Package ‘hot.deck’

October 13, 2022

Title Multiple Hot Deck Imputation

Version 1.2

Description
Performs multiple hot-deck imputation of categorical and continuous variables in a data frame.

License MIT + file LICENSE

Depends R (>= 3.5.0)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1.9001

Imports data.table, MASS, mice, tidyr, stats

Suggests knitr, mitools, miceadds, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Jeff Gill [aut, cre],
Skyler Cranmer [aut],
Natalie Jackson [aut],
Andreas Murr [aut],
Dave Armstrong [aut],
Simon Heuberger [aut]

Maintainer Jeff Gill <jgill@american.edu>

Repository CRAN

Date/Publication 2021-08-17 16:40:09 UTC

R topics documented:

hot.deck-package 2
affinity 3
ampData 4
D 5
hd.ord 5
hd2amelia 7

hot.deck 8
is.discrete 9
isq99 10
scaleContinuous 11

Index **12**

hot.deck-package *Multiple Hot Deck Imputation*

Description

This package contains all of the functions necessary to perform multiple hot deck imputation on an input data frame with missing observations using either the “best cell” method (default) or the “probabilistic draw” method as described in Cranmer and Gill (2013). This technique is best suited for missingness in discrete variables, though it also works well for continuous missing observations. The package also offers the possibility to impute data by specifically accounting for unevenly spaced distances between categories in ordinal variables.

Details

Package:	hot.deck
Type:	Package
Version:	1.2
Date:	2021-07-24
License:	What license is it under?

In multiple hot deck imputation, several observed values of the variable with missing observations are drawn conditional on the rest of the data and are used to impute each missing value. The advantage of this class of methods over multiple imputation is that the imputed values are actually draws from the observed data. As such, when discrete variables are imputed with a hot deck method, their discrete properties are maintained.

Two methods for weighting the imputations are provided in this package. The “best cell” [called as “best.cell”] technique uses the degree of affinity between the row with missing data and each potential donor row to generate weights such that rows more closely resembling the row with missingness are more likely to be drawn as donors. The probabilistic draw method is the default method. The “probabilistic draw” [called as “p.draw”] technique is also available. The best cell method draws randomly from the cell of best matches to the row with a missing observation.

Multiple hot deck imputation can also be implemented by specifically accounting for ordinal variables. An ordered probit approach here accounts for unevenly spaced distances and re-estimates ordinal categories that make sense given the data at hand before imputing the data.

Author(s)

Skyler Cranmer, Jeff Gill, Natalie Jackson, Andreas Murr, Dave Armstrong and Simon Heuberger
Maintainer: Dave Armstrong <dave@quantoid.net>

References

Cranmer, S.J. and Gill, J.M.. (2013) "We Have to Be Discrete About This: A Non-Parametric Imputation Technique for Missing Categorical Data." *British Journal of Political Science* 43:2 (425-449). Heuberger, S. (2021) "What People Think: Advances in Public Opinion Measurement Using Ordinal Variables." *PhD Dissertation*.

`affinity` *Affinity Calculation.*

Description

Calculates affinity based on Cranmer and Gill (2013). The function performs the original method (as described in the article) and also a method that takes into account the correlation structure of the observed data that increases efficiency in making matches. Affinity is calculated by first identifying whether two observations are sufficiently 'close' on each variable. Consider the target observation number 1. If observation *i* is close to the target observation on variable *j*, then `A[i,j] = 1` otherwise, it equals zero. Close for two discrete variables is defined by them taking on the same value. Close for continuous variables is taking on a distance no greater than 1 from each other. While this may seem restrictive and arbitrary, arguments exist in the main package function `hot.deck` that allows the user to set how many standard deviations equal a distance of 1 (with the `cutoffSD` argument.

Usage

```
affinity(data, index, column = NULL, R = NULL, weighted = FALSE)
```

Arguments

`data` A data frame or matrix of values for which affinity should be calculated.

