Package 'CGMissingDataR'

January 21, 2026

Title Missingness Benchmark (MICE Imputation, Random Forest, kNN)

Version 0.0.0.9000

Description Evaluates predictive performance under feature-level missingness in repeated-measures continuous glucose monitoring-like data. The benchmark injects missing values at user-specified rates, imputes incomplete feature matrices using an iterative chained-equations approach inspired by multivariate imputation by chained equations (MICE) (Melissa J. Azur, Elizabeth A. Stuart, Constantine Frangakis and Philip J. Leaf (2011) <doi:10.1002/mpr.329>), fits Random Forest regression models (Leo Breiman (2001) <doi:10.1023/A:1010933404324>) and k-nearest-neighbor regression models (Zhongheng Zhang (2016) <doi:10.21037/atm.2016.03.37>), and reports mean absolute percentage error (MAPE) and R-squared (R2) across missingness rates.

License GPL (>= 2)

Encoding UTF-8

Roxygen list(markdown = TRUE)

Depends R (>= 4.3)

RoxygenNote 7.3.3

Imports mice,
FNN,
Metrics,
ranger

Suggests testthat (>= 3.0.0),
spelling,
knitr,
rmarkdown

Config/testthat/edition 3

NeedsCompilation no

Language en-US

URL https://github.com/saraswatsh/CGMissingDataR, https://saraswatsh.github.io/CGMissingDataR/

BugReports https://github.com/saraswatsh/CGMissingDataR/issues

LazyData true

VignetteBuilder knitr

Contents

CGMExampleData 2
run_missingness_benchmark 2

Index **4**

`CGMExampleData` *Example dataset for CGMissingData*

Description

A small synthetic dataset intended for examples and tests of `run_missingness_benchmark()`.

Usage

```
CGMExampleData
```

Format

A data frame with 250 rows and 6 variables:

LBORRES Laboratory Observed Result for Glucose (numeric).

TimeSeries Numeric feature representing time series data.

TimeDifferenceMinutes Time difference in minutes between measurements (numeric).

USUBJID Numeric subject identifier.

SiteID Site identifier (character).

Visit Visit label (character).

Examples

```
data("CGMExampleData")
```

`run_missingness_benchmark`
Run missingness benchmark

Description

Benchmarks model performance under feature missingness. The function:

1. Filters to complete cases for `target_col` and `feature_cols` (baseline complete data),
2. Splits into training/validation,
3. Masks feature values at each rate using Bernoulli (cell-wise) missingness,
4. Imputes missing features using MICE on training data and applies the fitted imputation model to validation data via `mice::mice.mids(newdata = ...)` (reduces leakage),
5. Trains Random Forest (`ranger`) and kNN regression (`FNN::knn.reg`),
6. Returns MAPE and R-squared for each model and mask rate.

Feature columns must be numeric (or coercible to numeric without introducing new missing values). This mirrors workflows where features are treated as numeric arrays.

Usage

```
run_missingness_benchmark(
  data,
  target_col,
  feature_cols = NULL,
  mask_rates = c(0.05, 0.1, 0.2, 0.3),
  rf_n_estimators = 200,
  knn_k = 5,
  test_size = 0.2,
  seed = 42
)
```

Arguments

data	A data.frame (or object coercible to data.frame) containing the dataset.
target_col	Single character string: name of the outcome column.
feature_cols	Character vector of feature column names. If NULL, uses all columns except target_col.
mask_rates	Numeric vector in (0, 1): proportion of feature entries to mask per rate.
rf_n_estimators	Integer: number of trees for the random forest.
knn_k	Integer: number of neighbors for kNN regression.
test_size	Numeric in (0, 1): fraction of rows assigned to validation split.
seed	Integer: seed for data split and model reproducibility.

Details

Validation imputation is performed using `mice::mice.mids(newdata = ...)`, which generates imputations for new data according to the model stored in the training `mids` object.

MAPE is computed using `Metrics::mape()` on non-zero targets only to avoid instability when actual values are zero.

Value

A data.frame with columns `MaskRate`, `Model`, `MAPE`, and `R2`.

Author(s)

Shubh Saraswat, Hasin Shahed Shad, and Xiaohua Douglas Zhang

Examples

```
data("CGMExampleData")
run_missingness_benchmark(
  CGMExampleData,
  target_col = "LBORRES",
  feature_cols = c("TimeDifferenceMinutes", "TimeSeries", "USUBJID"),
  mask_rates = c(0.05, 0.10)
)
```

Index

∗ **datasets**
CGMExampleData, 2

CGMExampleData, 2

run_missingness_benchmark, 2