Package: FARS (via r-universe)

May 18, 2026

Type Package

Title Factor-Augmented Regression Scenarios

Version 0.8.0

Maintainer Gian Pietro Bellocca <gbellocc@est-econ.uc3m.es>

Description Provides a comprehensive framework in R for modeling and forecasting economic scenarios based on multi-level dynamic factor model. The package enables users to: (i) extract global and group-specific factors using a flexible multi-level factor structure; (ii) compute asymptotically valid confidence regions for the estimated factors, accounting for uncertainty in the factor loadings; (iii) obtain estimates of the parameters of the factor-augmented quantile regressions together with their standard deviations; (iv) recover full predictive conditional densities from estimated quantiles; (v) obtain risk measures based on extreme quantiles of the conditional densities; (vi) estimate the conditional density and the corresponding extreme quantiles when the factors are stressed.

Depends R (>= 3.5.0)

Imports rlang, magrittr, ggplot2, plotly, sn, nloptr, ellipse, SyScSelection, quantreg, tidyr, dplyr, forcats, MASS, reshape2, stringr, stats,

Suggests R.rsp, devtools, knitr, rmarkdown, markdown, openxlsx, readxl, zoo

VignetteBuilder R.rsp

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.3.2

URL https://arxiv.org/abs/2507.10679

NeedsCompilation no

Author Gian Pietro Bellocca [aut, cre], Ignacio Garrón [aut], Vladimir Rodríguez-Caballero [aut], Esther Ruiz [aut]

Repository https://cran.r-universe.dev

Date/Publication 2026-02-17 11:00:02 UTC

RemoteUrl https://github.com/cran/FARS

RemoteRef HEAD

RemoteSha 876d80b08e7b289edc7acd54a14c1f93c97dd35b

Contents

AIC.fars . . . 3
BIC.fars . . . 4
coef.fars . . . 4
compute_density . . . 5
compute_fars . . . 6
compute_stressed_factors . . . 7
correct_outliers . . . 8
create_scenario . . . 9
dep_variable . . . 10
factors . . . 11
factors.mldfm . . . 11
fitted.fars . . . 12
fitted.mldfm . . . 12
get_distribution . . . 13
get_distribution.fars_density . . . 14
get_ellipsoids . . . 14
get_ellipsoids.fars_scenario . . . 15
get_mldfm_list . . . 15
get_mldfm_list.mldfm_subsample . . . 16
get_mldfm_model . . . 16
get_mldfm_model.mldfm_subsample . . . 17
get_quantile_levels . . . 17
get_quantile_levels.fars . . . 18
get_rq_model . . . 18
get_rq_model.fars . . . 19
get_sigma_list . . . 19
inflation_data . . . 20
loadings . . . 21
loadings.mldfm . . . 21
logLik.fars . . . 22
mf_data . . . 22
mldfm . . . 23
mldfm_subsampling . . . 24
plot.fars . . . 26
plot.fars_density . . . 27
plot.fars_scenario . . . 27
plot.mldfm . . . 28
plot.mldfm_subsample . . . 29

predict.fars 29
print.fars 30
print.fars_density 30
print.fars_scenario 31
print.mldfm 31
print.mldfm_subsample 32
quantile_risk 32
residuals.fars 33
residuals.mldfm 34
summary.fars 34
summary.fars_density 35
summary.fars_scenario 35
summary.mldfm 36
summary.mldfm_subsample 36

Index **37**

`AIC.fars` *AIC for a* `fars` *Object*

Description

Computes AIC values for each quantile regression stored in a `fars` object.

Usage

```
## S3 method for class 'fars'
AIC(object, ..., k = 2)
```

Arguments

`object`	An object of class `fars`.
`...`	Additional arguments passed to the underlying `AIC()` method.
`k`	Numeric. Penalty per parameter (default 2), as in `stats::AIC()`.

Value

A named numeric vector of AIC values, one per quantile level.

Examples

```
fars_result <- compute_fars(dep_variable = rnorm(100),
                            factors = matrix(rnorm(100 * 3), ncol = 3))
AIC(fars_result)
```

BIC.fars *BIC for a* fars *Object*

Description

Computes BIC values for each quantile regression stored in a fars object.

The number of observations used in the BIC penalty term is computed as periods - h, reflecting the effective sample size of the h-step-ahead dynamic quantile regression.

Usage

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

Arguments

object	An object of class fars.
...	Additional arguments passed to the underlying logLik() method.

Value

A named numeric vector of BIC values, one per quantile level.

