Package ‘holodeck’

May 26, 2020

Title A Tidy Interface for Simulating Multivariate Data

Version 0.2.1

Description Provides pipe-friendly (%>%%) wrapper functions for MASS::mvrnorm() to create simulated multivariate data sets with groups of variables with different degrees of variance, covariance, and effect size.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

biocViews

Imports dplyr, tibble, MASS, purrr, rlang, assertthat

RoxygenNote 7.1.0

URL https://github.com/Aariq/holodeck

BugReports https://github.com/Aariq/holodeck/issues

Suggests testthat, covr, knitr, rmarkdown, mice, ggplot2

VignetteBuilder knitr

NeedsCompilation no

Author Eric Scott [aut, cre] (<https://orcid.org/0000-0002-7430-7879>)

Maintainer Eric Scott <scottericr@gmail.com>

Repository CRAN

Date/Publication 2020-05-26 10:20:03 UTC

R topics documented:

:= .......................... 2
holodeck .......................................................... 2
set_diag .......................................................... 2
sim_cat ......................................................... 3
sim_covar ...................................................... 4
sim_discr ....................................................... 5
sim_missing ................................................... 6
Index

:= \hspace{1cm} \textit{Definition operator}

Description

Internally, this package uses the definition operator, :=, to make assignments that require computing on the LHS.

Arguments

\begin{itemize}
\item \textit{x} \hspace{1cm} \text{An object to test.}
\item \textit{lhs, rhs} \hspace{1cm} \text{Expressions for the LHS and RHS of the definition.}
\end{itemize}

holodeck

\textit{holodeck: A package for simulating multivariate datasets}

Description

The ‘holodeck’ package contains functions for creating “chunks” of variables with different degrees of co-variance (collinearity) and discrimination among groups (i.e. levels of a categorical variable).

Details

What make it ’tidy’? All ‘sim_*‘ functions accept dataframes or tibbles as their first argument and return tibbles, meaning they work with the pipe operator (‘

set_diag

\textit{Pipe friendly wrapper to \texttt{\texttt{diag(x) \leftarrow value}}}’

Description

Pipe friendly wrapper to \texttt{\texttt{diag(x) \leftarrow value}}

Usage

\texttt{set_diag(x, value)}

Arguments

\begin{itemize}
\item \textit{x} \hspace{1cm} \text{a matrix}
\item \textit{value} \hspace{1cm} \text{either a single value or a vector of length equal to the diagonal of ‘x’.}
\end{itemize}
**Value**

a matrix

**Examples**

```r
library(dplyr)
matrix(0,3,3) %>%
set_diag(1)
```

---

<table>
<thead>
<tr>
<th>sim_cat</th>
<th>Simulate categorical data</th>
</tr>
</thead>
</table>

**Description**

This is a simple wrapper that creates a tibble of length ‘n_obs’ with a single column ‘groups’. It will warn if there are fewer than three replicates per group.

**Usage**

```r
sim_cat(.data = NULL, n_obs = NULL, n_groups, name = "group")
```

**Arguments**

- `.data` An optional dataframe. If a dataframe is supplied, simulated categorical data will be added to the dataframe. Either `.data` or ‘n_obs’ must be supplied.
- `n_obs` Total number of observations/rows to simulate if `.data` is not supplied.
- `n_groups` How many groups or treatments to simulate.
- `name` The column name for the grouping variable. Defaults to "group".

**Details**

To-do:
- Make this optionally create multiple categorical variables as being nested or crossed or random

**Value**

a tibble

**See Also**

`sim_covar`, `sim_discr`

Other multivariate normal functions: `sim_covar()`, `sim_discr()`

**Examples**

```r
df <- sim_cat(n_obs = 30, n_groups = 3)
```
**sim_covar**  
*Simulate co-varying variables*

**Description**

Adds a group of variables (columns) with a given variance and covariance to a data frame or tibble.

**Usage**

```r
sim_covar(.data = NULL, n_obs = NULL, n_vars, var, cov, name = NA, seed = NA)
```

**Arguments**

- `.data` An optional dataframe. If a dataframe is supplied, simulated categorical data will be added to the dataframe. Either `.data` or `n_obs` must be supplied.
- `n_obs` Total number of observations/rows to simulate if `.data` is not supplied.
- `n_vars` Number of variables to simulate.
- `var` Variance used to construct variance-covariance matrix.
- `cov` Covariance used to construct variance-covariance matrix.
- `name` An optional name to be appended to the column names in the output.
- `seed` An optional seed for random number generation. If ‘NA’ (default) a random seed will be used.

**Value**

a tibble

**See Also**

- `sim_cat`, `sim_discr`
- Other multivariate normal functions: `sim_cat()`, `sim_discr()`

**Examples**

```r
library(dplyr)
sim_cat(n_obs = 30, n_groups = 3) %>%
sim_covar(n_vars = 5, var = 1, cov = 0.5, name = "correlated")
```
Simulate co-varying variables with different means by group

Description

To-do: make this work with `dplyr::group_by()` instead of `group =`

Usage

```
sim_discr(.data, n_vars, var, cov, group_means, name = NA, seed = NA)
```

Arguments

- `.data`: A dataframe containing a grouping variable column.
- `n_vars`: Number of variables to simulate.
- `var`: Variance used to construct variance-covariance matrix.
- `cov`: Covariance used to construct variance-covariance matrix.
- `group_means`: A vector of the same length as the number of grouping variables.
- `name`: An optional name to be appended to the column names in the output.
- `seed`: An optional seed for random number generation. If `NA` (default) a random seed will be used.

Value

A tibble

See Also

`sim_cat`, `sim_covar`

Other multivariate normal functions: `sim_cat()`, `sim_covar()`

Examples

```
library(dplyr)
sim_cat(n_obs = 30, n_groups = 3) %>%
group_by(group) %>%
sim_discr(n_vars = 5, var = 1, cov = 0.5, group_means = c(-1, 0, 1), name = "descr")
```
sim_missing

Simulate missing values

Description
Takes a data frame and randomly replaces a user-supplied proportion of values with ‘NA’.

Usage
sim_missing(.data, prop, seed = NA)

Arguments
.data A dataframe.
prop Proportion of values to be set to ‘NA’.
seed An optional seed for random number generation. If ‘NA’ (default) a random seed will be used.

Value
A dataframe with NAs

Examples
library(dplyr)
df <- sim_cat(n_obs = 10, n_groups = 2) %>%
sim_covar(n_vars = 10, var = 1, cov = 0.5) %>%
sim_missing(0.05)
Index

:=, 2
holodeck, 2
set_diag, 2
sim_cat, 3, 4, 5
sim_covar, 3, 4, 5
sim_discr, 3, 4, 5
sim_missing, 6