Package ‘imputeMulti’

February 18, 2023

**Title**  Imputation Methods for Multivariate Multinomial Data

**Version**  0.8.4

**Description**  Implements imputation methods using EM and Data Augmentation for multinomial data following the work of Schafer 1997 <ISBN: 978-0-412-04061-0>.

**Depends**  R (>= 3.5),

**Imports**  gtools (>= 3.3), methods, parallel, Rcpp (>= 0.11.4),
            data.table (>= 1.14.2)

**License**  GPL-3

**LazyData**  true

**Suggests**  testthat, knitr, R.rsp, covr

**LinkingTo**  Rcpp

**RoxygenNote**  7.1.2

**Encoding**  UTF-8

**VignetteBuilder**  knitr, R.rsp

**Collate**  'RcppExports.R' 'class_imputeMulti.R' 'data-tract2221.R'
            'data_dep_prior_multi.R' 'imputeMulti-package.R'
            'int-count_levels.R' 'int-impute_multinomial.R'
            'int-search_z_Os_y.R' 'int-splitRows.R' 'merge_imputed.R'
            'methods_imputeMulti.R' 'multinomial_data_aug.R'
            'multinomial_em.R' 'multinomial_impute.R' 'multinomial_stats.R'

**NeedsCompilation**  yes

**Author**  Alex Whitworth [aut, cre]

**Maintainer**  Alex Whitworth <whitworth.alex@gmail.com>

**Repository**  CRAN

**Date/Publication**  2023-02-18 20:10:02 UTC

\texttt{R} topics documented:

\begin{verbatim}
data_dep_prior_multi .................................................. 2
imputeMulti-class .................................................... 3
\end{verbatim}
**data_dep_prior_multi**

Data Dependent Prior for Multinomial Distribution

**Description**

Creates a data dependent prior for p-dimensional multinomial distributions using a conjugate prior (eg \( \text{Dirichlet}(\alpha) \)) based on 20

**Usage**

```r
data_dep_prior_multi(dat)
```

**Arguments**

- `dat`  
  A `data.frame`. All variables must be factors

**Value**

A `data.frame` containing identifiers for all possible \( P(Y = y) \) and the associated prior-counts, \( \alpha \)

**References**


**See Also**

`expand.grid`
Description
A multivariate multinomial model imputed by EM or Data Augmentation is represented as a mod_imputeMulti object. A complete dataset and model is represented as an imputeMulti object. Inherits from mod_imputeMulti. Additional slots are supplied for (1) the call to multinomial_impute; (2) the missing and imputed data; and (3) the number of observations with missing values.

Usage
## S4 method for signature 'imputeMulti'
show(object)

get_imputations(object)

## S4 method for signature 'imputeMulti'
get_imputations(object)

n_miss(object)

Arguments
object an object of class "imputeMulti"

Slots
Gcall the call to multinomial_impute
method the modeling method
mle_call the call to the estimation function
mle_iter the number of iterations in estimation
mle_log_lik the final log-likelihood
mle_cp the conjugate prior if any
mle_x_y the MLE estimate of the sufficient statistics and parameters
data a list of the missing and imputed data
nmiss the number of observations with missing data

Objects from the class
Objects are created by calls to multinomial_impute, multinomial_em, or multinomial_data_aug.

See Also
multinomial_impute, multinomial_em, multinomial_data_aug
is.imputeMulti  
Check imputeMulti Class

Description
Function that checks if the target object is an imputeMulti object.

Usage
is.imputeMulti(x)

Arguments
x  any R object.

Value
Returns TRUE if its argument has class "imputeMulti" among its classes and FALSE otherwise.

is.mod_imputeMulti  
Check mod_imputeMulti Class

Description
Function that checks if the target object is a mod_imputeMulti object.

Usage
is.mod_imputeMulti(x)

Arguments
x  any R object.

Value
Returns TRUE if its argument has class "mod_imputeMulti" among its classes and FALSE otherwise.
merge_imputed  

Merge imputed data and original dataset

Description

Merge the imputed dataset from an imputeMulti object with the original dataset. Merging is done by rownames, since imputeMulti maintains row-order during imputation.

Usage

merge_imputed(impute_obj, y, ...)