Examples

```
fars_result <- compute_fars(
  dep_variable = rnorm(100),
  factors = matrix(rnorm(100 * 3), ncol = 3),
  h = 1
)
BIC(fars_result)
```

coef.fars *Coefficients for* fars *Object*

Description

Returns a matrix of estimated coefficients from all quantile regressions stored in a fars object.

Usage

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

Arguments

`object`	An object of class `fars`.
`...`	Additional arguments (ignored).

Value

A numeric matrix with one column per quantile level and one row per coefficient. Column names correspond to quantile levels (e.g. `0.05`, `0.25`, ...), and row names to coefficient names.

Examples

```
fars_result <- compute_fars(dep_variable = rnorm(100),
                            factors = matrix(rnorm(100 * 3), ncol = 3))
coef(fars_result)
```

`compute_density` *Compute Skew-t Densities from Quantiles*

Description

Computes the skew-t density from a matrix of quantiles. It allows for both linear and nonlinear optimization methods.

Usage

```
compute_density(
  quantiles,
  levels = c(0.05, 0.25, 0.5, 0.75, 0.95),
  est_points = 512,
  random_samples = 5000,
  support = c(-10, 10),
  nl = FALSE,
  seed = NULL
)
```

Arguments

`quantiles`	A matrix of quantiles. Each row represents a time observation, and each column corresponds to a quantile level.
`levels`	A numeric vector of the quantile levels corresponding to the columns of the quantile matrix (default: c(0.05, 0.25, 0.50, 0.75, 0.95)).
`est_points`	Integer. The number of evaluation points for the estimated density (default: 512).
`random_samples`	Integer. The number of random samples to draw from the fitted skew-t distribution (default: 5000).

support	Numeric vector of length 2. Defines the lower and upper limits of the density evaluation range. Used with est_points to create the evaluation grid. Default: c(-10, 10).
nl	Logical. If TRUE, uses nonlinear optimization via nloptr; if FALSE (Default), uses linear optimization via optim.
seed	Optional integer to set the random seed for reproducibility.

Value

An object of class "fars_density", which is a list containing:

density A matrix of estimated densities for each time period (rows) across estimation points (columns).

distribution A matrix of random draws from the fitted skew-t distribution for each time period.

optimization The optimization method used (either 'nloptr' or 'optim').

eval_points The sequence of evaluation points used to compute the density. Useful for plotting.

Examples

```
quantiles <- matrix(rnorm(500, mean = 0, sd = 1), nrow = 100, ncol = 5)
density_result <- compute_density(quantiles, seed = 42)
```

---

compute_fars *Compute Factor Augmented Quantile Regressions*

Description

Performs quantile regressions of a dependent variable on factors estimates.

Usage

```
compute_fars(dep_variable, factors, h = 1, edge = 0.05)
```

Arguments

dep_variable	Numeric vector of length T representing the dependent variable (e.g., GDP growth, inflation).
factors	Numeric matrix or data frame of dimension T x r, containing factor estimates.
h	Integer (>= 1). Lag order used in the regression (default = 1)
edge	Numeric value specifying the trimming amount applied to the outermost quantiles. Default is 0.05.

Value

An object of class `fars`, which is a list containing:

`models` List of five S3 `quantreg::rq` fitted objects named `tau_0.xx`.

`h` The forecast horizon used.

`levels` The vector of quantile levels effectively estimated (`c(edge, 0.25, 0.50, 0.75, 1 - edge)`).
.

`periods` Integer. The number of fitted periods

`n_factors` Integer. The number of factors included in the regression.

`call` The matched function call.

Examples

```
set.seed(123)
T <- 100; r <- 3
Y <- rnorm(T)
F <- matrix(rnorm(T * r), T, r)
fars_result <- compute_fars(dep_variable = Y, factors = F, h = 1, edge = 0.05)
```

---

compute_stressed_factors

Compute Stressed Factors

Description

Computes stressed factors.

Usage

```
compute_stressed_factors(
  dep_variable,
  factors,
  ellipsoids,
  h = 1,
  qtau = 0.05,
  direction = c("min", "max")
)
```

Arguments

`dep_variable`	Numeric vector of length T representing the dependent variable (e.g., GDP growth, inflation).
`factors`	Numeric matrix or data frame of dimension T x r, containing factor estimates.
`ellipsoids`	List of matrices. Each matrix represents a stressed ellipsoid for a given time period.

h	Integer (>= 1). Lag order used in the regression (default = 1)
qtau	Numeric between 0 and 1. The quantile level used in quantile regression (default = 0.05).
direction	Character, either "min" or "max". Determines whether to minimize or maximize

Value

A numeric matrix of dimension T x r containing the stressed factors for each time period.