`index` A row number identifying the target observation. Affinity will be calculated between this observation and all others in the dataset.

`column` A column number identifying the variable with missing information. This is only needed for the optional correlation-weighted affinity score. The correlation that is used is the correlation of all variables with the focus variable (i.e., the column).

`R` A correlation matrix for `data`.

`weighted` Logical indicating whether or not the correlation-weighted affinity measure should be used.

Value

A number of missing observation-variable combinations-by-number of observations in data matrix of affinity scores.

Examples

```
data(D)
out <- hot.deck(D)
```

ampData *Example data for multiple hot deck imputation with ordinal variables.*

Description

Simulated example data for multiple hot deck imputation with ordinal variables.

Usage

```
data(ampData)
```

Format

A data frame with 1000 observations on the following 20 variables.

`Ind` a numeric binary vector indicating Independent party ID

`Black` a numeric binary vector indicating African-American ethnicity

`Hisp` a numeric binary vector indicating Hispanic ethnicity

`Asian` a numeric binary vector indicating Asian ethnicity

`Empl` a numeric binary vector indicating employment

`Stud` a numeric binary vector indicating students

`Interest` a numeric vector indicating political interest

`Educ` a numeric vector indicating education level

`Religious` a numeric binary vector indicating religious affiliation

`InternetHome` a numeric binary vector indicating the presence of internet at home

`OwnHome` a numeric binary vector indicating home ownership

`Rally` a numeric binary vector indicating attendance at political rallies

`Donate` a numeric binary vector indicating donations

`Moderate` a numeric binary vector indicating moderate political ideology

`Married` a numeric binary vector indicating marriage

`Separated` a numeric binary vector indicating separation

`Dem` a numeric binary vector indicating Democratic party ID, contains missing values

`Male` a numeric binary vector indicating men, contains missing values

`Inc` a numeric vector indicating income, contains missing values

`Age` a numeric vector indicating age, contains missing values

Examples

```
data(ampData)
hd.ord(data = ampData,
      ord = c("Educ", "Interest"),
      evs = c("Dem", "Black", "Empl", "Male", "Inc", "Age"))
```

D *Example data for multiple hot deck imputation.*

Description

Simulated example data for multiple hot deck imputation.

Usage

```
data(D)
```

Format

A data frame with 20 observations on the following 5 variables.

`x1` a numeric vector

`x2` a numeric vector

`x3` a numeric vector

`x4` a numeric vector

`x5` a numeric vector

Examples

```
data(D)
out <- hot.deck(D)
```

hd.ord *Implement hot deck multiple imputation with ordinal variables.*

Description

This function adapts the “hot.deck” function to impute data with missing observations by specifically accounting for ordinal variables. The ordinal variable is regressed on specified meaningful explanatory variables with the `polr` ordered probit approach. The approach assumes an underlying latent continuous variable and estimates the distances between ordinal variable categories. Ordinal levels are replaced with mid-cutpoints of the newly estimated intercepts. Categories that are not supported by the data are dropped. The resulting categories are used to impute the data with multiple hot deck imputation with either the “best cell” method (default) or the “probabilistic draw” method. Any number of ordinal variables can be specified. The specified ordinal variables must not contain missing values.

Usage

```
hd.ord(data, ord, evs, m = 5, method=c("best.cell", "p.draw"),
cutoff=10, sdCutoff=1, optimizeSD = FALSE, optimStep = 0.1, optimStop = 5,
weightedAffinity = FALSE, impContinuous = c("HD", "mice"), IDvars = NULL, ...)
```

Arguments

`data` A data frame with missing values to be imputed using multiple hot deck imputation.

`ord` A vector of ordinal variables to be used on the LHS of the ordered probit regression. Variables must not contain missing values

`evs` A vector of explanatory variables to be used on the RHS of the ordered probit regression. Variables may contain missing values.

`m` Number of imputed datasets required.

`method` Method used to draw donors based on affinity either “best.cell” (the default) or “p.draw” for probabilistic draw.

