Package ‘INFOSET’

May 7, 2026

Type Package

Title Computing a New Informative Distribution Set of Asset Returns

Version 4.1.1

Author Gloria Polinesi [aut, cre],
Francesca Mariani [aut],
Maria Cristina Recchioni [aut]

Maintainer Gloria Polinesi <g.polinesi@staff.univpm.it>

Description Estimation of the most-left informative set of gross returns
(i.e., the informative set).
The procedure to compute the informative set adjusts the method
proposed by
Mariani et al. (2022a) <doi:10.1007/s11205-020-02440-6>
and
Mariani et al. (2022b) <doi:10.1007/s10287-022-00422-2>
to gross returns of financial assets.
This is accomplished through an adaptive algorithm
that identifies sub-groups of gross returns in
each iteration by approximating their distribution with a
sequence of two-component log-normal mixtures.
These sub-groups emerge when a significant change
in the distribution occurs below the median of the
financial returns, with their boundary termed as
the “change point" of the mixture.
The process concludes when no further change points are detected.
The outcome encompasses parameters of the leftmost mixture
distributions and change points of the
analyzed financial time series.
The functionalities of the INFOSET package include: (i) modelling asset distribution
detecting the parameters which describe left tail behaviour (infoset function), (ii) clustering, (iii) labeling of the financial
series for predictive and classification purposes through a Left Risk measure based on the first change point (LR_cp function)
(iv) portfolio construction (ptf_construction function).
The package also provide a specific function to construct rolling windows of different length size and overlapping time.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Imports Matrix, colorspace, dendextend, quadprog, mixtools, stats, graphics

Depends R (>= 2.10)

RoxygenNote 7.3.3

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2026-02-11 15:50:02 UTC

Contents

`asset.label` *Data for clustering and labeling ETFs*

Description

Contains asset class of ETFs

Usage

```
asset.label
```

Format

A data frame with 44 observations (rows) on 3 variables (columns)

id name of ETF

label from 1 to 5 according to the specific asset class

class specific asset class (5 categories)

Source

Created in-house to serve as an example

Examples

```
data(asset.label)
```

`create_overlapping_windows` *Function to create overlapping windows.*

Description

Function to create overlapping windows.

Usage

```
create_overlapping_windows(data, FT = 1290, ov = 125)
```

Arguments

`data`	vector or data frame
`FT`	Window size. By default set to 1290 training days (five years).
`ov`	Number of different days for two consecutive time windows.. By default set to 125 training days (six months).

Value

a list containing the rolling windows

g_ret — *Function to compute gross returns.*

Description

Calculate gross returns from prices.

Usage

```
g_ret(x)
```

Arguments

x	data object containing ordered price observations

Value

An object of the same class as x with the gross returns

infoset — *Procedure to find the most-left distribution set.*

Description

Estimation of the vector of unknown parameters for the density functions associated with the two mixture components.

Usage

```
infoset(y, plot_cp)
```

Arguments

y	object of class "g_ret"
plot_cp	option

Value

An object of class "infoset" is a list containing the following components for the firse two iterations (k=2):

change.points a vector of change points.

prior.probability the a priori probabilities.

first.type.errors the cumulative distribution functions associated with the leftmost component of the mixture.

second.type.errors the cumulative distribution functions associated with the rightmost component of the mixture.

mean the parameters (drift) of the left-hand component of the log-normal mixture.

sd the parameters (volatility) of the left-hand component of the log-normal mixture.

References

Mariani, F., Polinesi, G., Recchioni, M. C. (2022). A tail-revisited Markowitz mean-variance approach and a portfolio network centrality. Computational Management Science, 19(3), 425-455.

Mariani, F., Ciommi, M., Chelli, F. M., Recchioni, M. C. (2020). An iterative approach to stratification: Poverty at regional level in Italy. Social Indicators Research, 1-31.