Examples

```
set.seed(42)
T <- 100; r <- 3
Y <- rnorm(T)
F <- matrix(rnorm(T * r), T, r)
E <- replicate(T, matrix(rnorm(50 * r), nrow = 50, ncol = r), simplify = FALSE)
stressed_factors <- compute_stressed_factors(Y, F, E, h = 1, qtau = 0.05, direction = "min")
```

---

correct_outliers *Correct Dataset Outliers*

Description

This function identifies and corrects outliers in a dataset using principal component analysis (PCA). It scales the data, performs PCA, computes idiosyncratic components, and replaces values that fall outside a defined outlier threshold with the median of 5 previous values. The outlier threshold is determined using the interquartile range (IQR) method.

Usage

```
correct_outliers(data, r)
```

Arguments

data	A numeric matrix or data frame where rows represent observations and columns represent variables.
r	An integer specifying the number of principal components to use for PCA.

Value

A list containing:

data	A matrix with corrected data where outliers are replaced by the median of previous values.
outliers	A binary matrix (same dimensions as the input data) indicating the position of outliers.

Examples

```
data <- matrix(rnorm(100), nrow = 10, ncol = 10)
result <- correct_outliers(data, r = 3)
corrected_data <- result$data
outliers_matrix <- result$outliers
```

`create_scenario` *Create Stressed Scenarios*

Description

Constructs hyperellipsoids around MLDFM factor estimates using subsampling. These regions capture estimation uncertainty and allow for the simulation of stressed scenarios.

Usage

```
create_scenario(model, subsamples, alpha = 0.95, fpr = FALSE)
```

Arguments

`model`	An object of class `mldfm`, containing the factor estimates.
`subsamples`	An object of class `mldfm_subsample`, returned by `mldfm_subsampling`.
`alpha`	Numeric. Confidence level for the hyperellipsoids (e.g., 0.95).
`fpr`	Logical. If `TRUE`, uses FPR Gamma (Fresoli, Poncela, Ruiz, 2024); otherwise, uses standard time-varying Gamma.

Value

An object of class `fars_scenario`, which is a list containing:

ellipsoids List of matrices defining the hyperellipsoids at each time.

center Matrix of factor estimates (centers of the ellipsoids).

sigma List of covariance matrices used to define the ellipsoids.

periods Number of time observations.

n_points Number of points used to define each ellipsoid.

alpha Confidence level for the hyperellipsoids.

call Function call.

Examples

```
data <- matrix(rnorm(100*300), nrow = 100, ncol = 300)
block_ind <- c(150, 300)
global = 1
local <- c(1, 1)
mldfm_result <- mldfm(data, blocks = 2, block_ind = block_ind,
global = global, local = local)
mldfm_subsampling_result <- mldfm_subsampling(data, blocks = 2,
block_ind = block_ind, global = global,
local = local, n_samples = 100, sample_size = 0.9)
scenario <- create_scenario(mldfm_result, mldfm_subsampling_result,
alpha = 0.95)
```

---

dep_variable *US GDP Growth Series*

Description

Quarterly US GDP growth series used as the dependent variable in the replication.

Usage

```
data(dep_variable)
```

Format

A time series object with 59 quarterly observations.

Details

The original series contains quarterly GDP levels for 63 countries. For replication, all series are converted to log-differenced annualized growth rates (`diff(log(x)) * 400`). From this dataset, the U.S. series is extracted and the first observation dropped to obtain 59 observations in total.

Source

Replication materials of González-Rivera et al. (2024).

factors *Generic Function to Extract Estimated Factors*

Description

Generic Function to Extract Estimated Factors

Usage

```
factors(x, ...)
```

Arguments

x	An object from which to extract the estimated factors.
...	Additional arguments.

Value

A matrix of estimated factors.

factors.mldfm *Extract Estimated Factors from a* mldfm *Object*

Description

Extract Estimated Factors from a mldfm Object

Usage

```
## S3 method for class 'mldfm'
factors(x, ...)
```

Arguments

x	An object of class mldfm.
...	Further arguments (ignored).

Value

A matrix containing the estimated factors.

Examples

```
mldfm_result <- mldfm(data = matrix(rnorm(100 * 5), 100, 5), blocks = 1, global = 2)
factors(mldfm_result)
```

fitted.fars *Fitted Values for* fars *Object*

Description

Returns a matrix of fitted values from all quantile regressions stored in a fars object.