`cutoff` A numeric scalar such that any variable with fewer than `cutoff` unique non-missing values will be considered discrete and necessarily imputed with hot deck imputation.

`sdCutoff` Number of standard deviations between observations such that observations fewer than `sdCutoff` standard deviations away from each other are considered sufficiently close to be a match, otherwise they are considered too far away to be a match.

`optimizeSD` Logical indicating whether the `sdCutoff` parameter should be optimized such that the smallest possible value is chosen that produces no thin cells from which to draw donors. Thin cells are those where the number of donors is less than `m`.

`optimStep` The size of the steps in the optimization if `optimizeSD` is `TRUE`.

`optimStop` The value at which optimization should stop if it has not already found a value that produces no thin cells. If this value is reached and thin cells still exist, a warning will be returned, though the routine will continue using `optimStop` as `sdCutoff`.

`weightedAffinity`
Logical indicating whether a correlation-weighted affinity score should be used.

`impContinuous` Character string indicating how continuous missing data should be imputed. Valid options are “HD” (the default) in which case hot-deck imputation will be used, or “mice” in which case multiple imputation by chained equations will be used.

`IDvars` A character vector of variable names not to be used in the imputation, but to be included in the final imputed datasets.

`...` Optional additional arguments to be passed down to the `mice` routine.

Value

The output is a list with the following elements:

- dataAn object of class `mi` which contains `m` imputed datasets.
- affinityA matrix of affinity scores see `affinity`.
- donorsA list of donors for each missing observation based on the affinity score.
- drawsThe `m` observations drawn from donors that were used for the multiple imputations.

- max.emp.affNormalization constant for each row of affinity scores; the maximum possible value of the affinity scores if correlation-weighting is used.
- max.the.affNormalization constant for each row of affinity scores; the number of columns in the original data.
- data.origOriginal data fed into the function
- data.orig.na.omitOriginal data without missing values
- data.cutData after cutpoint replacements
- plr.outResults `polr`
- plr.dfResults of `polr` as a data frame
- int.dfsA list of intercepts as data frames
- ord.new.levNew ordinal variable levels
- ord.new.lev.numNumeric version of new ordinal levels

Examples

```
data(ampData)
hd.ord(data = ampData,
      ord = c("Educ", "Interest"),
      evs = c("Dem", "Black", "Empl", "Male", "Inc", "Age"))
```

hd2amelia *Convert hot.deck output to Amelia format.*

Description

Converts the output from hot.deck to the format used by Amelia for use with the Zelig package.

Usage

```
hd2amelia(object)
```

Arguments

`object`	Output from a run of the `hot.deck` function.

Value

An object of class “amelia” that can be used with Zelig.

`hot.deck` *Multiple Hot Deck Imputation.*

Description

This function performs multiple hot deck imputation on an input data frame with missing observations using either the "best cell" method (default) or the "probabilistic draw" method as described in Cranmer and Gill (2013). This technique is best suited for missingness in discrete variables, though it also performs well on continuous missing data.

Usage

```
hot.deck(data, m = 5, method = c("best.cell", "p.draw"), cutoff = 10, sdCutoff = 1,
optimizeSD = FALSE, optimStep = 0.1, optimStop = 5, weightedAffinity = FALSE,
impContinuous = c("HD", "mice"), IDvars = NULL, ...)
```

Arguments

`data` A data frame with missing values to be imputed using multiple hot deck imputation.

`m` Number of imputed datasets required.

`method` Method used to draw donors based on affinity either "best.cell" (the default) or "p.draw" for probabilistic draw.

`cutoff` A numeric scalar such that any variable with fewer than `cutoff` unique non-missing values will be considered discrete and necessarily imputed with hot deck imputation.

`sdCutoff` Number of standard deviations between observations such that observations fewer than `sdCutoff` standard deviations away from each other are considered sufficiently close to be a match, otherwise they are considered too far away to be a match.