Arguments

impute_obj An object of class "imputeMulti".
y The dataset from which the missing data was imputed.
... Arguments to be passed to other methods

mod_imputeMulti-class Class "mod_imputeMulti"

Description

A multivariate multinomial model imputed by EM or Data Augmentation is represented as a mod_imputeMulti object. A complete dataset and model is represented as an imputeMulti object. Slots for mod_imputeMulti objects include: (1) the modeling method; (2) the call to the estimation function; (3) the number of iterations in estimation; (4) the final log-likelihood; (5) the conjugate prior if any; (6) the MLE estimate of the sufficient statistics and parameters.

Usage

## S4 method for signature 'mod_imputeMulti'
show(object)

get_parameters(object)

## S4 method for signature 'mod_imputeMulti'
get_parameters(object)

get_prior(object)

## S4 method for signature 'mod_imputeMulti'
get_prior(object)

get_iterations(object)
## S4 method for signature 'mod_imputeMulti'
get_iterations(object)

get_logLik(object)

## S4 method for signature 'mod_imputeMulti'
get_logLik(object)

## S4 method for signature 'mod_imputeMulti'
get_method(object)

## S4 method for signature 'mod_imputeMulti'
get_method(object)

## S4 method for signature 'imputeMulti'
n_miss(object)

### Arguments

- **object**: an object of class "mod_imputeMulti"

### Slots

- **method**: the modeling method
- **mle_call**: the call to the estimation function
- **mle_iter**: the number of iterations in estimation
- **mle_log_lik**: the final log-likelihood
- **mle_cp**: the conjugate prior if any
- **mle_x_y**: the MLE estimate of the sufficient statistics and parameters

### Objects from the class

Objects are created by calls to `multinomial_impute`, `multinomial_em`, or `multinomial_data_aug`.

### See Also

- `multinomial_impute`, `multinomial_em`, `multinomial_data_aug`

---

**multinomial_data_aug**  
*Data Augmentation algorithm for multinomial data*

### Description

Implement the Data Augmentation algorithm for multivariate multinomial data given observed counts of complete and missing data \( (Y_{o,bs} \text{ and } Y_{m,is}) \). Allows for specification of a Dirichlet conjugate prior.
Usage

multinomial_data_aug(
  x_y,
  z_Os_y,
  enum_comp,
  conj_prior = c("none", "data.dep", "flat.prior", "non.informative"),
  alpha = NULL,
  burnin = 100,
  post_draws = 1000,
  verbose = FALSE
)

Arguments

  x_y          A data.frame of observed counts for complete observations.
  z_Os_y       A data.frame of observed marginal-counts for incomplete observations.
  enum_comp    A data.frame specifying a vector of all possible observed patterns.
  conj_prior   A string specifying the conjugate prior. One of c("none", "data.dep", "flat.prior", "non.informative").
  alpha        The vector of counts $\alpha$ for a $\text{Dir}(\alpha)$ prior. Must be specified if conj_prior is either c("data.dep", "flat.prior"). If flat.prior, specify as a scalar. If data.dep, specify as a vector with key matching enum_comp.
  burnin       A scalar specifying the number of iterations to use as a burnin. Defaults to 100.
  post_draws   An integer specifying the number of draws from the posterior distribution. Defaults to 1000.
  verbose      Logical. If TRUE, provide verbose output on each iteration.

Value

An object of class mod_imputeMulti-class.

See Also

multinomial_em, multinomial_impute

Examples

## Not run:
data(tract2221)
x_y <- multinomial_stats(tract2221[,1:4], output= "x_y")
z_Os_y <- multinomial_stats(tract2221[,1:4], output= "z_Os_y")
x_possible <- multinomial_stats(tract2221[,1:4], output= "possible.obs")
imputeDA_mle <- multinomial_data_aug(x_y, z_Os_y, x_possible, n_obs= nrow(tract2221),
  conj_prior = "none", verbose= TRUE)

## End(Not run)
multinomial_em

EM algorithm for multinomial data

Description

Implement the EM algorithm for multivariate multinomial data given observed counts of complete and missing data ($Y_{obs}$ and $Y_{mis}$). Allows for specification of a Dirichlet conjugate prior.