Usage

```
## S3 method for class 'fars'
fitted(object, ...)
```

Arguments

object	An object of class fars.
...	Additional arguments (ignored).

Value

A numeric matrix with one column per quantile level and as many rows as observations used in the fit.

Examples

```
fars_result <- compute_fars(dep_variable = rnorm(100),
                            factors = matrix(rnorm(100 * 3), ncol = 3))
fitted(fars_result)
```

fitted.mldfm *Extract Fitted Values from a* mldfm *Object*

Description

Extract Fitted Values from a mldfm Object

Usage

```
## S3 method for class 'mldfm'
fitted(object, ...)
```

Arguments

object	An object of class mldfm.
...	Further arguments (ignored).

Value

A matrix containing the fitted values.

Examples

```
mldfm_result <- mldfm(data = matrix(rnorm(100 * 5), 100, 5), blocks = 1, global = 2)
fitted(mldfm_result)
```

`get_distribution` *Generic Function to Extract Distribution*

Description

Generic Function to Extract Distribution

Usage

```
get_distribution(object, ...)
```

Arguments

`object`	An object from which to extract the distribution.
`...`	Additional arguments (ignored).

Value

A matrix containing the random draws from the fitted skew-t distribution.

Examples

```
fars_density_result <- compute_density(
  quantiles = matrix(rnorm(100 * 5), nrow = 100, ncol = 5)
)
get_distribution(fars_density_result)
```

get_distribution.fars_density *Extract Distribution from a* fars_density *Object*

Description

Extracts the distribution from a fars_density object.

Usage

```
## S3 method for class 'fars_density'
get_distribution(object, ...)
```

Arguments

object	An object of class fars_density.
...	Further arguments (ignored).

Value

A matrix containing the random draws from the fitted skew-t distribution if available, otherwise NULL.

Examples

```
fars_density_result <- compute_density(quantiles = matrix(rnorm(100 * 5), nrow = 100, ncol = 5))
get_distribution(fars_density_result)
```

get_ellipsoids *Generic Function to Extract Ellipsoids*

Description

Generic Function to Extract Ellipsoids

Usage

```
get_ellipsoids(x, ...)
```

Arguments

x	An object from which to extract the ellipsoids
...	Additional arguments.

Value

A list of matrices defining the ellipsoids at each time.

get_ellipsoids.fars_scenario *Get Ellipsoids from a* fars_scenario *Object.*

Description

Returns the list of ellipsoids from a fars_scenario object.

Usage

```
## S3 method for class 'fars_scenario'
get_ellipsoids(x, ...)
```

Arguments

x	An object of class fars_scenario.
...	Additional arguments (ignored).

Value

A list of matrices defining the ellipsoids at each time.

get_mldfm_list *Generic Function to Extract List of MLDFMs*

Description

Generic Function to Extract List of MLDFMs

Usage

```
get_mldfm_list(x, ...)
```

Arguments

x	An object from which to extract the MLDFMs.
...	Additional arguments.

Value

A list of mldfm objects

get_mldfm_list.mldfm_subsample *Extract List of MLDFMs from a* mldfm_subsample *Object*

Description

Returns the list of all mldfm stored in a mldfm_subsample object.

Usage

```
## S3 method for class 'mldfm_subsample'
get_mldfm_list(x, ...)
```

Arguments

x	An object of class mldfm_subsample.
...	Additional arguments (ignored).

Value

A list of mldfm objects.

get_mldfm_model *Generic Function to Extract a Specific* mldfm *Object*

Description

Generic Function to Extract a Specific mldfm Object

Usage

```
get_mldfm_model(x, index, ...)
```

Arguments

x	An object from which to extract a mldfm Object
index	Integer. The position of the desired model
...	Additional arguments.

Value

A single mldfm object.

get_mldfm_model.mldfm_subsample *Extract a Specific* mldfm *Object from a* mldfm_subsample *Object*

Description

Returns the mldfm object at the specified position in a mldfm_subsample object.

Usage

```
## S3 method for class 'mldfm_subsample'
get_mldfm_model(x, index, ...)
```

Arguments

x	An object of class mldfm_subsample.
index	Integer. The position of the desired model (between 1 and n_samples).
...	Additional arguments (ignored).

Value

A single mldfm object.

get_quantile_levels *Generic Function to Extract Quantile Levels*

Description

Generic Function to Extract Quantile Levels

Usage

```
get_quantile_levels(x, ...)
```

Arguments

x	An object from which to extract the quantile levels
...	Additional arguments.

Value

A vector of quantile levels.

get_quantile_levels.fars *Extract Quantile Levels from a* fars *Object*

Description

Returns the quantile levels from an object of class fars.