`optimizeSD` Logical indicating whether the `sdCutoff` parameter should be optimized such that the smallest possible value is chosen that produces no thin cells from which to draw donors. Thin cells are those where the number of donors is less than `m`.

`optimStep` The size of the steps in the optimization if `optimizeSD` is `TRUE`.

`optimStop` The value at which optimization should stop if it has not already found a value that produces no thin cells. If this value is reached and thin cells still exist, a warning will be returned, though the routine will continue using `optimStop` as `sdCutoff`.

`weightedAffinity`
Logical indicating whether a correlation-weighted affinity score should be used.

`impContinuous` Character string indicating how continuous missing data should be imputed. Valid options are "HD" (the default) in which case hot-deck imputation will be used, or "mice" in which case multiple imputation by chained equations will be used.

`IDvars` A character vector of variable names not to be used in the imputation, but to be included in the final imputed datasets.

`...` Optional additional arguments to be passed down to the `mice` routine.

Value

The output is a list with the following elements:

- dataAn object of class `mi` which contains `m` imputed datasets.
- affinityA matrix of affinity scores see `affinity`.
- donorsA list of donors for each missing observation based on the affinity score.
- drawsThe `m` observations drawn from donors that were used for the multiple imputations.
- max.emp.affNormalization constant for each row of affinity scores; the maximum possible value of the affinity scores if correlation-weighting is used.
- max.the.affNormalization constant for each row of affinity scores; the number of columns in the original data.

Examples

```
data(D)
hot.deck(D)
```

`is.discrete` *Identify whether variables are discrete or continuous.*

Description

Variables are considered discrete if they have fewer unique, non-missing values than `cutoff` or they are factors. Otherwise, variables are considered continuous.

Usage

```
is.discrete(data, cutoff = 10)
```

Arguments

`data` A data frame, matrix or vector of values to be evaluated.

`cutoff` A numeric scalar identifying the cutoff relative to the number of unique, non-missing values for ‘discreteness’.

Value

A logical vector indicating whether variables are discrete (`TRUE`) or continuous `FALSE`.

isq99 *Data from Poe, Tate and Keith 1999.*

Description

Data on Democracy, State Repression and other state-level characteristics

Usage

```
data(isq99)
```

Format

A data frame with 3222 observations on the following 13 variables.

`IDORIGIN` Country Code

`YEAR` Year

`AI` Amnesty International PTS Coding

`SD` State Department Country Report PTS Coding

`POLRT` Freedom House Political Rights Variable

`MIL2` Military Government

`LEFT` Leftist Government

`BRIT` British Colonial Influence

`PCGNP` GNP/capita

`LPOP` Log of population

`DEMOC3` Polity III Democracy

`CWARCOW` COW Civil War

`IWARCOW2` COW Interstate War

References

Steven Poe, C. Neal Tate and Linda Camp Keith. 1999. “Repression of the Human Right to Personal Integrity Revisited: A Global, Cross-National Study Covering the Years 1976-1993”. International Studies Quarterly. 43: 291-313.

`scaleContinuous` *Standardize continuous variables.*

Description

Standardizes (centers and scales) continuous variable in a dataset, leaving discrete variables untouched.

Usage

```
scaleContinuous(data, discrete, sdx = 1)
```

Arguments

`data`	A data frame or matrix of variables to be scaled.
`discrete`	Either a logical vector which is `TRUE` for discrete variables and `FALSE` for continuous ones or a vector of column numbers of discrete variables.
`sdx`	The standard deviation of the columns for the continuous variables.

Value

A data frame with the same dimensions as `data` where the continuous variables are centered and scaled.

Index

∗ **datasets**
    ampData, 4
    D, 5
    isq99, 10
∗ **multiple imputation**
    hot.deck-package, 2
∗ **package**
    hot.deck-package, 2

affinity, 3, *6*, *9*
ampData, 4

D, 5

hd.ord, 5
hd2amelia, 7
hot.deck, 8
hot.deck-package, 2

is.discrete, 9
isq99, 10

scaleContinuous, 11