Package 'blockwise'

June 24, 2026

Type Package

Title Reduced Modeling for Tabular Data with Blockwise Missingness

Version 0.1.2

Description Supervised learning on tabular data with blockwise missing patterns, using the Blockwise Reduced Modeling (BRM) method of Srinivasan, Currim, and Ram (2025) <doi:10.1287/ijds.2022.9016>. BRM partitions the training data into overlapping subsets based on per-row feature-missing patterns, fits one user-supplied learner per subset with minimal imputation, and at prediction time routes each test instance to the best-matching subset model. The interface is learner-agnostic: any fit-and-predict pair can be plugged in, and convenience specifications are provided for linear models, tree models, random forests, and gradient boosting.

License GPL-3

Language en-US

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Depends R (>= 3.6.0)

Imports stats, VIM, withr

Suggests testthat (>= 3.0.0), knitr, rmarkdown, rpart, ranger, gbm, ggplot2

VignetteBuilder knitr

Config/testthat/edition 3

URL https://github.com/KarAnalytics/blockwise

BugReports https://github.com/KarAnalytics/blockwise/issues

NeedsCompilation no

Author Karthik Srinivasan [aut, cre] (ORCID: <https://orcid.org/0000-0002-1608-6190>), Faiz Currim [aut], Sudha Ram [aut]

Maintainer Karthik Srinivasan <karthiks@ku.edu>

Repository CRAN

Date/Publication 2026-06-24 08:00:27 UTC

Contents

`adult` *UCI Adult income classification dataset*

Description

Census-based binary classification dataset: predict whether a person's annual income exceeds $50,000. Used as the classification demonstration in Srinivasan, Currim, and Ram (2025).

Usage

```
adult
```

Format

A data.frame with roughly 32,561 rows including `salary` (the 0/1-valued response) and typical demographic/employment predictors.

Source

UCI Machine Learning Repository: https://archive.ics.uci.edu/dataset/2/adult

bike *Capital Bikeshare hourly demand data*

Description

Hourly count of rental bikes between 2011 and 2012 in the Capital Bikeshare system with the corresponding weather and seasonal information. Used as the regression demonstration in Srinivasan, Currim, and Ram (2025).

Usage

```
bike
```

Format

A data.frame with roughly 17,380 rows and the following columns:

season, mnth, hr, weekday, weathersit Temporal and weather covariates.

temp, hum, windspeed Numeric weather covariates.

cnt Response: count of total rental bikes for that hour.

Source

UCI Machine Learning Repository: https://archive.ics.uci.edu/dataset/275/bike+sharing+dataset

brm *Fit a Blockwise Reduced Modeling (BRM) ensemble*

Description

BRM partitions the training data into `n_blocks` subsets based on per-row patterns of feature-missingness, fits one instance of the supplied `learner` per subset using only the features observed in that subset, and at prediction time routes each test row to the subset model whose training-time missingness pattern is closest.

Usage

```
brm(
  X,
  y,
  learner = learner_lm(),
  n_blocks = NULL,
  low_threshold = 0.05,
  n_restarts = 5L,
  overlap = TRUE
)
```

Arguments

X	A data.frame of predictors. May contain NA. Categorical predictors should be factors.
y	A numeric vector (regression) or a 0/1 numeric vector (binary classification) of length nrow(X).
learner	A learner specification. Defaults to learner_lm().
n_blocks	Integer number of blocks; if NULL, chosen automatically by choose_num_blocks.
low_threshold	Column-density threshold for including a predictor in a block's model. Default 0.05.
n_restarts	k-means restarts for block assignment. Default 5.
overlap	If TRUE (default) subsets are enlarged via the set-theoretic inclusion rule; if FALSE, the non-overlapping variant is used.

Details

The learner interface is intentionally minimal: any fit / predict pair can be plugged in via learner(), and convenience specs are provided for common families (learner_lm, learner_rpart, etc.).

Value

An object of class "brm".

References

Srinivasan, K., Currim, F., Ram, S. (2025). A Reduced Modeling Approach for Making Predictions With Incomplete Data Having Blockwise Missing Patterns. *INFORMS Journal on Data Science*.

Examples

```
data(bike, package = "blockwise")
bike_miss <- simulate_blockwise_missing(
  bike,
  blocks       = list(c("hum", "windspeed", "weekday"),
                      c("hr", "temp", "weathersit")),
  prop_missing = 0.3
)
X <- bike_miss[, setdiff(names(bike_miss), "cnt")]
y <- bike_miss$cnt
fit <- brm(X, y, learner = learner_lm())
preds <- predict(fit, X)
```

choose_num_blocks *Estimate the number of blocks via the elbow heuristic*

Description

Applies k-means to the binary missingness-indicator matrix for k = 1, ..., k_max and records, for each k, the fraction of still-missing cells after dropping columns that are sparse within each cluster. The curve is monotone-decreasing; BRM picks the k at its elbow.