Usage

```
## S3 method for class 'fars'
get_quantile_levels(x, ...)
```

Arguments

x	An object of class fars, typically the result of a computation such as compute_fars.
...	Additional arguments (ignored).

Value

A vector of quantile levels stored within the fars object.

Examples

```
fars_result <- compute_fars(dep_variable = rnorm(100),
                            factors = matrix(rnorm(100 * 3), ncol = 3))
get_quantile_levels(fars_result)
```

get_rq_model *Generic Function to Extract a Specific* rq *Object*

Description

Generic Function to Extract a Specific rq Object

Usage

```
get_rq_model(x, tau, ...)
```

Arguments

x	An object from which to extract a rq Object
tau	Numeric scalar in (0, 1) indicating the desired quantile.
...	Additional arguments.

Value

A single rq object.

---

get_rq_model.fars *Extract a Specific* rq *Object from a* fars *Object*

Description

Returns the rq fit stored inside a fars object at the requested quantile.

Usage

```
## S3 method for class 'fars'
get_rq_model(x, tau, ...)
```

Arguments

x	An object of class mldfm_subsample.
tau	Numeric scalar in (0, 1) indicating the desired quantile.
...	Additional arguments (ignored).

Value

A single rq object.

Examples

```
fars_result <- compute_fars(dep_variable = rnorm(100),
                            factors = matrix(rnorm(100 * 3), ncol = 3))
get_rq_model(fars_result, tau = 0.05)
```

---

get_sigma_list *Generic Function to Get Sigma List*

Description

Returns the list of covariance matrices used to construct the ellipsoids.

Usage

```
get_sigma_list(x, ...)

get_sigma_list(x, ...)
```

Arguments

x	An object of class fars_scenario.
...	Additional arguments (ignored).

Value

A list of covariance matrices.

A list of covariance matrices (one per period).

---

inflation_data *European Countries Inflation Series*

Description

Monthly inflation series for 38 European countries

Usage

```
data(inflation_data)
```

Format

A numeric matrix with 239 rows and 38 columns.

Details

Derived from the Excel file *inflation.xlsx* included in inst/extdata/. The original series contains monthly HCPI series for 38 European countries. For replication, HCPI prices are transformed into annualized month-on-month (mom) inflation, with each inflation series sequentially cleaned of seasonal effects and outliers

Source

Ha, Kose, and Ohnsorge (2023)

loadings *Generic Function to Extract Factor Loadings*

Description

Generic Function to Extract Factor Loadings

Usage

```
loadings(x, ...)
```

Arguments

x	An object from which to extract the factor loadings.
...	Additional arguments.

Value

A matrix of factor loadings.

loadings.mldfm *Extract Factor Loadings from a* mldfm *Object*

Description

Extract Factor Loadings from a mldfm Object

Usage

```
## S3 method for class 'mldfm'
loadings(x, ...)
```

Arguments

x	An object of class mldfm.
...	Further arguments (ignored).

Value

A matrix containing the estimated factor loadings.

Examples

```
mldfm_result <- mldfm(data = matrix(rnorm(100 * 5), 100, 5), blocks = 1, global = 2)
loadings(mldfm_result)
```

logLik.fars *Log-Likelihoods for* fars *Object*

Description

Returns the log-likelihood for a single quantile regression stored in a fars object (selected via tau).

Usage

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

Arguments

object	An object of class fars.
tau	Numeric. Quantile level to select (e.g. 0.50).
...	Additional arguments (ignored).

Value

An object of class "logLik", as returned by the underlying quantile regression model.

Examples

```
fars_result <- compute_fars(dep_variable = rnorm(100),
                            factors = matrix(rnorm(100 * 3), ncol = 3))
logLik(fars_result, tau = 0.50)
```

mf_data *Macro-Financial Database*

Description

Macro-financial variables used in the replication exercise.

Usage

```
data(mf_data)
```

Format

A numeric matrix with 59 rows and 519 columns.

Details

The original dataset contains 624 variables. For replication, the first 519 variables are selected, converted to a numeric matrix, and outliers are corrected using the function `correct_outliers(..., threshold = 5)` provided in FARS.

Source

Replication materials of González-Rivera et al. (2024).

`mldfm` *Multi-Level Dynamic Factor Model (MLDFM)*

Description

Estimates a Multi-Level Dynamic Factor Modelfrom time series data. Supports both Single-bBlock (DFM) and hierarchical Multi-Block (MLDFM) structures with customizable factor extraction settings.

