Package: glemb (via r-universe)

October 15, 2026

Title Multiple Imputation Using the General Location Model with the EMB Algorithm

Version 0.1.4

Description Provides multiple imputation for mixed continuous and categorical data using the General Location Model (Olkin and Tate (1961) <doi:10.1214/aoms/1177705052>; Little and Schluchter (1985) <doi:10.1093/biomet/72.3.497>) combined with the Expectation-Maximization with Bootstrapping (EMB) algorithm of Honaker, King and Blackwell (2011) <doi:10.18637/jss.v045.i07>. Categorical variables are specified via the noms option; all other variables are treated as continuous. The package wraps the EM implementation in the 'mix' package and extends it with a bootstrap wrapper to produce proper multiple imputations in the sense of Rubin (1987, ISBN:978-0-471-08705-2).

License GPL-3

Encoding UTF-8

LazyData true

ByteCompile true

Imports mix

Suggests mice, mitml, testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

Depends R (>= 4.0.0)

URL https://github.com/pauldallison/glemb

BugReports https://github.com/pauldallison/glemb/issues

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Paul Allison [aut, cre]

Maintainer Paul Allison <allison@statisticalhorizons.com>

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

Date/Publication 2026-10-15 19:17:03 UTC

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

RemoteRef HEAD

RemoteSha a82755cc9150d9c1c3054b855c919b213a5ceafb

Contents

as.mids.glemb 2
glemb 3
nlsymiss 7
print.glemb 8
summary.glemb 8

Index **10**

`as.mids.glemb` *Convert a glemb object to a mids object*

Description

Converts a `glemb` object to the `mids` format used by the `mice` package, enabling use of `mice::pool()` for results pooling.

Usage

```
as.mids.glemb(x, data, ...)
```

Arguments

`x`	A `glemb` object returned by `glemb()`.
`data`	The original incomplete data frame that was passed to `glemb()`.
`...`	Ignored.

Details

The function stacks the original data (as imputation 0) and all `m` imputed datasets in long format, adds `.imp` and `.id` columns, and calls `mice::as.mids()` to construct the `mids` object.

Value

A `mids` object.

See Also

`mice::as.mids()`, `mice::pool()`

Examples

```
if (requireNamespace("mice", quietly = TRUE)) {
  imp <- glemb(nlsymiss, m = 5,
               noms = c(race, pov, momwork, divorce, gender),
               output = "list", seed = 1)
  mids_obj <- as.mids.glemb(imp, data = nlsymiss)
  fit <- with(mids_obj, lm(anti ~ self + pov + race + childage + divorce +
                           gender + momwork))
  mice::pool(fit)
}
```

---

glemb — *Multiple Imputation Using the General Location Model with the EMB Algorithm*

Description

`glemb()` performs multiple imputation of mixed continuous and categorical data using the General Location Model (GLM) combined with the Expectation-Maximization with Bootstrapping (EMB) algorithm.

Usage

```
glemb(
  data,
  m = 20L,
  noms = NULL,
  idvars = NULL,
  cat.interact = 2L,
  cat.prior = NULL,
  empri = NULL,
  maxits = 1000L,
  seed = NULL,
  output = "mids",
  max.cells = 500000L,
  meanmodel = "main",
  p2s = 1L
)
```

Arguments

`data`	A data frame containing the data to be imputed. All variables not listed in `noms` or `idvars` are treated as continuous. There must be at least one continuous variable. Character variables are not accepted; convert to numeric first.
`m`	Positive integer. Number of imputed datasets to generate. Default is `20`.

`noms` Variable names to treat as categorical (nominal) in the imputation model. Can be supplied quoted or unquoted: `noms = c("pov", "race")` or `noms = c(pov, race)`. These variables may be stored as numeric (e.g. 0/1 binary codes, 1/2/3 integer codes) or as `factor`. All `noms` variables are returned as factors after imputation, ensuring they are treated as categorical in downstream regression calls. At least one variable must be listed.

`idvars` Character vector of column names to be carried through to the output unchanged but excluded from the imputation model. Typically used for subject or case identifiers. Default is `NULL`.

`cat.interact` Integer, either `2` (default) or `3`. Maximum order of interactions among categorical variables in the log-linear model. `2` includes all pairwise interactions; `3` includes all three-way (and lower) interactions. Option `3` should be used with caution as it may greatly increase the number of parameters to be estimated.

`cat.prior` Numeric scalar or `NULL`. Pseudocount for the Dirichlet prior on categorical cell probabilities, used to stabilise estimation with sparse cells. `NULL` (default) selects an automatic value proportional to `1 / n_cells`. `0` applies no prior. A positive value specifies a fixed pseudocount per cell.

