Package ‘MCS’

October 19, 2026

Type Package

Title Model Confidence Set Procedure

Version 0.2.1

Date 2026-10-19

Maintainer Leopoldo Catania <leopoldo.catania@econ.au.dk>

Depends R (>= 3.0.1)

Description Implements the Model Confidence Set procedure of Hansen, Lunde and Nason (2011) <doi:10.3982/ECTA5771> using bootstrap tests of equal predictive ability. Includes loss functions for level, volatility and quantile forecasts.

License GPL-2

Encoding UTF-8

Imports methods, stats

NeedsCompilation no

Author Leopoldo Catania [aut, cre] (ORCID: <https://orcid.org/0000-0002-0981-1921>)

Config/roxygen2/version 8.0.0

Repository CRAN

Date/Publication 2026-10-19 12:30:02 UTC

Contents

LossLevel 2
LossVaR 2
LossVol 3
MCSprocedure 4
show,SSM-method 6
SSM-class 6

Index **7**

LossLevel — *Loss Function for Level Forecasts*

Description

Calculate squared or absolute forecast errors for one or more models.

Usage

```
LossLevel(realized, evaluated, which = "SE")
```

Arguments

`realized`	A finite numeric vector or one-column matrix of observations.
`evaluated`	A finite numeric vector, matrix or numeric data frame of forecasts with the same number of observations as realized.
`which`	"SE" for squared error or "AE" for absolute error.

Details

Matrix columns are evaluated against the same realized series. Empty or non-finite inputs and non-finite calculated losses raise errors.

Value

A numeric matrix with the dimensions and dimnames of evaluated.

Author(s)

Leopoldo Catania

LossVaR — *Loss Function for VaR Forecasts*

Description

Calculate quantile losses for one or more forecast series.

Usage

```
LossVaR(realized, evaluated, which = "asymmetricLoss",
        type = "normal", delta = 25, tau)
```

Arguments

`realized`	A finite numeric vector or one-column matrix of returns.
`evaluated`	A finite numeric vector, matrix or numeric data frame of quantile forecasts, with the same number of rows as realized.
`which`	Must be "asymmetricLoss".
`type`	"normal" for the quantile check loss, or "differentiable" for the logistic smooth approximation.
`delta`	A finite positive scalar controlling the smooth approximation.
`tau`	Quantile probability in (0,1). For a 99 percent lower-tail VaR of returns use tau=0.01, not 0.99.

Details

With u=realized-evaluated, the normal loss is (tau-1{u<0})*u. The differentiable version replaces the indicator with plogis(-delta*u); it is an approximation for finite delta. Inputs must be finite and nonempty. Non-finite calculated losses raise an error rather than being passed silently to MCSprocedure.

Value

A numeric matrix with the dimensions and dimnames of evaluated.

Author(s)

Leopoldo Catania

References

Koenker R, Bassett G (1978). Regression quantiles. Econometrica, 46(1), 33-50.

Gonzalez-Rivera G, Lee TH, Mishra S (2004). Forecasting volatility: A reality check based on option pricing, utility function, value-at-risk, and predictive likelihood. International Journal of Forecasting, 20(4), 629-645. doi:10.1016/j.ijforecast.2003.10.003.

`LossVol` *Loss Function for Volatility Forecasts*

Description

Calculate forecast losses for standard deviations, not variances.

Usage

```
LossVol(realized, evaluated, which = "SE1")
```

Arguments

`realized`	A finite nonnegative vector or one-column matrix of realized standard deviations. Strict positivity is required for R2LOG.
`evaluated`	A finite nonnegative vector, matrix or numeric data frame of forecast standard deviations. Strict positivity is required for QLIKE and R2LOG. The number of rows must equal the number of realized observations.
`which`	One of "SE1", "SE2", "QLIKE", "R2LOG", "AE1", "AE2".

Details

For realized standard deviation s and forecast standard deviation f, the losses are (s-f)^2 (SE1), (s^2-f^2)^2 (SE2), log(f^2)+s^2/f^2 (QLIKE), [log(s^2/f^2)]^2 (R2LOG), abs(s-f) (AE1), and abs(s^2-f^2) (AE2). Zero realized volatility is allowed for QLIKE; R2LOG is undefined there and raises an error. Non-finite losses from arithmetic overflow raise an error. Matrix forecast columns are evaluated against the same realized series.

Value

A numeric matrix with the dimensions and dimnames of evaluated.

