalue"`, `"datevaluecomma"`, `"free"`, `"freecomma"`, `"x13save"`, `"tramo"` or an X-11 or Fortran format.
start	vector of length 2, time of the first observation (only for formats `"free"` and `"freecomma"` and the Fortran formats.)
frequency	the number of observations per unit of time (only for formats `"free"`, `"freecomma"` and the X-11 or Fortran formats.)
name	(X-11 formats only) name of the series, to select from a file with multiple time series. Omit if you want to read all time series from an X-11 format file.

Value

an object of class `ts` or `mts`

See Also

`import.spc()`, for importing X-13 `.spc` files.

`seas()` for the main function of seasonal.

Examples

```
tdir <- tempdir()
seas(x = AirPassengers, dir = tdir)
import.ts(file.path(tdir, "iofile.dta"))
import.ts(file.path(tdir, "iofile.rsd"), format = "x13save")
```

---

na.x13   *Handle Missing Values by X-13*

Description

Utility function to substitute NA values by -99999. Useful as a value for the na.action argument
in seas().

Usage

```
na.x13(x)
```

Arguments

x   an object of class "ts"

Value

a time series, with NA values substituted by -99999.

Examples

```
AirPassengersNA <- AirPassengers
AirPassengersNA[20] <- NA
na.x13(AirPassengersNA)

seas(AirPassengersNA, na.action = na.x13)
```

---

out   *Display X-13ARIMA-SEATS Output*

Description

The out function shows the full content of the X-13ARIMA-SEATS output in the browser. If you
want to use a specific statistic in R, the udg() function is preferable.

Usage

```
out(x, browser = getOption("browser"), ...)
```

Arguments

x   an object of class "seas".

browser   browser to be used, passed on to browseURL().

...   additional spec-arguments options sent to X-13ARIMA-SEATS during re-evaluation,
                       passed to update().

Details

To keep the size of "seas" objects small, seas does not save the output by default. Instead, out re-evaluates the model.

Value

displays the output as a side effect.

References

Vignette with a more detailed description: http://www.seasonal.website/seasonal.html

Comprehensive list of R examples from the X-13ARIMA-SEATS manual: http://www.seasonal.website/examples.html

Official X-13ARIMA-SEATS manual: https://www2.census.gov/software/x-13arima-seats/x13as/windows/documentation/docx13as.pdf

See Also

seas() for the main function of seasonal.

Examples

```
## Not run:
m <- seas(AirPassengers)
out(m)
# customizing the output with additional elements
out(m, automdl.print = "autochoicemdl")


## End(Not run)
```

---

outlier   *Outlier Time series*

Description

Returns an object of class "ts" that contains the names of the outliers.

Usage

```
outlier(x, full = FALSE)
```

Arguments

x	an object of class "seas".
full	logical, should the full label of the outlier be shown? If FALSE, only the type of the outlier is shown.

Value

character string time series with outliers.

Examples

```
x <- seas(AirPassengers)
outlier(x)
```

---

plot.seas   *Seasonal Adjustment Plots*

Description

Functions to graphically analyze a "seas" object.

Usage

```
## S3 method for class 'seas'
plot(
  x,
  outliers = TRUE,
  trend = FALSE,
  main = "Original and Adjusted Series",
  xlab = "Time",
  ylab = "",
  transform = c("none", "PC", "PCY"),
  ...
)

residplot(
  x,
  outliers = TRUE,
  main = "residuals of regARIMA",
  xlab = "Time",
  ylab = "",
  ...
)

## S3 method for class 'seas'
monthplot(x, choice = c("seasonal", "irregular"), main, ...)
```

Arguments

x	an object of class "seas", usually, a result of a call to [seas()](#).
outliers	logical, should the outliers be drawn.
trend	logical, should the trend be drawn.

main	character string, title of the graph.
xlab	character string, title for the x axis.
ylab	character string, title for the y axis.
transform	character string, optionally transform the data to period to period ″PC″ or year to year″PCY″ percentage change rates.
...	further arguments passed to the plotting functions.
choice	character string, ″seasonal″ (default) or ″irregular″.

Details

plot calls the plot method for class ″seas″. It plots the adjusted and unadjusted series, as well as the outliers. Optionally draws the trend series.

residplot plots the residuals and the outliers.

monthplot calls the monthplot method for class ″seas″. It plot the seasonal and SI component periodwise. Despite its name, monthplot can be used for series of all frequencies.

Value

All plot functions return a plot as their side effect.

References

Vignette with a more detailed description: http://www.seasonal.website/seasonal.html

Comprehensive list of R examples from the X-13ARIMA-SEATS manual: http://www.seasonal.website/examples.html

Official X-13ARIMA-SEATS manual: https://www2.census.gov/software/x-13arima-seats/x13as/windows/documentation/docx13as.pdf

See Also

seas(), for the main function.

udg(), for diagnostical statistics.

Examples

```
m <- seas(AirPassengers)

plot(m)
plot(m, outliers = FALSE)
plot(m, trend = TRUE)

residplot(m)
residplot(m, outliers = FALSE)

monthplot(m)
```

```
# use standard R functions to analyze "seas" models
pacf(resid(m))
spectrum(diff(resid(m)))
plot(density(resid(m)))
qqnorm(resid(m))
```

---

predict.seas   *Seasonal Adjusted Series*

Description

Returns the seasonally adjusted series of an (optionally re-evaluated) model of class "seas". Without further arguments, this is equivalent to a call to the [final()](#) function.

Usage

```
## S3 method for class 'seas'
predict(object, newdata, ...)
```

Arguments

object	an object of class "seas".
newdata	an object of class "ts". new data values for the x argument in the [seas()](#) function.
...	further arguments, passed to [update.seas()](#), to re-evaluate the model.

Details

With the newdata argument supplied, the "seas" object is re- evaluated, using the original model call. This is equivalent of calling final(update(m, x = newdata)).

Value

Object of class "ts".

Examples

```
# Using data from Dec. 59 to estimate a model
ap.short <- window(AirPassengers, end = c(1959, 12))
m <- seas(ap.short)
predict(m)
final(m)      # equivalent

# Use Dec. 59 model specification to estimate data up to Dec. 60
predict(m, AirPassengers)
```

---

seas   *Seasonal Adjustment with X-13ARIMA-SEATS*

Description

Main function of the seasonal package. With the default options, seas calls the automatic pro-
cedures of X-13ARIMA-SEATS to perform a seasonal adjustment that works well in most cir-
cumstances. Via the ... argument, it is possible to invoke almost all options that are available in
X-13ARIMA-SEATS (see details). The default options of seas are listed as explicit arguments and
are discussed in the arguments section. A full-featured graphical user interface can be accessed by
the [view()](#) function.