Usage

```r
multinomial_em(
  x_y, 
  z_Os_y, 
  enum_comp, 
  n_obs, 
  conj_prior = c("none", "data.dep", "flat.prior", "non.informative"), 
  alpha = NULL, 
  tol = 5e-07, 
  max_iter = 10000, 
  verbose = FALSE
)
```

Arguments

- `x_y`: A data.frame of observed counts for complete observations.
- `z_Os_y`: A data.frame of observed marginal-counts for incomplete observations.
- `enum_comp`: A data.frame specifying a vector of all possible observed patterns.
- `n_obs`: An integer specifying the number of observations in the original data.
- `conj_prior`: A string specifying the conjugate prior. One of c("none", "data.dep", "flat.prior", "non.informative").
- `alpha`: The vector of counts $\alpha$ for a $Dir(\alpha)$ prior. Must be specified if conj_prior is either c("data.dep", "flat.prior"). If flat.prior, specify as a scalar. If data.dep, specify as a vector with key matching enum_comp.
- `tol`: A scalar specifying the convergence criteria. Defaults to 5e-7
- `max_iter`: An integer specifying the maximum number of allowable iterations. Defaults to 10000.
- `verbose`: Logical. If TRUE, provide verbose output on each iteration.

Value

An object of class `mod_imputeMulti-class`.

See Also

`multinomial_data_aug`, `multinomial_impute`
Examples

```r
## Not run:
data(tract2221)
x_y <- multinomial_stats(tract2221[,1:4], output= "x_y")
z_Os_y <- multinomial_stats(tract2221[,1:4], output= "z_Os_y")
x_possible <- multinomial_stats(tract2221[,1:4], output= "possible.obs")
imputeEM_mle <- multinomial_em(x_y, z_Os_y, x_possible, n_obs= nrow(tract2221),
                               conj_prior= "none", verbose= TRUE)
## End(Not run)
```

---

**multinomial_impute**  Impute Values for missing multinomial values

Description

Impute values for multivariate multinomial data using either EM or Data Augmentation.

Usage

```r
multinomial_impute(
  dat,
  method = c("EM", "DA"),
  conj_prior = c("none", "data.dep", "flat.prior", "non.informative"),
  alpha = NULL,
  verbose = FALSE,
  ...
)
```

Arguments

- `dat` A data.frame. All variables must be factors.
- `method` c("EM", "DA") A string specifying EM or Data Augmentation (DA)
- `conj_prior` A string specifying the conjugate prior. One of c("none", "data.dep", "flat.prior", "non.informative").
- `alpha` The vector of counts $\alpha$ for a $Dir(\alpha)$ prior. Must be specified if conj_prior is either c("data.dep", "flat.prior"). If flat.prior, specify as a scalar. If data.dep, specify as a vector with key matching enum_comp.
- `verbose` Logical. If TRUE, provide verbose output on each iteration.
- `...` Arguments to be passed to other methods

Value

An object of class `imputeMulti-class`
multinomial_stats

References


See Also

data_dep_prior_multi, multinomial_em

Examples

```r
## Not run:
data(tract2221)
imputeEM <- multinomial_impute(tract2221[,1:4], method= "EM",
           conj_prior = "none", verbose= TRUE)
imputeDA <- multinomial_impute(tract2221[,1:4], method= "DA",
           conj_prior = "non.informative", verbose= TRUE)
## End(Not run)
```

multinomial_stats

Multinomial Sufficient Statistics

Description

Calculate observed-data sufficient statistics, marginally-observed summary statistics or enumerate all possible observed patterns from a multivariate multinomial dataset.

Usage

```r
multinomial_stats(dat, output = c("x_y", "z_0s_y", "possible.obs"))
```

Arguments

- **dat**: A data.frame. All variables must be factors.
- **output**: A string specifying the desired output. One of c("x_y", "z_0s_y", "possible.obs"). "x_y" indicates the observed-data sufficient statistics, "z_0s_y" indicates the marginally-observed summary statistics, and "possible.obs" indicates the possible observed patterns.

Value

A data.frame containing either sufficient statistics or possible observed patterns.
Examples

```r
## Not run:
data(tract2221)
obssuff_stats <- multinomial_stats(tract2221, output= "x_y")
marg_obssuff_stats <- multinomial_stats(tract2221, output= "z_0s_y")
## End(Not run)
```

## Summary, imputeMulti-method

**Summary method for class "imputeMulti"**

**Usage**

```r
## S4 method for signature 'imputeMulti'
summary(object, ...)
```

**Arguments**

- `object`: an object of class "imputeMulti"
- `...`: further arguments passed to or from other methods.