`empri` Numeric scalar or `NULL`. Reserved for a future ridge prior on the within-cell continuous covariance matrix, analogous to the `empri` argument in `Amelia::amelia()`. Currently accepted but not applied, as `mix::ecm.mix()` does not expose this parameter. Set to `0` or `NULL`.

`maxits` Positive integer. Maximum number of ECM iterations allowed per bootstrap sample. Default is `1000`. A warning is issued for any bootstrap iteration that reaches this limit without converging.

`seed` Positive integer or `NULL`. Random seed for reproducibility. When supplied, both R's random number generator (used for bootstrap sampling) and `mix`'s internal generator (used for imputation draws) are seeded deterministically as `seed + b` on iteration `b`. Must be a positive integer no greater than 2,000,000,000. Default is `NULL` (no seed set). For reproducibility, it is essential to set the seed within the `glemb()` function. Using `set.seed()` before calling `glemb()` will have no effect on the output.

`output` Character scalar. `"mids"` (default) returns a `mids` object compatible with `mice::pool()` (requires the `mice` package). `"mitml"` returns a `mitml.list` object compatible with `mitml::testEstimates()` (requires the `mitml` package). `"list"` returns a plain list of `m` completed data frames.

`max.cells` Positive integer or `Inf`. Maximum number of cells in the multiway contingency table before `glemb()` stops with an error. The `mix` package allocates arrays proportional to the cell count; with many nominal variables or variables with many levels the cell count can reach millions, exhausting available RAM and crashing R before any imputation begins. The default limit of `500000` catches most dangerous cases while allowing complex but feasible problems. Set to `Inf` to disable the check (expert use only; ensure you have sufficient RAM first).

`meanmodel` Character scalar. Controls the parameterisation of the within-cell continuous variable means. `"main"` (default) includes only the main effects of the categorical variables, giving `1 + sum(K_j - 1)` mean parameters per continuous variable (where `K_j` is the number of levels of the j-th categorical variable).

"saturated" allows a separate mean vector for every cell of the multiway contingency table, giving n_cells mean parameters per continuous variable. The saturated model is more flexible but requires substantially more data and is prone to instability when cells are sparse.

p2s Integer, 0 or 1. Verbosity of console output. 0 = silent; 1 = show progress (default).

Details

Statistical model:

The General Location Model (Olkin & Tate 1961; Little & Schluchter 1985) factorises the joint distribution of continuous variables **y** and categorical variables **x** as:

$$p(\mathbf{y}, \mathbf{x}) = p(\mathbf{y} \mid \mathbf{x})\, p(\mathbf{x})$$

The categorical margin $p(\mathbf{x})$ is modelled with a log-linear model restricted to at most cat.interact-way interactions among the categorical variables. The continuous conditional $p(\mathbf{y} \mid \mathbf{x} = c)$ is multivariate normal with a covariance matrix **shared** across all cells. How the cell means are parameterised depends on meanmodel. With "main" (default) the mean of each continuous variable depends on an intercept plus the main effects of the categorical variables — the same parameterisation as a standard regression with dummy variables. With "saturated" each cell has its own unrestricted mean vector. Both models imply that the conditional mean of every continuous variable is a linear function of all other variables; the saturated model additionally allows all interactions among the categorical variables to enter the means. The conditional distribution of every categorical variable is a logistic regression function of all the other variables.

EMB algorithm:

Proper multiple imputations are generated via the EMB algorithm (Honaker, King & Blackwell 2011):

1. Draw a bootstrap sample (with replacement) from the incomplete data.
2. Run the ECM algorithm (mix::ecm.mix()) on the bootstrap sample to obtain parameter estimates.
3. Draw one set of imputed values from the posterior predictive distribution given those parameters (mix::imp.mix()).
4. Repeat steps 1–3 m times to produce m completed datasets.

Sparsity:

When the number of cells in the multiway table for the categorical variables is large relative to sample size, some cells may contain few or no observations. A Dirichlet prior (controlled by cat.prior) is applied by adding cat.prior pseudo-observations per cell to the bootstrap sample before estimation. glemb() warns when the number of cells exceeds n / 5.

Value

An object of class "glemb", which is a list with components:

imputations A list of m completed data frames.

m Number of imputations.

`call` The matched call.

`categorical` Names of variables treated as categorical.

`continuous` Names of variables treated as continuous.

`idvars` Names of ID variables excluded from the imputation model.

`cat.interact` Interaction order used in the log-linear model.

`cat.prior` Dirichlet pseudocount used (after automatic selection if applicable).

`empri` Ridge prior strength used (after automatic selection if applicable).

`maxits` Maximum ECM iterations per bootstrap sample.

`seed` Seed used, or `NULL`.

`meanmodel` Mean model used: `"main"` or `"saturated"`.