Usage

```
seas(
  x = NULL,
  xreg = NULL,
  xtrans = NULL,
  seats.noadmiss = "yes",
  transform.function = "auto",
  regression.aictest = c("td", "easter"),
  outlier = "",
  automdl = "",
  composite = NULL,
  na.action = na.omit,
  out = FALSE,
  dir = NULL,
  multimode = c("x13", "R"),
  ...,
  list = NULL
)
```

Arguments

x	object of class "ts" or "mts", or a list of "ts"' objects: time series to seasonally adjust.
xreg	(optional) object of class "ts": one or several user defined exogenous variables for regARIMA modeling, can be used both with regression or x11regression.
xtrans	(optional) object of class "ts": one or two user defined exogenous variables for the transform spec. Can be specifed together with xreg.
seats.noadmiss	spec 'seats' with argument noadmiss = "yes" (default). Seasonal adjustment by SEATS, if SEATS decomposition is invalid, an alternative model is used (a message is returned). If noadmiss = "no", no approximation is done. If the seats spec is removed (seats = NULL), no seasonal adjustment is performed.

transform.function

> spec transform with argument function = "auto" (default). Automatic log transformation detection. Set equal to "none", "log" or any value that is allowed by X-13 to turn it off.

regression.aictest

> spec regression with argument aictest = c("td", "easter") (default). AIC test for trading days and Easter effects. Set equal to NULL to turn it off.

outlier

> spec outlier without arguments (default). Automatic outlier detection. Set equal to NULL to turn it off.

automdl

> spec automdl without arguments (default). Automatic model search with the automdl spec. Set equal to NULL to turn it off.

composite

> spec composite. A named list with spec-arguments for the aggregation of multiple series. Also requries series.comtype = "add" or similar. Set equal to NULL to turn it off (default). See vignette("multiple").

na.action

> a function which indicates what should happen when the data contain NAs. na.omit (default), na.exclude or na.fail. If na.action = na.x13, NA handling is done by X-13, i.e. NA values are substituted by -99999.

out

> logical. Should the X-13ARIMA-SEATS standard output be saved in the "seas" object? (this increases object size substantially, it is recommended to re-evaluate the model using the [out()](#) function instead.)

dir

> character string with a user defined file path. If specified, the X-13ARIMA-SEATS output files are copied to this folder. Useful for debugging.

multimode

> one of "x13" or "R". When multiple series are supplied, should they be processed in a single call ("x13") or processed individually ("R"). See vignette("multiple").

...

> additional spec-arguments options sent to X-13ARIMA-SEATS (see details).

list

> a named list with additional spec-arguments options. This is an alternative to the ... argument. It is useful for programming.

Details

It is possible to use the almost complete syntax of X-13ARIMA-SEAT via the ... argument. The syntax of X-13ARIMA-SEATS uses *specs* and *arguments*, and each spec optionally contains some arguments. In seas, an additional spec-argument can be added by separating spec and argument by a dot (.) (see examples). Alternatively, spec-argument combinations can be supplied as a named list, which is useful for programming.

Similarly, the [series()](#) function can be used to read almost all series from X-13ARIMA-SEATS. The [udg()](#) function provides access to a large number of diagnostical statistics.

For a more extensive description, consider vignette("seas") or the wiki page, which contains replications of almost all examples from the official X-13ARIMA-SEATS manual.

Value

returns an object of class "seas", essentially a list with the following components:

series

> a list containing the output tables of X-13. To be accessed by the series function.

data	seasonally adjusted data, the raw data, the trend component, the irregular component and the seasonal component (deprecated).
err	warning messages from X-13ARIMA-SEATS
udg	content of the .udg output file
est	content of the .est output file
model	list with the model specification, similar to "spc". It typically contains "regression", which contains the regressors and parameter estimates, and "arima", which contains the ARIMA specification and the parameter estimates.
fivebestmdl	Best Five ARIMA Models (unparsed)
x	input series
spc	object of class "spclist", a list containing the content of the .spc file that is used by X-13ARIMA-SEATS. Each spec is on the first level, each argument is on the second level.
call	function call
wdir	temporary directory in which X-13ARIMA-SEATS has been run

The final function returns the final adjusted series, the plot method shows a plot with the unadjusted and the adjusted series. summary gives an overview of the regARIMA model. The [udg()](udg()) function returns diagnostical statistics.

References

Sax C, Eddelbuettel D (2018). "Seasonal Adjustment by X-13ARIMA-SEATS in R." *Journal of Statistical Software*, *87*(11), 1-17. [doi:10.18637/jss.v087.i11](doi:10.18637/jss.v087.i11).

On-Line Interface to seasonal http://www.seasonal.website

Comprehensive list of R examples from the X-13ARIMA-SEATS manual: http://www.seasonal.website/examples.html

Official X-13ARIMA-SEATS manual: https://www2.census.gov/software/x-13arima-seats/x13as/windows/documentation/docx13as.pdf

See Also

view(), for accessing the graphical user interface.

update.seas(), to update an existing "seas" model.

static(), to return the 'static' call, with automated procedures substituted by their choices.

series(), for universal X-13 table series import.

[out()](out()), to view the full X-13 diagnostical output.

Examples

```
# Basic call

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

```
# Graphical user interface
## Not run:
view(m)

## End(Not run)


# invoke X-13ARIMA-SEATS options as 'spec.argument' through the ... argument
# (consult the X-13ARIMA-SEATS manual for many more options and the list of
# R examples for more examples)
seas(AirPassengers, regression.aictest = c("td"))  # no easter testing
seas(AirPassengers, force.type = "denton")  # force equality of annual values
seas(AirPassengers, x11 = "")  # use x11, overrides the 'seats' spec

# 'spec.argument' combinations can also be supplied as a named list, which is
# useful for programming
seas(AirPassengers, list = list(regression.aictest = c("td"), outlier = NULL))
# constructing the list step by step
ll <- list()
ll[["x"]] <- AirPassengers
ll[["regression.aictest"]] <- "td"
ll["outlier"] <- list(NULL)  # assigning NULL to a list using single brackets
seas(list = ll)

# options can be entered as vectors
seas(AirPassengers, regression.variables = c("td1coef", "easter[1]"))
seas(AirPassengers, arima.model = c(0, 1, 1, 0, 1, 1))
seas(AirPassengers, arima.model = "(0 1 1)(0 1 1)")      # equivalent