Usage

```
mldfm(
  data,
  blocks = 1,
  block_ind = NULL,
  global = 1,
  local = NULL,
  middle_layer = NULL,
  method = 0,
  tol = 1e-06,
  max_iter = 1000,
  center = TRUE,
  scale = TRUE,
  verbose = TRUE
)
```

Arguments

`data`	A numeric matrix or data frame containing the time series data. Rows represent time points; columns represent observed variables.
`blocks`	Integer. Number of blocks into which the data is divided.
`block_ind`	Integer vector. End column indices for each block. Must be of length `blocks` and in increasing order.
`global`	Integer. Number of global factors extracted from the entire dataset.
`local`	Integer vector of length `blocks`. Specifies the number of local factors for each block.

middle_layer	Named list. Each name is a string specifying a group of blocks (e.g., "1-3" or "2-3"), and each value is the number of factors to extract.
method	Integer. Method used to initialize the factors: 0 for Canonical Correlation Analysis (CCA), 1 for Principal Component Analysis (PCA).
tol	Numeric. The tolerance level for the residual sum of squares (RSS) minimization process. Used as a convergence criterion.
max_iter	Integer. The maximum number of iterations allowed for the RSS minimization process.
center	Logical. If TRUE (default) center data columns.
scale	Logical. If TRUE (default) scale data columns.
verbose	Logical. If TRUE (default), print a summary of the mldfm.

Value

An object of class mldfm, which is a list containing:

factors Matrix of estimated factors.

loadings Matrix of factor loadings.

residuals Matrix of residuals.

fitted Matrix of fitted values.

method Initialization method used (CCA or PCA).

iterations Number of iterations before convergence.

factors_list List of estimated factors for each node.

call Function call.

Examples

```
mldfm_result <- mldfm(data = matrix(rnorm(100 * 5), 100, 5), blocks = 1, global = 2)
```

---

mldfm_subsampling *Subsampling Procedure for MLDFM Estimation*

Description

Applies the MLDFM estimation to randomly drawn subsamples of the input data.

Usage

```
mldfm_subsampling(
  data,
  blocks = 1,
  block_ind = NULL,
  global = 1,
  local = NULL,
  middle_layer = NULL,
  method = 0,
  tol = 1e-06,
  max_iter = 1000,
  center = TRUE,
  scale = TRUE,
  n_samples = 10,
  sample_size = 0.9,
  seed = NULL
)
```

Arguments

data	A numeric matrix or data frame containing the time series data. Rows represent time points; columns represent observed variables.
blocks	Integer. The number of blocks into which the data is divided.
block_ind	A vector of integers indicating the end index of each block. Must be of length blocks and in increasing order. Required if blocks > 1.
global	Integer. Number of global factors extracted from the entire dataset.
local	Integer vector of length blocks. Specifies the number of local factors for each block.
middle_layer	Named list. Each name is a string specifying a group of blocks (e.g., "1-3" or "2-3"), and each value is the number of factors to extract.
method	Integer. The method used to initialize the factors: 0 for Canonical Correlation Analysis (CCA), 1 for Principal Component Analysis (PCA).
tol	Numeric. The tolerance level for the residual sum of squares (RSS) minimization process. Used as a convergence criterion.
max_iter	Integer. The maximum number of iterations allowed for the RSS minimization process.
center	Logical. If TRUE (default) center data columns.
scale	Logical. If TRUE (default) scale data columns.
n_samples	Number of subsamples to generate.
sample_size	Proportion of the original sample to retain (e.g., 0.9 for 90%).
seed	Optional integer. Seed for reproducibility of the subsampling process. If NULL, random draws will differ each run.

Value

An object of class `mldfm_subsample`, which is a list containing:

- `models`: A list of `mldfm` objects, one for each subsample.
- `n_samples`: Number of subsamples generated.
- `sample_size`: Proportion of the sample used for each subsample.
- `seed`: Seed used for random sampling (if any).
- `call`: Function call.

Examples

```
data <- matrix(rnorm(1000), nrow = 100, ncol = 100)
block_ind <- c(50,100)
local <- c(1, 1)
result <- mldfm_subsampling(data, blocks = 2, block_ind = block_ind, global = 1,
local = local, n_samples = 100, sample_size = 0.9)
```

`plot.fars` *Plot Method for* `fars` *Object*

Description

Generates a line plot of the estimated quantiles from a `fars` object. If `newdata` is `NULL`, the function plots in-sample fitted quantiles; otherwise, it plots predictions computed on `newdata`. The x-axis can be indexed by a provided `dates` vector; if missing, an integer index is used.