Usage

```
choose_num_blocks(X, low_threshold = 0.05, n_restarts = 10, k_max = NULL)
```

Arguments

X	A data.frame of predictors; may contain NA.
low_threshold	Fraction below which a column is considered absent in a candidate subset. Default 0.05.
n_restarts	Number of k-means restarts per k. Default 10.
k_max	Upper bound on k. Default min(ncol(X), 50).

Details

Typically called internally by brm when n_blocks = NULL.

Value

A list with n_blocks (the chosen k) and missing_curve (numeric vector of length k_max).

References

Srinivasan, K., Currim, F., Ram, S. (2025). A Reduced Modeling Approach for Making Predictions With Incomplete Data Having Blockwise Missing Patterns. *INFORMS Journal on Data Science*.

house *King County, WA house sales*

Description

The King County house-sales dataset. Used as a regression demonstration in Srinivasan, Currim, and Ram (2025).

Usage

```
house
```

Format

A data.frame with roughly 21,600 rows including `price` (the response) and typical property covariates such as `bedrooms`, `bathrooms`, `sqft_living`, `sqft_lot`, `grade`, and `yr_built`.

Source

Kaggle "House Sales in King County, USA" dataset.

`learner` *Learner specification for BRM*

Description

BRM trains one model per overlapping subset. The learner interface makes that choice user-controlled: supply a `fit` function that takes `(X, y)` and returns a fitted model, and a `predict` function that takes `(model, X_new)` and returns a numeric prediction vector (or a positive-class probability for binary classification).

Usage

```
learner(fit, predict, type = c("regression", "classification"))

learner_lm()

learner_glm_binomial()

learner_rpart(method = "anova", ...)

learner_ranger(...)

learner_gbm(distribution = "gaussian", n.trees = 500, ...)
```

Arguments

`fit`	A function of the form `function(X, y) -> model`.
`predict`	A function of the form `function(model, X_new) -> numeric`.
`type`	Either `"regression"` or `"classification"`.
`method`	rpart split method; one of `"anova"`, `"class"`, etc.
`...`	Additional arguments passed to the underlying fitter.
`distribution`	gbm distribution (e.g. `"gaussian"`, `"bernoulli"`, `"poisson"`).
`n.trees`	Number of trees.

Value

An object of class `"brm_learner"`.

Examples

```
my_learner <- learner(
  fit     = function(X, y) lm(y ~ ., data = cbind(X, y = y)),
  predict = function(m, X_new) predict(m, newdata = X_new),
  type    = "regression"
)
```

`predict.brm` *Predict from a fitted BRM ensemble*

Description

Each row of `newdata` is routed to the block whose training-time missingness center is closest (Euclidean) to the row's missingness pattern. The corresponding block model then predicts on that row, using only that block's feature columns. Any `NA`s remaining in those columns are filled by simple mean/mode imputation against the training reference.

Usage

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

Arguments

`object`	A fitted `brm` object.
`newdata`	A data.frame of predictors. May contain `NA`.
`...`	Unused.

Value

A numeric vector of length `nrow(newdata)`.

`simulate_blockwise_missing`
Simulate a blockwise missing pattern on otherwise complete data

---

Description

Joint-masks groups of columns on randomly chosen rows, optionally adding light column-wise random-NA noise on top. Useful for benchmarking BRM on complete datasets; the default arguments reproduce the simulation design used in Srinivasan, Currim, and Ram (2025).

Usage

```
simulate_blockwise_missing(
  data,
  blocks,
  prop_missing,
  noise = 0.05,
  seed = NULL
)
```

Arguments

`data`	A data.frame.
`blocks`	A list of character vectors; each vector names the columns masked jointly for one block of rows.
`prop_missing`	Proportion of rows affected per block. Either a scalar (applied to each block) or a numeric vector of length `length(blocks)`.
`noise`	Extra per-column random-NA rate applied on top, restricted to columns named in any block. Default 0.05. Set to 0 to disable.
`seed`	Optional integer base seed for reproducibility. If `NULL` (the default), the current RNG state is used and not modified. If supplied, the seed is applied locally via `with_seed` so that the caller's RNG state is preserved.

Value

A data.frame of the same shape as `data`, with `NA`s introduced in the specified pattern.

Examples

```
df <- data.frame(a = 1:100, b = 1:100, c = 1:100, d = 1:100)
simulate_blockwise_missing(df,
  blocks       = list(c("a", "b"), c("c", "d")),
  prop_missing = 0.3,
  seed         = 1234L)
```

Index