# turn off the automatic procedures
seas(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")

# static replication of 'm <- seas(AirPassengers)'
static(m)  # this also tests the equivalence of the static call
static(m, test = FALSE)  # no testing (much faster)
static(m, coef = TRUE)  # also fixes the coefficients

# updating an existing model
update(m, x11 = "")

# specific extractor functions
final(m)
predict(m)    # equivalent
original(m)
resid(m)
coef(m)
fivebestmdl(m)
spc(m)                  # the .spc input file to X-13 (for debugging)

# universal extractor function for any X-13ARIMA-SEATS output (see ?series)
```

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

# user defined regressors (see ?genhol for more examples)
# a temporary level shift in R base
tls <- ts(0, start = 1949, end = 1965, freq = 12)
window(tls, start = c(1955, 1), end = c(1957, 12)) <- 1
seas(AirPassengers, xreg = tls, outlier = NULL)
# identical to a X-13ARIMA-SEATS specification of the the level shift
seas(AirPassengers, regression.variables = c("tl1955.01-1957.12"),
     outlier = NULL)

# forecasting an annual series without seasonal adjustment
m <- seas(airmiles, seats = NULL, regression.aictest = NULL)
series(m, "forecast.forecasts")

# NA handling
AirPassengersNA <- window(AirPassengers, end = 1962, extend = TRUE)
final(seas(AirPassengersNA, na.action = na.omit))    # no NA in final series
final(seas(AirPassengersNA, na.action = na.exclude)) # NA in final series
# final(seas(AirPassengersNA, na.action = na.fail))    # fails

# NA handling by X-13 (works with internal NAs)
AirPassengersNA[20] <- NA
final(seas(AirPassengersNA, na.action = na.x13))

## performing 'composite' adjustment
seas(
  cbind(mdeaths, fdeaths),
  composite = list(),
  series.comptype = "add"
)
```

---

series	*Import X-13ARIMA-SEATS Output Tables*

Description

The series function imports all tables that can be saved in X-13ARIMA-SEATS.

Usage

```
series(x, series, reeval = TRUE, verbose = TRUE)
```

Arguments

x   an object of class "seas".

series   character vector, short or long names of an X-13ARIMA-SEATS table. If a long
           name is specified, it needs to be combined with the spec name and separated by
           a dot (it is not unique, otherwise. See list below.). More than one series can be
           specified (see examples).

| reeval | logical, if TRUE, the model is re-evaluated with the corresponding specs enabled. |
| verbose | logical, if TRUE, a message is returned if a spec is added during reevaluation. |

Details

If the save argument is not specified in the model call, series re-evaluates the call with the corresponding specs enabled (also returning a message). Note that re-evaluation doubles the overall computational time. If you want to accelerate the procedure, you have to be explicit about the output in the model call (see examples).

List of all importable tables from X-13ARIMA-SEATS:

spec	long name	short name	description
check	check.acf	acf	autocorrelation function of residuals with standard errors
check	check.acfsquared	ac2	autocorrelation function of squared residuals with standar
check	check.pacf	pcf	partial autocorrelation function of residuals with standard
composite	composite.adjcompositesrs	b1	aggregated time series data, prior adjusted, with associate
composite	composite.calendaradjcomposite	cac	aggregated time series data, adjusted for regARIMA calen
composite	composite.compositesrs	cms	aggregated time series data, with associated dates
composite	composite.indadjsatot	iaa	final indirect seasonally adjusted series, with yearly totals
composite	composite.indadjustfac	iaf	final combined adjustment factors for the indirect seasona
composite	composite.indaoutlier	iao	final indirect AO outliers
composite	composite.indcalendar	ica	final calendar factors for the indirect seasonal adjustment
composite	composite.indirregular	iir	final irregular component for the indirect adjustment
composite	composite.indlevelshift	ils	final indirect LS outliers
composite	composite.indmcdmovavg	if1	MCD moving average of the final indirect seasonally adju
composite	composite.indmodirr	ie3	irregular component modified for extreme values from the
composite	composite.indmodoriginal	ie1	original series modified for extreme values from the indire
composite	composite.indmodsadj	ie2	seasonally adjusted series modified for extreme values fro
composite	composite.indreplacsi	id9	final replacement values for extreme SI-ratios (differences
composite	composite.indrevsachanges	i6a	percent changes for indirect seasonally adjusted series wit
composite	composite.indrndsachanges	i6r	percent changes (differences) in the rounded indirect seas
composite	composite.indrobustsa	iee	final indirect seasonally adjusted series modified for extre
composite	composite.indsachanges	ie6	percent changes (differences) in the indirect seasonally ad
composite	composite.indsadjround	irn	rounded indirect seasonally adjusted series
composite	composite.indseasadj	isa	final indirect seasonally adjusted series
composite	composite.indseasonal	isf	final seasonal factors for the indirect seasonal adjustment
composite	composite.indseasonaldiff	isd	final seasonal difference for the indirect seasonal adjustme
composite	composite.indtotaladjustment	ita	total indirect adjustment factors (only produced if the orig
composite	composite.indtrend	itn	final trend-cycle for the indirect adjustment
composite	composite.indtrendchanges	ie7	percent changes (differences) in the indirect final trend co
composite	composite.indunmodsi	id8	final unmodified SI-ratios (differences) for the indirect adj
composite	composite.origchanges	ie5	percent changes (differences) in the original series
composite	composite.outlieradjcomposite	oac	aggregated time series data, adjusted for outliers.
composite	composite.prioradjcomposite	ia3	composite series adjusted for user-defined prior adjustmer
estimate	estimate.armacmatrix	acm	correlation matrix of ARMA parameter estimates if used
estimate	estimate.iterations	itr	detailed output for estimation iterations, including log-like
estimate	estimate.regcmatrix	rcm	correlation matrix of regression parameter estimates if use
estimate	estimate.regressioneffects	ref	Xb matrix of regression variables multiplied by the vector

