Package 'robets'

October 14, 2022

Type Package

Title Forecasting Time Series with Robust Exponential Smoothing

Version 1.4

Date 2018-03-06

Description
We provide an outlier robust alternative of the function ets() in the 'forecast' package of Hyndman and Khandakar (2008) <DOI:10.18637/jss.v027.i03>. For each method of a class of exponential smoothing variants we made a robust alternative. The class includes methods with a damped trend and/or seasonal components. The robust method is developed by robustifying every aspect of the original exponential smoothing variant. We provide robust forecasting equations, robust initial values, robust smoothing parameter estimation and a robust information criterion. The method is described in more detail in Crevits and Croux (2016) <DOI:10.13140/RG.2.2.11791.18080>.

License GPL-3

Depends R (>= 3.1.1)

Imports Rcpp (>= 0.12.2), forecast

LinkingTo Rcpp

LazyData true

ByteCompile true

BugReports https://github.com/RubenCrevits/robets/issues

URL http://github.com/RubenCrevits/robets

RoxygenNote 6.0.1

NeedsCompilation yes

Author Ruben Crevits [aut, cre],
Christoph Bergmeir [aut],
Rob Hyndman [aut],
Ross Ihaka [ctb],
R Core Team [ctb]

Maintainer Ruben Crevits <ruben.crevits@gmail.com>

Repository CRAN

Date/Publication 2018-03-06 16:46:30 UTC

R topics documented:

`coef.robets` *Coef robets model*

Description

Coef robets model

Usage

```
## S3 method for class 'robets'
coef(object, ...)
```

Arguments

`object`	An object of class `robets`.
`...`	Other undocumented arguments.

Examples

```
model <- robets(nottem)
coef(model)
```

`forecast.robets` *Forecasting using ROBETS models*

Description

Returns forecasts and other information for univariate ROBETS models.

Usage

```
## S3 method for class 'robets'
forecast(object, h = ifelse(object$m > 1, 2 * object$m, 10),
  level = c(80, 95), PI = TRUE, lambda = object$lambda, ...)
```

Arguments

object	An object of class "robets". Usually the result of a call to robets.
h	Number of periods for forecasting
level	Confidence level for prediction intervals.
PI	If TRUE, prediction intervals are calculated.
lambda	Box-Cox transformation parameter. Ignored if NULL. Otherwise, forecasts back-transformed via an inverse Box-Cox transformation.
...	Other arguments.

Details

The code of this function is based on the function forecast.ets of the package forecast of Hyndman and Khandakar (2008).

Value

An object of class "forecast". The function summary is used to obtain and print a summary of the results, while the function plot produces a plot of the forecasts. The generic accessor functions fitted.values and residuals extract useful features of the value returned by forecast.robets. An object of class "forecast" is a list containing at least the following elements:

- model: A list containing information about the fitted model
- method: The name of the forecasting method as a character string
- mean: Point forecasts as a time series
- x: The original time series (either object itself or the time series used to create the model stored as object).
- residuals: Residuals from the fitted model. For models with additive errors, the residuals are x - fitted values. For models with multiplicative errors, the residuals are equal to x /(fitted values) - 1.
- fitted: Fitted values (one-step ahead forecasts)

Author(s)

Ruben Crevits, <ruben.crevits@kuleuven.be>, https://rcrevits.wordpress.com/research

References

Crevits, R., and Croux, C (2016) "Forecasting with Robust Exponential Smoothing with Damped Trend and Seasonal Components".*Working paper*. https://doi.org/10.13140/RG.2.2.11791.18080

Hyndman, R. J., and Khandakar, Y (2008) "Automatic time series forecasting: The forecasting package for R".*Journal of Statistical Software* **27**(3). https://doi.org/10.18637/jss.v027.i03

See Also

robets

Examples

```
library(forecast)
model <- robets(nottem)
plot(forecast(model))
```

`plot.robets` *Plot robets model*

Description

Plot robets model

Usage

```
## S3 method for class 'robets'
plot(x, ...)
```

Arguments

`x`	An object of class `robets`.
`...`	Other plotting parameters.

See Also

`plotOutliers`, `plot.ets`

Examples

```
model <- robets(nottem)
plot(model)
```

`plotOutliers` *Plot outliers detected by robets model*

Description

Plot outliers detected by robets model

Usage

```
plotOutliers(object, xlab = "", ylab = "", type = "l", ...)
```

Arguments

object	An object of class robets.
xlab	Label of the x-axis.
ylab	Label of the y-asix.
type	Character indicating the type of plot, just as in plot.
...	Other plotting parameters.

See Also

plot.robets

Examples

```
model <- robets(nottem)
plotOutliers(model)
```

---

print.robets *Print robets model*

Description

Print robets model

Usage

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

Arguments

x	An object of class robets.
...	Other undocumented arguments.

Examples

```
model <- robets(nottem)
print(model)
```

robets *Robust exponential smoothing model*

Description

Returns robets model applied to y.