Author(s)

Leopoldo Catania

References

Hansen PR, Lunde A (2005). A forecast comparison of volatility models: does anything beat a GARCH(1,1)? Journal of Applied Econometrics, 20(7), 873-889. doi:10.1002/jae.800.

Bernardi M, Catania L (2018). The Model Confidence Set package for R. International Journal of Computational Economics and Econometrics, 8(2), 144-158. doi:10.1504/IJCEE.2018.091037.

`MCSprocedure` *Model Confidence Set Procedure*

Description

Construct a model confidence set from competing loss series.

Usage

```
MCSprocedure(Loss, alpha = 0.15, B = 1000, statistic = "Tmax",
             k = NULL, min.k = 3, verbose = TRUE, seed = NULL)
```

Arguments

`Loss`	A finite numeric matrix, numeric data frame, or object coercible to a numeric matrix, with observations in rows and models in columns. At least two observations and one model are required. Column names, if supplied, must be unique and nonempty.
`alpha`	Significance level in (0,1). The MCS confidence level is 1-alpha.
`B`	Integer number of bootstrap replications, at least 2. Values below 100 produce a warning.
`statistic`	Either "Tmax" or "TR".
`k`	Integer block length with 0 <= k < nrow(Loss). Positive lengths use a non-circular moving-block bootstrap; k=0 uses iid resampling (as does k=1). If NULL, use the maximum number of coefficients significant at the two-sided 5 percent level in AIC-selected Yule-Walker AR models fitted to all pairwise loss differences, subject to min.k. Constant differences contribute zero.
`min.k`	Nonnegative integer lower bound for automatic block length, smaller than nrow(Loss) when automatic selection is used.
`verbose`	A single logical value controlling progress messages.
`seed`	NULL or an integer between 0 and .Machine$integer.max. If NULL, a seed is drawn and recorded. The function calls set.seed and advances R's global random-number state.

Details

The complete elimination path is computed. MCS p-values are running maxima of sequential p-values; models with MCS p-value >= alpha are included. Equality is retained, consistent with rejection when p < alpha. Bootstrap p-values use mean(T_boot >= T_observed), including ties.

Zero loss contrasts contribute zero statistics. Numerically constant nonzero contrasts contribute signed infinite observed statistics and zero centered bootstrap statistics. All-identical loss matrices return p-values of 1; constant strictly worse losses can be eliminated. Constancy is checked to a relative tolerance of 64*.Machine$double.eps when bootstrap errors are tiny. Zero bootstrap variance for a varying contrast raises an error. These are explicit degeneracy conventions, not additional asymptotic coverage claims. Inference for nondegenerate losses requires the stationarity, dependence and moment conditions of Hansen, Lunde and Nason (2011).

Bootstrap model means are computed once and reused. No B-by-m-by-m arrays are allocated. A single-model input returns that model with p-values 1; automatic block selection and resampling are skipped, and automatic k is reported as 0 in that case.

Value

An S4 object of class SSM. Slot show contains average losses, sequential test p-values and cumulative MCS p-values. Slot Info contains included/excluded model names, settings, elimination.order and timing.

Author(s)

Leopoldo Catania

References

Hansen PR, Lunde A, Nason JM (2011). The model confidence set. Econometrica, 79(2), 453-497. doi:10.3982/ECTA5771.

Bernardi M, Catania L (2018). The Model Confidence Set package for R. International Journal of Computational Economics and Econometrics, 8(2), 144-158. doi:10.1504/IJCEE.2018.091037.

Examples

```
set.seed(123)
y <- rnorm(500)
forecasts <- matrix(rep(seq(-0.5, 0.5, length.out=11), 500),
nrow=500, byrow=TRUE)
Loss <- LossLevel(y, forecasts, which="SE")
MCSprocedure(Loss, seed=123, verbose=FALSE)
```

`show,SSM-method` *Display a Model Confidence Set*

Description

Display the result table and settings used for inference.

Usage

```
## S4 method for signature 'SSM'
show(object)
```

Arguments

`object`	An object of class SSM.

`SSM-class` *Model Confidence Set Result*

Description

An S4 container returned by `MCSprocedure`.

Slots

`Info` A list containing model.names, included, excluded, n_elim, statistic, alpha, B, k, seed, elimination.order and elapsed.time.

`show` A matrix containing average losses, sequential test p-values and cumulative MCS p-values. Membership is based on MCS p-value >= alpha.

Index

LossLevel, 2
LossVaR, 2
LossVol, 3

MCSprocedure, 4

show,SSM-method, 6
SSM-class, 6