estimate	estimate.residuals	rsd	model residuals with associated dates or observation num
estimate	estimate.roots	rts	roots of the autoregressive and moving average operators
force	force.forcefactor	ffc	factors applied to get seasonally adjusted series with cons
force	force.revsachanges	e6a	percent changes (differences) in seasonally adjusted series
force	force.revsachangespct	p6a	percent changes in seasonally adjusted series with forced
force	force.rndsachanges	e6r	percent changes (differences) in rounded seasonally adjus
force	force.rndsachangespct	p6r	percent changes in rounded seasonally adjusted series
force	force.saround	rnd	rounded final seasonally adjusted series (if round = yes) o
force	force.seasadjtot	saa	final seasonally adjusted series with constrained yearly to
forecast	forecast.backcasts	bct	point backcasts on the original scale, along with upper an
forecast	forecast.forecasts	fct	point forecasts on the original scale, along with upper and
forecast	forecast.transformed	ftr	forecasts on the transformed scale, with corresponding fo
forecast	forecast.transformedbcst	btr	backcasts on the transformed scale, with corresponding fo
forecast	forecast.variances	fvr	forecast error variances on the transformed scale, showing
history	history.armahistory	amh	history of estimated AR and MA coefficients from the reg
history	history.chngestimates	che	concurrent and most recent estimate of the month-tomont
history	history.chngrevisions	chr	revision from concurrent to most recent estimate of the m
history	history.fcsterrors	fce	revision history of the accumulated sum of squared foreca
history	history.fcsthistory	fch	listing of the forecast and forecast errors used to generate
history	history.indsaestimates	iae	concurrent and most recent estimate of the indirect seaso
history	history.indsarevisions	iar	revision from concurrent to most recent estimate of the in
history	history.lkhdhistory	lkh	history of AICC and likelihood values
history	history.outlierhistory	rot	record of outliers removed and kept for the revisions histo
history	history.saestimates	sae	concurrent and most recent estimate of the seasonally adj
history	history.sarevisions	sar	revision from concurrent to most recent estimate of the se
history	history.seatsmdlhistory	smh	SEATS ARIMA model history
history	history.sfestimates	sfe	concurrent and most recent estimate of the seasonal facto
history	history.sfilterhistory	sfh	record of seasonal filter selection for each observation in t
history	history.sfrevisions	sfr	revision from concurrent to most recent estimate of the se
history	history.tdhistory	tdh	history of estimated trading day regression coefficients fro
history	history.trendchngestimates	tce	concurrent and most recent estimate of the month-tomont
history	history.trendchngrevisions	tcr	revision from concurrent to most recent estimate of the m
history	history.trendestimates	tre	concurrent and most recent estimate of the trend compone
history	history.trendrevisions	trr	revision from concurrent to most recent estimate of the tre
identify	identify.acf	iac	sample autocorrelation function(s), with standard errors a
identify	identify.pacf	ipc	sample partial autocorrelation function(s) with standard e
outlier	outlier.finaltests	fts	t-statistics for every time point and outlier type generated
outlier	outlier.iterations	oit	detailed results for each iteration of outlier detection inclu
regression	regression.aoutlier	ao	regARIMA additive (or point) outlier factors (table A8.A
regression	regression.holiday	hol	regARIMA holiday factors (table A7)
regression	regression.levelshift	ls	regARIMA level shift, temporary level shift and ramp out
regression	regression.outlier	otl	combined regARIMA outlier factors (table A8)
regression	regression.regressionmatrix	rmx	values of regression variables with associated dates
regression	regression.regseasonal	a10	regARIMA user-defined seasonal factors (table A10)
regression	regression.seasonaloutlier	so	regARIMA seasonal outlier factors (table A8.SO)
regression	regression.temporarychange	tc	regARIMA temporary change outlier factors (table A8.TC
regression	regression.tradingday	td	regARIMA trading day factors (table A6)
regression	regression.transitory	a13	regARIMA transitory component factors from userdefined

regression	regression.userdef	usr	factors from user-defined regression variables (table A9)
seats	seats.adjustfac	s16	final SEATS combined adjustment factors
seats	seats.adjustfacpct	psa	combined adjustment factors, expressed as percentages if
seats	seats.adjustmentratio	s18	final SEATS adjustment ratio
seats	seats.componentmodels	mdc	models for the components
seats	seats.cycle	cyc	cycle component
seats	seats.difforiginal	dor	fully differenced transformed original series
seats	seats.diffseasonaladj	dsa	fully differenced transformed SEATS seasonal adjustmen
seats	seats.difftrend	dtr	fully differenced transformed SEATS trend
seats	seats.filtersaconc	fac	concurrent finite seasonal adjustment filter
seats	seats.filtersasym	faf	symmetric finite seasonal adjustment filter
seats	seats.filtertrendconc	ftc	concurrent finite trend filter
seats	seats.filtertrendsym	ftf	symmetric finite trend filter
seats	seats.irregular	s13	final SEATS irregular component
seats	seats.irregularoutlieradj	se3	final SEATS irregular component, outlier adjusted
seats	seats.irregularpct	psi	final irregular component, expressed as percentages if app
seats	seats.longtermtrend	ltt	long term trend
seats	seats.pseudoinnovsadj	pia	pseudo-innovations of the final SEATS seasonal adjustme
seats	seats.pseudoinnovseasonal	pis	pseudo-innovations of the seasonal component
seats	seats.pseudoinnovtransitory	pit	pseudo-innovations of the transitory component
seats	seats.pseudoinnovtrend	pic	pseudo-innovations of the trend component
seats	seats.seasadjconst	sec	final SEATS seasonal adjustment with constant term inclu
seats	seats.seasonal	s10	final SEATS seasonal component
seats	seats.seasonaladj	s11	final SEATS seasonal adjustment
seats	seats.seasonaladjfcstdecomp	afd	forecast of the final SEATS seasonal adjustment
seats	seats.seasonaladjoutlieradj	se2	final SEATS seasonal adjustment, outlier adjusted
seats	seats.seasonaladjse	ase	standard error of final seasonally adjusted series
seats	seats.seasonalfcstdecomp	sfd	forecast of the seasonal component
seats	seats.seasonalpct	pss	final seasonal factors, expressed as percentages if appropr
seats	seats.seasonalse	sse	standard error of final steasonal component
seats	seats.seasonalsum	ssm	seasonal-period-length sums of final SEATS seasonal con
seats	seats.seriesfcstdecomp	ofd	forecast of the series component
seats	seats.squaredgainsaconc	gac	squared gain for finite concurrent seasonal adjustment filt
seats	seats.squaredgainsasym	gaf	squared gain for finite symmetric seasonal adjustment filt
seats	seats.squaredgaintrendconc	gtc	squared gain for finite concurrent trend filter
seats	seats.squaredgaintrendsym	gtf	squared gain for finite symmetric trend filter
seats	seats.timeshiftsaconc	tac	time shift for finite concurrent seasonal adjustment filter
seats	seats.timeshifttrendconc	ttc	time shift for finite concurrent trend filter
seats	seats.totaladjustment	sta	total adjustment factors for SEATS seasonal adjustment
seats	seats.transitory	s14	final SEATS transitory component
seats	seats.transitoryfcstdecomp	yfd	forecast of the transitory component
seats	seats.transitorypct	psc	final transitory component, expressed as percentages if ap
seats	seats.transitoryse	cse	standard error of final transitory component
seats	seats.trend	s12	final SEATS trend component
seats	seats.trendadjls	stl	level shift adjusted trend
seats	seats.trendconst	stc	final SEATS trend component with constant term included
seats	seats.trendfcstdecomp	tfd	forecast of the trend component
seats	seats.trendse	tse	standard error of final trend component