Usage

```
## S3 method for class 'fars'
plot(x, newdata = NULL, dates = NULL, ...)
```

Arguments

`x`	An object of class `fars`.
`newdata`	Optional matrix or data frame with one column for the lagged dependent variable and `r` columns for the factors (same `r` used in `compute_fars()`).
`dates`	Optional vector of dates (as `Date` or `zoo::yearqtr`) to use for the x-axis. If not provided, a simple index is used.
`...`	Additional arguments (ignored).

Value

Invisibly returns a `ggplot` object.

plot.fars_density *Plot Method for* fars_density *Object*

Description

Plots the evolution of the estimated density over time as a 3D surface.

Usage

```
## S3 method for class 'fars_density'
plot(x, time_index = NULL, ...)
```

Arguments

x	An object of class fars_density.
time_index	Optional vector for the time axis (default is 1:nrow).
...	Additional arguments (ignored).

plot.fars_scenario *Plot Method for* fars_scenario *Object*

Description

Plots the hyperellipsoid for a given time observation (only for 1D or 2D cases).

Usage

```
## S3 method for class 'fars_scenario'
plot(x, obs = 1, ...)
```

Arguments

x	An object of class fars_scenario.
obs	Integer. Time index to plot (default = 1).
...	Additional arguments (ignored).

`plot.mldfm` *Plot Method for MLDFM object*

Description

Dispatches to specific plot functions for factors, loadings, or residuals.

Usage

```
## S3 method for class 'mldfm'
plot(
  x,
  which = "factors",
  dates = NULL,
  var_names = NULL,
  flip = NULL,
  fpr = FALSE,
  ...
)
```

Arguments

`x`	An object of class `mldfm`.
`which`	What to plot: one of `"factors"` (default), `"loadings"`, or `"residuals"`.
`dates`	Optional vector of dates (as `Date` or `zoo::yearqtr`) to use for the x-axis. If not provided, a simple index (1:N) is used.
`var_names`	Optional vector of variable names to label loadings and residual axis.
`flip`	Optional vector of length equal to the number of factors. Set 1 to flip sign for a specific factor (and related loadings); 0 to leave unchanged.
`fpr`	Logical. If `TRUE`, uses FPR Gamma (Fresoli, Poncela, Ruiz, 2024); otherwise, uses standard time-varying Gamma.
`...`	Additional arguments (ignored)

Value

No return value. Called for plots generation.

plot.mldfm_subsample *Plot Method for* mldfm_subsample *Object*

Description

Plots a histogram of the number of iterations used in each subsample estimation.

Usage

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

Arguments

x	An object of class mldfm_subsample.
...	Additional arguments (ignored).

Value

A ggplot object (invisibly).

predict.fars *Predict Method for* fars *Object*

Description

Computes predictions from all quantile regressions in a fars object. newdata must contain (in this order) the lagged dependent variable column followed by the factor columns. Column names are generated internally as LagY, F1, F2, ..., Fr.

Usage

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

Arguments

object	An object of class fars.
newdata	A matrix or data frame with one column for the lagged dependent variable and r columns for the factors (same r used in compute_fars()).
...	Additional arguments (ignored).

Value

A numeric matrix with one column per quantile level and one row per observation in newdata.

print.fars *Print Method for* fars *Object*

Description

Prints a short summary of the fars object.

Usage

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

Arguments

x	An object of class fars.
...	Additional arguments (ignored).

Value

The input fars object, returned invisibly.

print.fars_density *Print Method for* fars_density *Object*

Description

Displays a brief summary of the fars_density object.

Usage

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

Arguments

x	An object of class fars_density.
...	Additional arguments (ignored).

Value

The input fars_density object, invisibly.

print.fars_scenario *Print Method for* fars_scenario *Object*

Description

Prints a short summary of the FARS scenario object.

Usage

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

Arguments

x	An object of class fars_scenario.
...	Additional arguments (ignored).

Value

The input fars_scenario object, invisibly.

print.mldfm *Print Method for* mldfm *Object*

Description

Prints a short summary of the MLDFM object.

Usage

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

Arguments

x	An object of class mldfm.
...	Additional arguments (ignored).

Value

The input mldfm object, invisibly.

print.mldfm_subsample *Print Method for* mldfm_subsample *Object*

Description

Prints a brief summary of the mldfm_subsample object.

Usage

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

Arguments

x	An object of class mldfm_subsample.
...	Additional arguments (ignored).