Usage

```
robets(y, model = "ZZZ", damped = NULL, alpha = NULL, beta = NULL,
  gamma = NULL, phi = NULL, additive.only = FALSE, lambda = NULL,
  lower = c(rep(1e-04, 3), 0.8), upper = c(rep(0.9999, 3), 0.98),
  opt.crit = c("roblik", "tau2", "lik", "mse", "amse", "sigma", "mae"),
  bounds = c("both", "usual", "admissible"), ic = c("robaicc", "robaic",
  "robbic", "aicc", "bic", "aic"), use.initial.values = TRUE,
  opt.initial.values = FALSE, rob.start.initial.values = TRUE,
  opt.sigma0 = FALSE, k = 3, nmse = 1, ...)
```

Arguments

y	a numeric vector or time series
model	A three-letter string indicating the method using the framework terminology of Hyndman et al. (2008). The first letter denotes the error type ("A", "M" or "Z"); the second letter denotes the trend type ("N","A" or "Z"); and the third letter denotes the season type ("N","A","M" or "Z"). In all cases, "N"=none, "A"=additive, "M"=multiplicative and "Z"=automatically selected. So, for example, "ANN" is simple exponential smoothing with additive errors, "MAM" is multiplicative Holt-Winters' method with multiplicative errors, and so on. It is also possible for the model to be of class "robets", and equal to the output from a previous call to robets. In this case, the same model is fitted to y without re-estimating any smoothing parameters. See also the use.initial.values argument.
damped	If TRUE, use a damped trend. If NULL, both damped and non-damped trends will be tried and the best model (according to the information criterion ic) will be returned.
alpha	Value of alpha. If NULL, it is estimated.
beta	Value of beta. If NULL, it is estimated.
gamma	Value of gamma. If NULL, it is estimated.
phi	Value of phi. If NULL, it is estimated.
additive.only	If TRUE, will only consider additive models. Default is FALSE.
lambda	Box-Cox transformation parameter. Ignored if NULL. Otherwise, data transformed before model is estimated. When lambda=TRUE, additive.only is set to FALSE.
lower	Lower bounds for the parameters (alpha, beta, gamma, phi)

upper — Upper bounds for the parameters (alpha, beta, gamma, phi)

opt.crit — Optimization criterion. One of "roblik" (Robust Log-likelihood, default), "tau2" (Tau squared error of the residuals), "mse" (Mean Square Error), "amse" (Average MSE over first nmse forecast horizons), "sigma" (Standard deviation of residuals), "mae" (Mean of absolute residuals), or "lik" (Log-likelihood).

bounds — Type of parameter space to impose: "usual" indicates all parameters must lie between specified lower and upper bounds; "admissible" indicates parameters must lie in the admissible space; "both" (default) takes the intersection of these regions.

ic — Information criterion to be used in model selection.

use.initial.values — If TRUE (default) and model is of class "robets", then the initial values in the model are also not re-estimated.

opt.initial.values — If FALSE (default) a robust heuristic is used for chosing the initial values. If TRUE the initial values are part of the problem to optimize opt.crit. Neglected if use.initial.values is TRUE and model is of class "robets".

rob.start.initial.values — If TRUE (default) the initial values are computed via the robust heuristic described in Crevits and Croux (2016). If FALSE the initial values are computed via the same heuristic as in Hyndman et al. (2008). The initial values computed with these methods are further optimized if opt.initial.values is TRUE.

opt.sigma0 — If FALSE (default) sigma0 is equal to the value computed together with the other initial values via a heuristic. If TRUE sigma0 is included as a variable in the optimization problem. It is not recommended to set opt.sigma0 = TRUE.

k — Value of k in forecasting equations. k=3 is default. If NULL, k is included as a variable in the optimization problem. It is not recommended to set k = NULL.

nmse — Number of steps for AMSE (1<=nmse<=30), nmse=1 is default.

... — Other undocumented arguments.

Details

The code is an extended version of the code of the function ets of the package forecast of Hyndman and Khandakar (2008). The methodology is an extended version of Gelper et al. (2008). In Crevits and Croux (2016) the methodology of robets is described in full.

Value

An object of class "robets".

Author(s)

Ruben Crevits, <ruben.crevits@kuleuven.be>, https://rcrevits.wordpress.com/research

References

Crevits, R., and Croux, C (2016) "Forecasting with Robust Exponential Smoothing with Damped Trend and Seasonal Components".*Working paper*. https://doi.org/10.13140/RG.2.2.11791.18080

Gelper S., Fried R. and Croux C. (2010) "Robust Forecasting with Exponential and Holt-Winters Smoothing".*Journal of Forecasting*, **29**, 285-300. https://doi.org/10.1002/for.1125

Hyndman, R. J., and Khandakar, Y (2008) "Automatic time series forecasting: The forecasting package for R".*Journal of Statistical Software* **27**(3). https://doi.org/10.18637/jss.v027.i03

See Also

forecast.robets, plot.robets, plotOutliers, tau2, ets

Examples

```
library(forecast)
model <- robets(nottem)
plot(forecast(model))
```

`summary.robets` *Summary robets model*

Description

Summary robets model

Usage

```
## S3 method for class 'robets'
summary(object, ...)
```

Arguments

`object`	An object of class `robets`.
`...`	Other undocumented arguments.

Value

A number of training set error measures: ME (mean error), RMSE (root mean squared error), MAE (mean absolute error), MPE (mean percentage error), MAPE (mean absolute percentage error), MedianE (median error), RTSE (root tau squared error), RTSPE (root tau squared percentage error).

Examples

```
model <- robets(nottem)
summary(model)
```

tau2 *Compute the tau2 estimator of scale*

Description

The tau2-estimator is a robust measure of the scale. The exact formula of the estimator is in Crevits and Croux (2016), equation 3.10.

Usage

```
tau2(x)
```

Arguments

x	A vector of residuals.

Value

The tau2 estimate of scale.

References

Crevits, R., and Croux, C (2016) "Forecasting with Robust Exponential Smoothing with Damped Trend and Seasonal Components".*Working paper*. https://doi.org/10.13140/RG.2.2.11791.18080

Examples

```
set.seed(100)
e <- 10*rnorm(100)
mse <- mean(e^2)
tse <- tau2(e)
```

Index