## Summary, mod_imputeMulti-method

**Summary method for class "mod_imputeMulti"**

**Usage**

```r
## S4 method for signature 'mod_imputeMulti'
summary(object, ...)
```

**Arguments**

- `object`: an object of class "mod_imputeMulti"
- `...`: further arguments passed to or from other methods.
supDistC  \hspace{1cm} \textit{Calculate the sup of L1 distance between x and y}

Description
sup of L1 distance between x and y

Usage
supDistC(x, y)

Arguments
\begin{itemize}
  \item \textbf{x} \hspace{0.5cm} A numeric vector
  \item \textbf{y} \hspace{0.5cm} A numeric vector
\end{itemize}

Value
a numeric scalar.

tract2221  \hspace{1cm} \textit{Observational data on individuals living in census tract 2221}

Description
A dataset containing attributes of 3974 individuals living in census tract 2221 in Los Angeles County, CA. Data comes from the 5-year American Community Survey with end year 2014. Missing values have been inserted.

Usage
tract2221

Format
A data.frame with 3974 rows and 10 variables. All variables are of class factor:

\begin{itemize}
  \item \textbf{age} \hspace{0.5cm} The individual’s age coded in roughly 5 year age buckets.
  \item \textbf{gender} \hspace{0.5cm} The individual’s gender – Male, Female
  \item \textbf{marital_status} \hspace{0.5cm} The individual’s marital status. Takes one of 5 levels: never_mar never married; married married; mar_apart married but living apart; divorced divorced; and widowed widowed
\end{itemize}
edu_attain  The individual's educational attainment. Takes one of 7 levels: lt_hs less than high school; some_hs completed some high school but did not graduate; hs_grad high school graduate; some_col completed some college but did not graduate; assoc_dec completed an associates degree; ba_deg obtained a bachelors degree; grad_deg obtained a graduate or professional degree

emp_status  The individual's employment status. Takes one of 3 levels: employed individual is in the labor force and employed; unemployed individual is in the labor force and unemployed; not_in_labor_force individual is not in the labor force

nativity  The individual's nativity status. Takes one of 4 values: born_state_residence born in the state of residence; born_other_state born in another US state; born_out_us a US citizen born outside the US; foreigner foreign born

pov_status  The individual's poverty status in the past year. Takes one of 2 levels: below_pov_level below the poverty level; at_above_pov_level at or above the poverty level

geog_mobility  The individual's geographic mobility in the last year. Takes one of 5 values: same house lived in the same house; same_county moved within the same county; same_state moved within the same state; diff_state moved from a different county within the same state; diff_state moved from abroad moved from another country

ind_income  The individual's annual income. Takes one of 9 levels: no_income no income; 1_lt10k income <$10,000; 10k_lt15k $10000-$14999; 15k_lt25k $15000-$24999; 25k_lt35k $25000-$34999; 35k_lt50k $35000-$49999; 50k_lt65k $50000-$64999; 65k_lt75k $65000-$74999; gt75k $75000+

race  The individual's ethnicity.
Index

* datasets
  tract2221, 12

data_dep_prior_multi, 2, 10
expand.grid, 2
get_imputations (imputeMulti-class), 3
get_imputations, imputeMulti-method (imputeMulti-class), 3
get_iterations (mod_imputeMulti-class), 5
get_iterations, mod_imputeMulti-method (mod_imputeMulti-class), 5
get_logLik (mod_imputeMulti-class), 5
get_logLik, mod_imputeMulti-method (mod_imputeMulti-class), 5
get_method (mod_imputeMulti-class), 5
get_method, mod_imputeMulti-method (mod_imputeMulti-class), 5
get_parameters (mod_imputeMulti-class), 5
get_parameters, mod_imputeMulti-method (mod_imputeMulti-class), 5
get_prior (mod_imputeMulti-class), 5
get_prior, mod_imputeMulti-method (mod_imputeMulti-class), 5

imputeMulti, 3, 5
imputeMulti-class, 3
is.imputeMulti, 4
is.mod_imputeMulti, 4

merge_imputed, 5
mod_imputeMulti, 3, 5
mod_imputeMulti-class, 5
multinomial_data_aug, 3, 6, 6, 8
multinomial_em, 3, 6, 7, 8, 10
multinomial_impute, 3, 6–8, 9
multinomial_stats, 10

n_miss (imputeMulti-class), 3
n_miss, imputeMulti-method (mod_imputeMulti-class), 5

show, imputeMulti-method (imputeMulti-class), 3
show, mod_imputeMulti-method (mod_imputeMulti-class), 5
show-imputeMulti (imputeMulti-class), 3
show-mod_imputeMulti (mod_imputeMulti-class), 5
summary, imputeMulti-method, 11
summary, mod_imputeMulti-method, 11
supDistC, 12

tract2221, 12