Package 'MIGEE'

December 5, 2024

Type Package

Title Impute Missing Values and Fitting Linear Mixed Effect Model

Version 0.1.0

Maintainer Atanu Bhattacharjee <atanustat@gmail.com>

Description

Implements methods for estimating generalized estimating equations (GEE) with advanced options for flexible modeling and handling missing data. This package provides tools to fit and analyze GEE models for longitudinal data, allowing users to address missingness using a variety of imputation techniques. It supports both univariate and multivariate modeling, visualization of missing data patterns, and facilitates the transformation of data for efficient statistical analysis. Designed for researchers working with complex datasets, it ensures robust estimation and inference in longitudinal and clustered data settings.

License GPL-3

Encoding UTF-8

LazyData true

LazyDataCompression xz

ByteCompile Yes

Author Atanu Bhattacharjee [aut, cre, ctb],
Gajendra Kumar Vishwakarma [aut, ctb],
Neelesh Kumar [aut, ctb]

Imports mice,VIM,ggplot2,lme4,ggeffects,dplyr,readr,reshape2

RoxygenNote 7.3.2

Depends R (>= 2.10)

NeedsCompilation no

Repository CRAN

Date/Publication 2024-12-05 18:50:09 UTC

Contents

`flexImputeModel` *Flexible Missing Data Imputation and Statistical Modeling*

Description

This function provides a comprehensive solution for handling missing data, offering flexible imputation methods and advanced modeling options. It allows users to choose how missing values should be imputed, visualize the missingness patterns, and fit both univariate and multivariate models to the data. The function also offers a convenient workflow for splitting datasets and applying user-specified models.

Usage

```
flexImputeModel(
  data,
  id_col,
  time_col,
  y_col,
  x_col,
  age_col,
  gender_col,
  columns_to_impute = NULL,
  methods = c("pmm", "kNN", "norm", "rf", "norm.nob", "sample"),
  k = 5,
  univariate_vars = NULL,
  multivariate_vars = NULL,
  max_multivariate_vars = 5
)
```

Arguments

`data`	A data frame containing the dataset to be used for analysis. It should include columns for the unique ID, time variable, outcome variable, predictor variables, and any other relevant covariates such as age and gender. The data may contain missing values in columns that require imputation.
`id_col`	A string. The name of the column representing the unique identifier for each subject or observation.
`time_col`	A string. The name of the column representing time, such as the number of days.
`y_col`	A string. The name of the outcome or dependent variable column (e.g., "y_val").
`x_col`	A string. The name of the independent variable column (e.g., "x_val").
`age_col`	A string. The name of the age column (e.g., "Age").
`gender_col`	A string. The name of the gender column (e.g., "Gender").

`columns_to_impute`
: A character vector. The names of the columns that have missing values and need imputation (e.g., `c("x_val", "y_val")`). If `NULL`, all columns with missing data will be imputed.

`methods`
: A character vector. The list of imputation methods to be applied. Defaults to `c("pmm", "kNN", "norm", "rf", "norm.nob", "sample")`.

`k`
: An integer. The number of neighbors to use for k-Nearest Neighbors (kNN) imputation. Defaults to 5.

`univariate_vars`
: A character vector. The variables used for univariate analysis. Defaults to `c("x_val", "Age")`.

`multivariate_vars`
: A character vector. The variables used for multivariate analysis. Defaults to `c("x_val", "Gender")`.

`max_multivariate_vars`
: Maximum number of variables allowed for multivariate analysis is 3.

Details

The function first addresses missing values in specified columns, including 'x_val' and 'y_val'. Users can select from a range of imputation techniques such as predictive mean matching ('pmm'), k-nearest neighbors ('kNN'), normal linear regression imputation ('norm'), random forest imputation ('rf'), or simple random sampling ('sample'), depending on the nature of their data and the desired analysis.

Once missing data is handled, the function splits the dataset into several parts, allowing for more efficient processing or cross-validation. This feature enables users to evaluate imputation and modeling strategies across different portions of the data.

The imputation process is highly customizable, letting users specify which variables to impute and which methods to apply. This flexibility ensures that the imputation strategy aligns with the specific requirements of the analysis.

After handling missing data, the function transforms the 'x_val' variable from a long to a wide format, facilitating modeling of its relationship with 'y_val'. A generalized linear model (GLM) is then applied to examine how these variables relate, providing insights into their interaction. Additionally, the function generates heatmaps that offer a visual representation of the missing and non-missing values within the dataset, helping users understand the distribution of their data.