seats	seats.wkendfilter	wkf	end filters of the semi-infinite Wiener-Kolmogorov filter
series	series.adjoriginal	b1	original series, adjusted for prior effects and forecast exte
series	series.calendaradjorig	a18	original series adjusted for regARIMA calendar effects
series	series.outlieradjorig	a19	original series adjusted for regARIMA outliers
series	series.seriesmvadj	mv	original series with missing values replaced by regARIMA
series	series.span	a1	time series data, with associated dates (if the span argume
slidingspans	slidingspans.chngspans	chs	month-to-month (or quarter-to-quarter) changes from all s
slidingspans	slidingspans.indchngspans	cis	indirect month-to-month (or quarter-to-quarter) changes f
slidingspans	slidingspans.indsaspans	ais	indirect seasonally adjusted series from all sliding spans
slidingspans	slidingspans.indsfspans	sis	indirect seasonal factors from all sliding spans
slidingspans	slidingspans.indychngspans	yis	indirect year-to-year changes from all sliding spans
slidingspans	slidingspans.sfspans	sfs	seasonal factors from all sliding spans
slidingspans	slidingspans.tdspans	tds	trading day factors from all sliding spans
slidingspans	slidingspans.ychngspans	ycs	year-to-year changes from all sliding spans
spectrum	spectrum.speccomposite	is0	spectral plot of first-differenced aggregate series
spectrum	spectrum.specextresiduals	ser	spectrum of the extended residuals
spectrum	spectrum.specindirr	is2	spectral plot of outlier-modified irregular series from the i
spectrum	spectrum.specindsa	is1	spectral plot of the first-differenced indirect seasonally ad
spectrum	spectrum.specirr	sp2	spectral plot of outlier-modified X-11 irregular series
spectrum	spectrum.specorig	sp0	spectral plot of the first-differenced original series
spectrum	spectrum.specresidual	spr	spectral plot of the regARIMA model residuals
spectrum	spectrum.specsa	sp1	spectral plot of differenced, X-11 seasonally adjusted seri
spectrum	spectrum.specseatsirr	s2s	spectrum of the final SEATS irregular
spectrum	spectrum.specseatssa	s1s	spectrum of the differenced final SEATS seasonal adjustm
spectrum	spectrum.spectukeycomposite	it0	Tukey spectrum of the first-differenced aggregate series
spectrum	spectrum.spectukeyextresiduals	ter	Tukey spectrum of the extended residuals
spectrum	spectrum.spectukeyindirr	it2	Tukey spectrum of the outlier-modified irregular series fro
spectrum	spectrum.spectukeyindsa	it1	Tukey spectrum of the first-differenced indirect seasonally
spectrum	spectrum.spectukeyirr	st2	Tukey spectrum of the outlier-modified X-11 irregular ser
spectrum	spectrum.spectukeyorig	st0	Tukey spectrum of the first-differenced original series
spectrum	spectrum.spectukeyresidual	str	Tukey spectrum of the regARIMA model residuals
spectrum	spectrum.spectukeysa	st1	Tukey spectrum of the differenced, X-11 seasonally adjus
spectrum	spectrum.spectukeyseatsirr	t2s	Tukey spectrum of the final SEATS irregular
spectrum	spectrum.spectukeyseatssa	t1s	Tukey spectrum of the differenced final SEATS seasonal a
transform	transform.permprior	a2p	permanent prior adjustment factors, with associated dates
transform	transform.permprioradjusted	a3p	prior adjusted series using only permanent prior factors, v
transform	transform.permprioradjustedptd	a4p	prior adjusted series using only permanent prior factors ar
transform	transform.prior	a2	prior adjustment factors, with associated dates
transform	transform.prioradjusted	a3	prior adjusted series, with associated dates
transform	transform.prioradjustedptd	a4d	prior adjusted series (including prior trading day adjustme
transform	transform.seriesconstant	a1c	original series with value from the constant argument add
transform	transform.tempprior	a2t	temporary prior adjustment factors, with associated dates
transform	transform.transformed	trn	prior adjusted and transformed data, with associated dates
x11	x11.adjoriginalc	c1	original series modified for outliers, trading day and prior
x11	x11.adjoriginald	d1	original series modified for outliers, trading day and prior
x11	x11.adjustdiff	fad	final adjustment difference (only for pseudo-additive seas
x11	x11.adjustfac	d16	combined seasonal and trading day factors
x11	x11.adjustfacpct	paf	combined adjustment factors, expressed as percentages if