Value

The input object object, invisibly.

quantile_risk *Extract Conditional Quantile from* fars_density *Object*

Description

Computes the conditional quantile (e.g., 5th percentile) from a simulated skew-t distribution, The result corresponds to the risk measure (e.g., Growth-at-Risk, Growth-in-Stress etc.).

Usage

```
quantile_risk(density, qtau = 0.05)
```

Arguments

density	An object of class fars_density, which is returned by compute_density()
qtau	A numeric value between 0 and 1 indicating the quantile to extract (e.g., 0.05 for the 5th percentile). Default is 0.05.

Value

A numeric vector of conditional quantiles (one observation for each time period).

Examples

```
quantiles <- matrix(rnorm(500), ncol = 5)
fars_density <- compute_density(quantiles, seed = 42)
GaR <- quantile_risk(fars_density, qtau = 0.05)
```

`residuals.fars` *Residuals for* `fars` *Object*

Description

Returns a matrix of residuals from all quantile regressions stored in a `fars` object.

Usage

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

Arguments

`object`	An object of class `fars`.
`...`	Additional arguments (ignored).

Value

A numeric matrix with one column per quantile level and as many rows as observations used in the fit.

Examples

```
fars_result <- compute_fars(dep_variable = rnorm(100),
                            factors = matrix(rnorm(100 * 3), ncol = 3))
residuals(fars_result)
```

residuals.mldfm *Extract Residuals from a* mldfm *Object*

Description

Extract Residuals from a mldfm Object

Usage

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

Arguments

object	An object of class mldfm.
...	Further arguments (ignored).

Value

A matrix containing the residuals.

Examples

```
mldfm_result <- mldfm(data = matrix(rnorm(100 * 5), 100, 5), blocks = 1, global = 2)
residuals(mldfm_result)
```

summary.fars *Summary Method for* fars *Object*

Description

Displays summaries of all quantile regressions stored in a fars object.

Usage

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

Arguments

object	An object of class fars.
...	Additional arguments (ignored).

Value

The input fars object, returned invisibly.

summary.fars_density *Summary Method for* fars_density *Object*

Description

Displays a complete summary of the fars_density object.

Usage

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

Arguments

object	An object of class fars_density.
...	Additional arguments (ignored).

Value

The input fars_density object, invisibly.

summary.fars_scenario *Summary Method for* fars_scenario *Object*

Description

Provides a detailed summary of the FARS scenario object.

Usage

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

Arguments

object	An object of class fars_scenario.
...	Additional arguments (ignored).

Value

The input fars_scenario object, invisibly.

summary.mldfm *Summary Method for* mldfm *Object*

Description

Provides a complete summary of the MLDFM object.

Usage

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

Arguments

object	An object of class mldfm.
...	Additional arguments (ignored).

Value

The input mldfm object, invisibly.

summary.mldfm_subsample *Summary Method for* mldfm_subsample *Object*

Description

Provides a structured summary of a mldfm_subsample object.

Usage

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

Arguments

object	An object of class mldfm_subsample.
...	Additional arguments (ignored).

Value

The input object object, invisibly.

Index

∗ **datasets**
    dep_variable, 10
    inflation_data, 20
    mf_data, 22

AIC.fars, 3

BIC.fars, 4

coef.fars, 4
compute_density, 5
compute_fars, 6
compute_stressed_factors, 7
correct_outliers, 8
create_scenario, 9

dep_variable, 10

factors, 11
factors.mldfm, 11
fitted.fars, 12
fitted.mldfm, 12

get_distribution, 13
get_distribution.fars_density, 14
get_ellipsoids, 14
get_ellipsoids.fars_scenario, 15
get_mldfm_list, 15
get_mldfm_list.mldfm_subsample, 16
get_mldfm_model, 16
get_mldfm_model.mldfm_subsample, 17
get_quantile_levels, 17
get_quantile_levels.fars, 18
get_rq_model, 18
get_rq_model.fars, 19
get_sigma_list, 19

inflation_data, 20

loadings, 21
loadings.mldfm, 21
logLik.fars, 22

mf_data, 22
mldfm, 23
mldfm_subsampling, 24

plot.fars, 26
plot.fars_density, 27
plot.fars_scenario, 27
plot.mldfm, 28
plot.mldfm_subsample, 29
predict.fars, 29
print.fars, 30
print.fars_density, 30
print.fars_scenario, 31
print.mldfm, 31
print.mldfm_subsample, 32

quantile_risk, 32

residuals.fars, 33
residuals.mldfm, 34

summary.fars, 34
summary.fars_density, 35
summary.fars_scenario, 35
summary.mldfm, 36
summary.mldfm_subsample, 36