Examples

```
gross.ret<-as.data.frame(lapply(sample.data, g_ret))
infoset(gross.ret$ETF_1, plot_cp = "T")

############################################################
## EXAMPLE 1: Clustering ETFs
############################################################


gross.ret<-as.data.frame(lapply(sample.data, g_ret))
result<-NULL
for(i in 1:ncol(gross.ret)){
 result[[i]]<-infoset(gross.ret[,i], plot_cp = "F")
}
output<-matrix(unlist(result),12,ncol=ncol(gross.ret)) # output contains the information set
output<-t(output)
rownames(output)<-colnames(gross.ret)
colnames(output)<-c("ch_1","ch_2","priori_1","priori_2","first_1",
                   "first_2","second_1","second_2","mean_1","mean_2","dev_1", "dev_2")
output<- as.data.frame(output)
group_label <- as.factor(asset.label$label)
d <- dist(output, method = 'euclidean')
hc_SIMS <- hclust(d, method = 'complete')
library(dendextend)
library(colorspace)
dend_SIMS <- as.dendrogram(hc_SIMS)
dend_SIMS <- color_branches(dend_SIMS, k = 4, col = c(1:4))
labels_colors(dend_SIMS) <-
     rainbow_hcl(5)[sort_levels_values(as.numeric(group_label)[order.dendrogram(dend_SIMS)])]
labels(dend_SIMS) <- paste(as.character(group_label)[order.dendrogram(dend_SIMS)],
       '(', labels(dend_SIMS), ')', sep = '')
dend_SIMS <- hang.dendrogram(dend_SIMS, hang_height = 0.001)
dend_SIMS <- assign_values_to_leaves_nodePar(dend_SIMS, 0.5, 'lab.cex')
dev.new()
old_par <- par(no.readonly = TRUE)
on.exit(par(old_par))
par(mar = c(1.8, 1.8, 1.8, 1))
plot(dend_SIMS, main = 'Complete linkage (the labels give the true ETF class)',
```

```
     horiz = TRUE, nodePar = list(cex = 0.007))
legend('topleft', legend = c('emerging equity Asia', 'emerging equity America',
                            'corporate bond', 'commodities', 'aggregate bond'),
         fill = c('#BDAB66', '#65BC8C', '#C29DDE', '#E495A5', '#55B8D0'), border = 'white')
```

LR_cp *Function to compute Left risk measure.*

Description

Function to compute Left risk measure.

Usage

```
LR_cp(data, FT, ov)
```

Arguments

data	A (T x N) matrix or data.frame containing the N time series over period T
FT	Window size.
ov	umber of different days for two consecutive time windows.

Value

A (N x T) data.frame containing the LR_cp measure for the N time series over time windows

Examples

```
LR <- LR_cp(sample.data, FT= 1290, ov = 125)
df <- as.data.frame(matrix(unlist(LR), nrow = length(LR), ncol = ncol(sample.data)))
colnames(df) <- c(paste("tw", rep(1:16)))
plot(df[,1], pch=19, col=asset.label$label, ylab="LR_cp", xlab="ETFs")
```

`plot_LR_cp` *Plot methods for a LR_cp object*

Description

Plot methods for a LR_cp object

Usage

```
plot_LR_cp(LR_cp_measure, asset_label)
```

Arguments

`LR_cp_measure`	object of class LR_cp
`asset_label`	vector containing asset label

Value

plot of LR_cp measures by asset classes

`plot_ptf` *Plot methods for a ptf_construction object*

Description

Plot methods for a ptf_construction object

Usage

```
plot_ptf(ptf.oos.values)
```

Arguments

`ptf.oos.values`	object of class ptf_construction

Value

plot oos portfolio values

ptf_construction *Function to compute portfolio values*

Description

Function to compute portfolio values

Usage

```
ptf_construction(
  data,
  FT,
  ov,
  LR_cp_measure,
  ptf = c("M", "C_M", "EDC", "C_EDC")
)
```

Arguments

`data`	A (T x N) matrix or data.frame containing the N time series over period T
`FT`	Window size.
`ov`	Overlap.
`LR_cp_measure`	object of class LR_cp (only for "C_M" and "C_EDC" asset allocation strategies)
`ptf`	Type of portfolio to be computed. Asset allocation strategies available are: "M" is the Markowitz portfolio, "C_M" is the combined Markowitz portfolio, "EDC" uses the extreme downside correlation and "C_EDC" is the combined extreme downside correlation portfolio

Value

An object of class "ptf_construction" is a list containing the following components for all the time windows considered:

ptf oos value a vector of out of sample returns.

weigths portfolio weights.

`sample.data` *Data for infoset function*

Description

Contains daily prices of ETFs

Usage

```
sample.data
```

Format

A data frame with 3174 rows and 44 columns

Source

Created in-house to serve as an example

Examples

```
data(sample.data)
```

`sample.data.ts` *Data with time points for portfolio construction using the LR_cp measure*

Description

Contains daily prices of ETFs

Usage

```
sample.data.ts
```

Format

A data frame with 3175 rows and 45 columns

Source

Created in-house to serve as an example

Examples

```
data(sample.data.ts)
```

summary_ptf *Plot methods for a ptf_construction object*

Description

Plot methods for a ptf_construction object

Usage

```
summary_ptf(ptf.oos.values)
```

Arguments

ptf.oos.values	object of class ptf_construction

Value

summary of oos portfolio values

tail_mixture *Function to find the most-left distribution set.*

Description

An adaptive clustering algorithm identifies sub-groups of gross returns at each iteration by approximating their distribution with a sequence of two-component log-normal mixtures.

Usage

```
tail_mixture(y, shift, n_it, plot)
```

Arguments

y	vector or data frame
shift	double
n_it	integer
plot	option

Value

data object

Index