x11	x11.adjustmentratio	e18	final adjustment ratios (original series/seasonally adjusted
x11	x11.biasfactor	bcf	bias correction factors
x11	x11.calendar	d18	combined holiday and trading day factors
x11	x11.calendaradjchanges	e8	percent changes (differences) in original series adjusted fo
x11	x11.calendaradjchangespct	pe8	percent changes in original series adjusted for calendar fa
x11	x11.combholiday	chl	combined holiday prior adjustment factors, A16 table
x11	x11.extreme	c20	extreme values, C iteration
x11	x11.extremeb	b20	extreme values, B iteration
x11	x11.irregular	d13	final irregular component
x11	x11.irregularadjao	ira	final irregular component adjusted for point outliers
x11	x11.irregularb	b13	irregular component, B iteration
x11	x11.irregularc	c13	irregular component, C iteration
x11	x11.irregularpct	pir	final irregular component, expressed as percentages if app
x11	x11.irrwt	c17	final weights for the irregular component
x11	x11.irrwtb	b17	preliminary weights for the irregular component
x11	x11.mcdmovavg	f1	MCD moving average of the final seasonally adjusted seri
x11	x11.modirregular	e3	irregular component modified for zero-weighted extreme
x11	x11.modoriginal	e1	original series modified for zero-weighted extreme values
x11	x11.modseasadj	e2	seasonally adjusted series modified for zero-weighted extr
x11	x11.modsic4	c4	modified SI-ratios (differences), C iteration
x11	x11.modsid4	d4	modified SI-ratios (differences), D iteration
x11	x11.origchanges	e5	percent changes (differences) in original series
x11	x11.origchangespct	pe5	percent changes in the original series
x11	x11.replacsi	d9	final replacement values for extreme SI-ratios (differences
x11	x11.replacsic9	c9	modified SI-ratios (differences), C iteration
x11	x11.robustsa	e11	robust final seasonally adjusted series
x11	x11.sachanges	e6	percent changes (differences) in seasonally adjusted series
x11	x11.sachangespct	pe6	percent changes in seasonally adjusted series
x11	x11.seasadj	d11	final seasonally adjusted series
x11	x11.seasadjb11	b11	seasonally adjusted series, B iteration
x11	x11.seasadjb6	b6	preliminary seasonally adjusted series, B iteration
x11	x11.seasadjc11	c11	seasonally adjusted series, C iteration
x11	x11.seasadjc6	c6	preliminary seasonally adjusted series, C iteration
x11	x11.seasadjconst	sac	final seasonally adjusted series with constant from transfo
x11	x11.seasadjd6	d6	preliminary seasonally adjusted series, D iteration
x11	x11.seasonal	d10	final seasonal factors
x11	x11.seasonaladjregsea	ars	seasonal factors adjusted for user-defined seasonal regAR
x11	x11.seasonalb10	b10	seasonal factors, B iteration
x11	x11.seasonalb5	b5	preliminary seasonal factors, B iteration
x11	x11.seasonalc10	c10	preliminary seasonal factors, C iteration
x11	x11.seasonalc5	c5	preliminary seasonal factors, C iteration
x11	x11.seasonald5	d5	preliminary seasonal factors, D iteration
x11	x11.seasonaldiff	fsd	final seasonal difference (only for pseudo-additive season
x11	x11.seasonalpct	psf	final seasonal factors, expressed as percentages if appropr
x11	x11.sib3	b3	preliminary unmodified SI-ratios (differences)
x11	x11.sib8	b8	unmodified SI-ratios (differences)
x11	x11.tdadjorig	c19	original series adjusted for final trading day
x11	x11.tdadjorigb	b19	original series adjusted for preliminary trading day

x11	x11.totaladjustment	tad	total adjustment factors (only printed out if the original se
x11	x11.trend	d12	final trend-cycle
x11	x11.trendadjls	tal	final trend-cycle adjusted for level shift outliers
x11	x11.trendb2	b2	preliminary trend-cycle, B iteration
x11	x11.trendb7	b7	preliminary trend-cycle, B iteration
x11	x11.trendc2	c2	preliminary trend-cycle, C iteration
x11	x11.trendc7	c7	preliminary trend-cycle, C iteration
x11	x11.trendchanges	e7	percent changes (differences) in final trend component ser
x11	x11.trendchangespct	pe7	percent changes in final trend cycle
x11	x11.trendconst	tac	final trend component with constant from transform spec
x11	x11.trendd2	d2	preliminary trend-cycle, D iteration
x11	x11.trendd7	d7	preliminary trend-cycle, D iteration
x11	x11.unmodsi	d8	final unmodified SI-ratios (differences)
x11	x11.unmodsiox	d8b	final unmodified SI-ratios, with labels for outliers and ext
x11	x11.yrtotals	e4	ratio of yearly totals of original and seasonally adjusted se
x11regression	x11regression.calendar	xca	final calendar factors (trading day and holiday)
x11regression	x11regression.calendarb	bxc	preliminary calendar factors
x11regression	x11regression.combcalendar	xcc	final calendar factors from combined daily weights
x11regression	x11regression.combcalendarb	bcc	preliminary calendar factors from combined daily weights
x11regression	x11regression.combtradingday	c18	final trading day factors from combined daily weights
x11regression	x11regression.combtradingdayb	b18	preliminary trading day factors from combined daily weig
x11regression	x11regression.extremeval	c14	irregulars excluded from the irregular regression, C iterati
x11regression	x11regression.extremevalb	b14	irregulars excluded from the irregular regression, B iterati
x11regression	x11regression.holiday	xhl	final holiday factors
x11regression	x11regression.holidayb	bxh	preliminary holiday factors
x11regression	x11regression.outlieriter	xoi	detailed results for each iteration of outlier detection inclu
x11regression	x11regression.priortd	a4	prior trading day weights and factors
x11regression	x11regression.tradingday	c16	final trading day factors and weights
x11regression	x11regression.tradingdayb	b16	preliminary trading day factors and weights
x11regression	x11regression.x11reg	c15	final irregular regression coefficients and diagnostics
x11regression	x11regression.x11regb	b15	preliminary irregular regression coefficients and diagnosti
x11regression	x11regression.xregressioncmatrix	xrc	correlation matrix of irregular regression parameter estim
x11regression	x11regression.xregressionmatrix	xrm	values of irregular regression variables with associated da

Value

depending on the table, either an object of class `"ts"` or `"data.frame"`.

References

Vignette with a more detailed description: <http://www.seasonal.website/seasonal.html>

Comprehensive list of R examples from the X-13ARIMA-SEATS manual: http://www.seasonal.website/examples.html

Official X-13ARIMA-SEATS manual: https://www2.census.gov/software/x-13arima-seats/x13as/windows/documentation/docx13as.pdf

See Also

seas() for the main function.

Examples

```
m <- seas(AirPassengers)
series(m, "fct")  # re-evaluate with the forecast spec activated

# more than one series
series(m, c("rsd", "fct"))

m <- seas(AirPassengers, forecast.save = "fct")
series(m, "fct") # no re-evaluation (much faster!)

# using long names
series(m, "forecast.forecasts")

# history spec
series(m, "history.trendestimates")
series(m, "history.sfestimates")
series(m, "history.saestimates")
series(m, c("history.sfestimates", "history.trendestimates"))

# slidingspans spec
series(m, "slidingspans.sfspans")
series(m, "slidingspans.ychngspans")

# fundamental identities of seasonal adjustment
# Y = T * I * (S * TD)
all.equal(AirPassengers, series(m, "seats.trend") *
          series(m, "seats.irregular") * series(m, "seats.adjustfac"))
# Y_sa = Y / (S * TD)
all.equal(final(m), AirPassengers / series(m, "seats.adjustfac"))

### Some X-13ARIMA-SEATS functions can be replicated in R:

# X-13ARIMA-SEATS spectrum
plot(series(m, "spectrum.specorig")[,-1], t = "l")
# R equivalent: spectrum from stats
spectrum(diff(log(AirPassengers)), method = "ar")

# X-13ARIMA-SEATS pacf
x13.pacf <- series(m, "identify.pacf")
plot(x13.pacf[,1], t = "h")
lines(x13.pacf[,2])
lines(-x13.pacf[,2])
# R equivalent: pacf from stats
pacf(AirPassengers, lag.max = 35)

# use with composite (see vignette("multiple", "seasonal"))
```

```
m_composite <- seas(
  cbind(mdeaths, fdeaths),
  composite = list(),
  series.comptype = "add"
)
series(m_composite, "composite.indseasadj")
```

---

spc   .spc *File Content*

Description

Access the content of the .spc file that governs the behavior of X-13ARIMA-SEATS.

