Package ‘fullRankMatrix’

June 28, 2024

Title Generation of Full Rank Design Matrix

Version 0.1.0

Description Creates a full rank matrix out of a given matrix.
The intended use is for one-hot encoded design matrices that should be used in linear models to ensure that significant associations can be correctly interpreted. However, 'fullRankMatrix' can be applied to any matrix to make it full rank.
It removes columns with only 0's, merges duplicated columns and discovers linearly dependent columns and replaces them with linearly independent columns that span the space of the original columns. Columns are renamed to reflect those modifications.
This results in a full rank matrix that can be used as a design matrix in linear models. The algorithm and some functions are inspired by Kuhn, M. (2008) <doi:10.18637/jss.v028.i05>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.1

Suggests knitr, rmarkdown, igraph, testthat (>= 3.0.0), WeightIt,
caret, plm, spelling

Config/testthat/edition 3

VignetteBuilder knitr

URL `https://github.com/Pweidemueller/fullRankMatrix`

BugReports `https://github.com/Pweidemueller/fullRankMatrix/issues`

Language en-US

NeedsCompilation no

Author Paula Weidemueller [aut, cre, cph]
(<`https://orcid.org/0000-0003-3867-3131`>, @PaulaH_W),
Constantin Ahlmann-Eltze [aut]
(<`https://orcid.org/0000-0002-3762-068X`>, @const_ae)

Maintainer Paula Weidemueller <`paulahw3214@gmail.com`>

Repository CRAN

Date/Publication 2024-06-28 09:10:02 UTC

Contents

---

find_connected_components

Find connected components in a graph

Description

The function performs a depths-first search to find all connected components.

Usage

```
find_connected_components(connections)
```

Arguments

`connections`	a list where each element is a vector with connected nodes. Each node must be either a character or an integer.

Value

a list where each element is a set of connected items.

Examples

```
find_connected_components(list(c(1,2), c(1,3), c(4,5)))
```

---

find_linear_dependent_columns

Find linear dependent columns in a design matrix

Description

Find linear dependent columns in a design matrix

Usage

```
find_linear_dependent_columns(mat, tol = 1e-12)
```

Arguments

`mat`	a matrix
`tol`	a double that specifies the numeric tolerance

Value

a list with vectors containing the indices of linearly dependent columns

See Also

The algorithm and function is inspired by the `internalEnumLC` function in the 'caret' package (GitHub)

Examples

```
mat <- matrix(rnorm(3 * 10), nrow = 10, ncol = 3)
mat <- cbind(mat, mat[,1] + 0.5 * mat[,3])
find_linear_dependent_columns(mat)  # returns list(c(1,3,4))
```

`make_full_rank_matrix` *Create a full rank matrix*

Description

First remove empty columns. Then discover linear dependent columns. For each set of linearly dependent columns, create orthogonal vectors that span the space. Add these vectors as columns to the final matrix to replace the linearly dependent columns.

Usage

```
make_full_rank_matrix(mat, verbose = FALSE)
```

Arguments

`mat`	A matrix.
`verbose`	Print how column numbers change with each operation.

Value

a list containing:

- `matrix`: A matrix of full rank. Column headers will be renamed to reflect how columns depend on each other.
  - `(c1_AND_c2)` If multiple columns are exactly identical, only a single instance is retained.
  - `SPACE_<i>_AXIS<j>` For each set of linearly dependent columns, a space `i` with `max(j)` dimensions was created using orthogonal axes to replace the original columns.
- `space_list`: A named list where each element corresponds to a space and contains the names of the original linearly dependent columns that are contained within that space.

Examples

```
# Create a 1-hot encoded (zero/one) matrix
c1 <- rbinom(10, 1, .4)
c2 <- 1-c1
c3 <- integer(10)
c4 <- c1
c5 <- 2*c2
c6 <- rbinom(10, 1, .8)
c7 <- c5+c6
# Turn into matrix
mat <- cbind(c1, c2, c3, c4, c5, c6, c7)
# Turn the matrix into full rank, this will:
# 1. remove empty columns (all zero)
# 2. merge columns with the same entries (duplicates)
# 3. identify linearly dependent columns
# 4. replace them with orthogonal vectors that span the same space
result <- make_full_rank_matrix(mat)
# verbose=TRUE will give details on how many columns are removed in every step
result <- make_full_rank_matrix(mat, verbose=TRUE)
# look at the create full rank matrix
mat_full <- result$matrix
# check which linearly dependent columns spanned the identified spaces
spaces <- result$space_list
```

validate_column_names *Validate Column Names*

Description

This function checks a vector of column names to ensure they are valid. It performs the following checks:

- The column names must not be NULL.
- The column names must not contain empty strings.
- The column names must not contain NA values.
- The column names must be unique.

Usage

```
validate_column_names(names)
```

Arguments

names	A character vector of column names to validate.

Value

Returns TRUE if all checks pass. If any check fails, the function stops and returns an error message.

Examples

```
validate_column_names(c("name", "age", "gender"))
```

Index