For statistical modeling, the function includes options for both univariate and multivariate analysis. It fits linear models (LM) and linear mixed-effects models (LME), allowing users to explore relationships between variables of interest while accounting for random effects if needed. Users can specify which variables to include in the models, making it easy to compare different modeling strategies or adjust for potential confounding variables.

Value

A list containing the fitted LM and LME models for both univariate and multivariate analyses, along with generated plots for each method.

Author(s)

Atanu Bhattacharjee, Gajendra Kumar Vishwakarma and Neelesh Kumar

References

Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons.

Examples

```
Results_with_pmm <- flexImputeModel(data = logdata,
                           id_col = "ID",
                           time_col = "Days",
                           y_col = "y_val",
                           x_col = "x_val",
                           age_col = "Age",
                           gender_col = "Gender",
                          columns_to_impute = c("x_val","y_val"),
                          methods = c("pmm"),
                          univariate_vars = c("x_val","Age"),
                          multivariate_vars = c("x_val", "Gender", "trt1"),
                          max_multivariate_vars = 3)
Results_with_pmm$model #summary of Univariate and Multivariate LM and LME model
Results_with_kNN <- flexImputeModel(data = logdata,
                           id_col = "ID",
                           time_col = "Days",
                           y_col = "y_val",
                           x_col = "x_val",
                           age_col = "Age",
                           gender_col = "Gender",
                          columns_to_impute = c("x_val","y_val"),
                          methods = c("kNN"),
                          k = 5,
                          univariate_vars = c("x_val","Age"),
                          multivariate_vars = c("x_val", "Gender", "trt1"),
                          max_multivariate_vars = 3)
Results_with_kNN$model #summary of Univariate and Multivariate LM and LME model
Results_with_norm <- flexImputeModel(data = logdata,
                           id_col = "ID",
                           time_col = "Days",
                           y_col = "y_val",
                           x_col = "x_val",
                           age_col = "Age",
                           gender_col = "Gender",
                          columns_to_impute = c("x_val","y_val"),
                          methods = c("norm"),
                          univariate_vars = c("x_val","Age"),
                          multivariate_vars = c("x_val", "Gender", "trt1"),
                          max_multivariate_vars = 3)
Results_with_norm$model #summary of Univariate and Multivariate LM and LME model
Results_with_rf <- flexImputeModel(data = logdata,
                           id_col = "ID",
```

```
                            time_col = "Days",
                            y_col = "y_val",
                            x_col = "x_val",
                            age_col = "Age",
                            gender_col = "Gender",
                           columns_to_impute = c("x_val","y_val"),
                           methods = c("rf"),
                           univariate_vars = c("x_val","Age"),
                           multivariate_vars = c("x_val", "Gender", "trt1"),
                           max_multivariate_vars = 3)
Results_with_rf$model #summary of Univariate and Multivariate LM and LME model
Results_with_norm.nob <- flexImputeModel(data = logdata,
                            id_col = "ID",
                            time_col = "Days",
                            y_col = "y_val",
                            x_col = "x_val",
                            age_col = "Age",
                            gender_col = "Gender",
                           columns_to_impute = c("x_val","y_val"),
                           methods = c("norm.nob"),
                           univariate_vars = c("x_val","Age"),
                           multivariate_vars = c("x_val", "Gender", "trt1"),
                           max_multivariate_vars = 3)
Results_with_norm.nob$model #summary of Univariate and Multivariate LM and LME model
Results_with_sample <- flexImputeModel(data = logdata,
                            id_col = "ID",
                            time_col = "Days",
                            y_col = "y_val",
                            x_col = "x_val",
                            age_col = "Age",
                            gender_col = "Gender",
                           columns_to_impute = c("x_val","y_val"),
                           methods = c("sample"),
                           univariate_vars = c("x_val","Age"),
                           multivariate_vars = c("x_val", "Gender", "trt1"),
                           max_multivariate_vars = 3)
Results_with_sample$model
 #summary of Univariate and Multivariate LM and LME model
```

`logdata` *Longitudinal clinical data for patients*

Description

Longitudinal clinical data including treatment variables and time-to-event outcomes

Usage

```
data(logdata)
```

Format

A dataframe with multiple rows and 11 variables:

ID ID of subjects

Days Time in days for each recorded event

Age Age of subjects

Gender Gender of subjects (Male/Female)

x_val Covariate values (numerical)

y_val Outcome variable representing time-to-event or measurement (numerical, possibly containing missing data)

trt1 Treatment group 1 (binary, 0/1)

trt2 Treatment group 2 (binary, 0/1)

fac1 Factor 1 (binary, 0/1)

fac2 Factor 2 (binary, 0/1)

Visit Visit number (categorical)

SEX Redundant variable for Gender (Male/Female)

Examples

```
data(logdata)
```

Index