Usage

```
spc(x)
```

Arguments

x   object of class "seas"

Value

returns an object of class "spclist", essentially a list that contains the information that is sent to X-13ARIMA-SEATS. The corresponding print method displays the content of the list as written to the .spc file.

References

Vignette with a more detailed description: <http://www.seasonal.website/seasonal.html>

Comprehensive list of R examples from the X-13ARIMA-SEATS manual: http://www.seasonal.website/examples.html

Official X-13ARIMA-SEATS manual: https://www2.census.gov/software/x-13arima-seats/x13as/windows/documentation/docx13as.pdf

See Also

seas() for the main function.

series(), for universal X-13 output extraction.

plot.seas(), for diagnostical plots.

[out()](out()), for accessing the full output of X-13ARIMA-SEATS.

Examples

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

SPECS   *List of Available X-13ARIMA-SEATS Outputs*

Description

The data is used by several functions as a look-up table. Users should consider the table in [series()](#) or in the official manual.

Format

An object of class `"data.frame"`

Source

United States Census Bureau

References

Official X-13ARIMA-SEATS manual: https://www2.census.gov/software/x-13arima-seats/x13as/windows/documentation/docx13as.pdf

---

static   *Static Call of a seas Object*

Description

In a 'static' call, the default automatic procedures in the model call are substituted by the choices they made.

Usage

```
static(
  x,
  coef = FALSE,
  x11.filter = FALSE,
  test = TRUE,
  fail = FALSE,
  evaluate = FALSE
)
```

Arguments

x	an object of class seas.
coef	logical. If TRUE, the coefficients are treated as fixed, instead of being estimated.
x11.filter	logical. X-11 only. if TRUE, the X-11 moving averages will be fixed as well. This leads to different filters at different stages, and the resulting series can be are slightly different. If test = TRUE, this may cause a warning message.
test	logical. By default the static call is executed and compared to the input call. If the final series is not identical, a message is returned. If FALSE, no test is performed (faster).
fail	logical. If TRUE, differences will cause an error. Ignored if test = FALSE.
evaluate	logical. If TRUE, the call is evaluated.

Details

If evaluate = TRUE, the call is evaluated. The call can be copy/pasted to a script and used for further manipulations or future evaluation of the same model.

By default, the static call is tested. It is executed and compared to the input call. If the final series is not identical, a message is returned.

If coef = TRUE, the coefficients are fixed as well. If x11.filter = TRUE, the X-11 moving averages are fixed as well.

Value

Object of class "call". Or an object of class "seas" if evaluate = TRUE.

References

Vignette with a more detailed description: http://www.seasonal.website/seasonal.html

Comprehensive list of R examples from the X-13ARIMA-SEATS manual: http://www.seasonal.website/examples.html

Official X-13ARIMA-SEATS manual: https://www2.census.gov/software/x-13arima-seats/x13as/windows/documentation/docx13as.pdf

See Also

stats::getCall() to extract the actual call.

seas() for the main function of seasonal.

Examples

```
m <- seas(AirPassengers)
getCall(m)                 # default call
static(m)                  # static call
static(m, test = FALSE)    # much faster
static(m, evaluate = TRUE) # returns an object of class "seas"
```

```
m <- seas(AirPassengers, x11 = "")

static(m, x11.filter = TRUE) # also fixes the X-11 filter (with a warning)
static(m, coef = TRUE)       # also fixes the coefficients
```

---

summary.seas   *Summary of a X13-ARIMA-SEATS seasonal adjustment*

Description

Like the corresponding method for ″lm″ objects, the method for ″seas″ objects returns the esti-
mated coefficients, its standard errors, z-statistics and corresponding (two-sided) p-values. Coeffi-
cients are returned both for the exogenous regressors and the coefficients of the ARIMA model.

Usage

```
## S3 method for class 'seas'
summary(object, stats = getOption("seas.stats"), ...)

## S3 method for class 'summary.seas'
print(
  x,
  digits = max(3, getOption("digits") - 3),
  signif.stars = getOption("show.signif.stars"),
  ...
)
```

Arguments

object	an object of class ″seas″, usually, a result of a call to seas().
stats	(experimental) character vector, additional stat to be shown in the summary out-put. function. For a list of all possible values, see the udg() function. If a value is not present, it will be ignored. Values can be specified via options. See examples.
...	further arguments passed to or from other methods.
x	an object of class ″summary.seas″, usually, a result of a call to summary.seas.
digits	the number of significant digits to use when printing.
signif.stars	logical. If TRUE, 'significance stars' are printed for each coefficient.

Details

The lower part of the output shows additional information on the estimation:

Adjustment use of SEATS or X11

ARIMA structure of the seasonal ARIMA model

Obs. number of observations

Transform prior transformation

AICc, BIC value of the information criterion (lower is better)

QS test for seasonality in the final series; null hypothesis: no seasonality in final; signif. codes are shown if the null hypothesis is rejected. QS statistics for more series (e.g., the original series) can be extracted with `qs()`.

Box-Ljung test for residual autocorrelation; null hypothesis: no autocorrelation in residuals; signif. codes are shown if the null hypothesis is rejected. The test statistic is the result of `Box.test(resid(m), lag = 24, type = "Ljung")`

Shapiro test for normality of the residuals; null hypothesis: normal distribution of the residuals; signif. codes are shown if the null hypothesis is rejected. The test statistic is the result of `shapiro.test(resid(m))`

Value

`summary.seas` returns a list containing the summary statistics included in `object`, and computes the following additional statistics:

`coefficients`	a named matrix containing coefficients, standard deviations, t-values and p-values
`transform`	character string with the type of initial transformation

The `print` method prints the summary output in a similar way as the method for `"lm"`.