References

Olkin, I. & Tate, R.F. (1961). Multivariate correlation models with mixed discrete and continuous variables. *Annals of Mathematical Statistics*, 32, 448–465. doi:10.1214/aoms/1177705052

Little, R.J.A. & Schluchter, M.D. (1985). Maximum likelihood estimation for mixed continuous and categorical data with missing values. *Biometrika*, 72, 497–512. doi:10.1093/biomet/72.3.497

Honaker, J., King, G. & Blackwell, M. (2011). Amelia II: A program for missing data. *Journal of Statistical Software*, 45(7), 1–47. doi:10.18637/jss.v045.i07

Schafer, J.L. (1997). *Analysis of Incomplete Multivariate Data*. Chapman & Hall.

See Also

`print.glemb()`, `summary.glemb()`, `as.mids.glemb()`

Examples

```
# Impute the NLSY data, treating five variables as categorical
imp <- glemb(nlsymiss, m = 5, noms = c(race, pov, momwork, divorce, gender),
             seed = 123)

# Fit a regression to each completed dataset and pool the results
if (requireNamespace("mice", quietly = TRUE)) {
  fit <- with(imp, lm(anti ~ self + pov + race + childage + divorce +
                        gender + momwork))
  summary(mice::pool(fit))
}

# Quoted names work too; output = "list" returns a glemb object
imp <- glemb(nlsymiss, m = 5,
             noms = c("race", "pov", "momwork", "divorce", "gender"),
             output = "list", seed = 123)
head(imp$imputations[[3]])


# Allow three-way interactions among categorical variables
imp <- glemb(nlsymiss, m = 5, noms = c(race, pov, momwork),
```

```
             cat.interact = 3, output = "list", seed = 123)

# Use a saturated mean model (cell-specific intercepts; more flexible,
# requires more data)
imp <- glemb(nlsymiss, m = 5, noms = c(race, pov, momwork),
             meanmodel = "saturated", output = "list", seed = 123)
```

`nlsymiss` *Children's behavior data from the NLSY, with missing values*

Description

A subsample of 581 children interviewed in 1990 as part of the National Longitudinal Survey of Youth (NLSY), used as a running example in Allison (2001). Five of the nine variables have missing values, and only 225 cases are complete. The data mix continuous and categorical variables, which makes them a natural application of the General Location Model.

Usage

```
nlsymiss
```

Format

A data frame with 581 rows and 9 variables:

anti Antisocial behavior score (0–6).

self Self-esteem score (9–24); 148 missing.

pov Family in poverty (1 = yes, 0 = no); 150 missing.

childage Child's age in years.

divorce Mother divorced (1 = yes, 0 = no).

gender Child's gender (1 = female, 0 = male).

momage Mother's age at the child's birth.

momwork Mother employed (1 = yes, 0 = no); 86 missing.

race Race/ethnicity: 1 = neither Black nor Hispanic, 2 = Black, 3 = Hispanic; 113 missing.

Source

U.S. Bureau of Labor Statistics, National Longitudinal Survey of Youth, Children data, 1990.

References

Allison, P.D. (2001). *Missing Data*. Sage Publications.

Examples

```
summary(nlsymiss)
colSums(is.na(nlsymiss))
```

print.glemb *Print a glemb object*

Description

Print a glemb object

Usage

```
## S3 method for class 'glemb'
print(x, digits = 3L, ...)
```

Arguments

x	A glemb object returned by glemb().
digits	Integer. Number of decimal places for numeric values. Default is 3.
...	Ignored.

Value

x, invisibly.

Examples

```
imp <- glemb(nlsymiss, m = 5, noms = c(race, pov, momwork, divorce, gender),
             output = "list", seed = 1)
print(imp)
```

summary.glemb *Summarise a glemb object*

Description

Prints a summary of the imputation model and the missingness in the first imputed dataset.

Usage

```
## S3 method for class 'glemb'
summary(object, digits = 3L, ...)
```

Arguments

object	A glemb object returned by glemb().
digits	Integer. Number of decimal places for numeric values. Default is 3.
...	Ignored.

Value

`object`, invisibly.

Examples

```
imp <- glemb(nlsymiss, m = 5, noms = c(race, pov, momwork, divorce, gender),
             output = "list", seed = 1)
summary(imp)
```

Index

∗ **datasets**
    nlsymiss, 7

Amelia::amelia(), *4*
as.mids.glemb, 2
as.mids.glemb(), *6*

glemb, 3
glemb(), *2*, *8*

mice::as.mids(), *2*
mice::pool(), *2*, *4*
mitml::testEstimates(), *4*
mix::ecm.mix(), *4*, *5*
mix::imp.mix(), *5*

nlsymiss, 7

print.glemb, 8
print.glemb(), *6*

summary.glemb, 8
summary.glemb(), *6*