Examples

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

### user defined stats from the udg function
# (experimental, see ?udg)

# also show some M quality statistics for X11 in summary
options(seas.stats = c("f3.m01", "f3.m02", "f3.m03", "f3.m04"))
summary(seas(AirPassengers, x11 = ""))

# this does not affect the SEATS output
summary(seas(AirPassengers))

# reset to default
options(seas.stats = NULL)
```

transformfunction   *Applied Transformation*

Description

Returns the transform function that has been applied.

Usage

```
transformfunction(x)
```

Arguments

x   object of class `"seas"`

References

Vignette with a more detailed description: <http://www.seasonal.website/seasonal.html>

Comprehensive list of R examples from the X-13ARIMA-SEATS manual: http://www.seasonal.website/examples.html

Official X-13ARIMA-SEATS manual: https://www2.census.gov/software/x-13arima-seats/x13as/windows/documentation/docx13as.pdf

See Also

[seas()](#) for the main function.

[series()](#), for universal X-13 output extraction.

[plot.seas()](#), for diagnostical plots.

[out()](#), for accessing the full output of X-13ARIMA-SEATS.

Examples

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

---

udg *Diagnostical Statistics*

Description

The udg function provides access to a large number of diagnostical statistics. The qs function and the AIC, BIC and logLik methods are wrappers that use udg to access some specific diagnostical statistics.

Usage

```
udg(x, stats = NULL, simplify = TRUE, fail = TRUE)

qs(x)

## S3 method for class 'seas'
AIC(object, ...)

## S3 method for class 'seas'
BIC(object, ...)

## S3 method for class 'seas'
nobs(object, ...)

## S3 method for class 'seas'
logLik(object, ...)
```

Arguments

x, object	an object of class "seas".
stats	character vector; if specified, only a subset of the available stats are returned. This speeds up the call, as only a subset needs to be type converted. Should be used for programming.
simplify	logical; should the result be simplified to a vector or matrix, if possible?
fail	logical; if TRUE, an error is dropped if an element of stats is missing in names(udg(x)).
...	further arguments (not used)

Value

qs returns the QS statistics for seasonality of input and output series and the corresponding p-values.

AIC, BIC, nobs and logLik return the corresponding statistics.

References

Vignette with a more detailed description: <http://www.seasonal.website/seasonal.html>

Comprehensive list of R examples from the X-13ARIMA-SEATS manual: http://www.seasonal.website/examples.html

Official X-13ARIMA-SEATS manual: https://www2.census.gov/software/x-13arima-seats/x13as/windows/documentation/docx13as.pdf

See Also

seas() for the main function.

series(), for universal X-13 output extraction.

plot.seas(), for diagnostical plots.

[out()](out()), for accessing the full output of X-13ARIMA-SEATS.

Examples

```
m <- seas(AirPassengers, x11 = "")

qs(m)
AIC(m)
BIC(m)
nobs(m)
logLik(m)

# a list with all entries from udg
udg(m)

# extracting a few selected stats from udg
udg(m, c("f3.m02", "f3.m05", "qsori"))  # returns a list
udg(m, c("f3.m02", "f3.m05"))           # returns a vector

# faster than:
udg(m)[c("f3.m01", "f3.m02", "qsori")]
```

---

unemp	*United States Unemployment Level*

Description

Thousands of Persons

Format

Each time series is an object of class `"ts"`.

Source

U.S. Bureau of Labor Statistics, retrieved from FRED, Federal Reserve Bank of St. Louis; `https://fred.stlouisfed.org/series/LNU03000000`, December 14, 2016.

Examples

```
data(seasonal)
unemp
```

---

update.seas   *Update and Re-evaluate a Seasonal Adjustment Model*

Description

Method to update and re-evaluate an object of class `"seas"`.

Usage

```
## S3 method for class 'seas'
update(object, ..., evaluate = TRUE)
```

Arguments

object	an object of class `"seas"`, usually, a result of a call to `seas()`.
...	spec-argument options sent to X-13 (with the same syntax as in `seas()`)
evaluate	logical. If `TRUE`, the call is evaluated.

Details

Contrary to the default method of `update()`, the `"seas"` method uses the evaluated call, rather than the actual call for re- evaluation. This means you can savely use it in other functions, which is useful with `lapply()` and friends (see examples.)

Value

Object of class `"seas"`. Or an object of class `"call"` if evaluate = FALSE.

See Also

`seas()` for the main function.

`static()`, to return the (optionally evaluated) static call of a `"seas"` object.

Examples

```
# updating the call
m <- seas(AirPassengers)
update(m, x11 = "")
update(m, x = sqrt(AirPassengers), x11 = "")

# 'update' can be also used with lapply (or mapply)

# a list of time series
dta <- list(fdeaths = fdeaths, mdeaths = mdeaths)

# use 'seas' via lapply
ll <- lapply(dta, seas, x11 = "")

# use 'update' via lapply
lapply(ll, update, arima.model = c(0, 1, 1, 0, 1, 1))
```

---

view   *Interactively Modify a Seasonal Adjustment Model*

Description

Interactively modify a "seas" object. The goal of view is to summarize all relevant options, plots
and statistics of a seasonal adjustment model. The view function in the **seasonal** package imports
the identical seasonalview::view() function from the **seasonalview** package, so there is no need
to explicitly load the **seasonalview** package.

Usage

```
view(x = NULL, story = NULL, quiet = TRUE, ...)
```

Arguments

x	an object of class "seas".
story	character, local file path or URL to an ".Rmd" file.
quiet	logical, if TRUE (default), error messages from calls in view are not shown in the console.
...	arguments passed to runApp. E.g., for selecting if the GUI should open in the browser or in the RStudio viewer pane.

Details

Frequently used options can be modified using the drop down selectors in the upper left box. Each
change will result in a re-estimation of the seasonal adjustment model. The R-call, the X-13 call,
the graphical output and the summary are updated accordingly.

Alternatively, the R call can be modified manually in the lower left box. Click 'Run Call' to re-estimate the model and to adjust the option selectors, the graphical output, and the summary. With the 'To console' button, the GUI is closed and the call is imported to R. The 'Static' button substitutes automatic procedures by the automatically chosen spec-argument options, in the same way as the `static()` function.

If you are familiar with the X-13 spec syntax, you can modify the X-13 call, with the same consequences as when modifying the R call.

The lower right panel shows the summary, as described in the help page of `summary.seas()`. The 'X-13 output' button opens the complete output of X-13 in a separate tab or window.

If you have the x13story package installed (not yet on CRAN, see references), you can call the function with the `story` argument. This will render an R Markdown document and produce a *story* on seasonal adjustment that can be manipulated interactively.

Value

`view` returns an object of class `"seas"`, the modified model; or `NULL`, if the `story` argument is supplied.

References

Seasonal vignette with a more detailed description: http://www.seasonal.website/seasonal.html

Development version of the x13story package: https://github.com/christophsax/x13story

Examples

```
## Not run:

m <- seas(AirPassengers)
view(m)

# store the model after closing the GUI, for further processing in R
m.upd <- view(m)

## End(Not run)
```

Index