

# 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](mailto:scottericr@gmail.com)>

**Repository** CRAN

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

## R topics documented:

<code>:=</code> . . . . .	2
<code>holodeck</code> . . . . .	2
<code>set_diag</code> . . . . .	2
<code>sim_cat</code> . . . . .	3
<code>sim_covar</code> . . . . .	4
<code>sim_discr</code> . . . . .	5
<code>sim_missing</code> . . . . .	6

**Index**[7](#)


---

:=	<i>Definition operator</i>
----	----------------------------

---

**Description**

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

**Arguments**

x	An object to test.
lhs, rhs	Expressions for the LHS and RHS of the definition.

---

holodeck	<i>holodeck: A package for simulating multivariate datasets</i>
----------	---

---

**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	<i>Pipe friendly wrapper to ‘diag(x) &lt;- value’</i>
----------	---

---

**Description**

Pipe friendly wrapper to ‘diag(x) <- value’

**Usage**

```
set_diag(x, value)
```

**Arguments**

x	a matrix
value	either a single value or a vector of length equal to the diagonal of ‘x’.

**Value**

a matrix

**Examples**

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

---

sim_cat	<i>Simulate categorical data</i>
---------	----------------------------------

---

**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**

```
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**

```
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

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

## Arguments

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

```
library(dplyr)
sim_cat(n_obs = 30, n_groups = 3) %>%
  sim_covar(n_vars = 5, var = 1, cov = 0.5, name = "correlated")
```

---

`sim_discr`*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

<code>.data</code>	A dataframe containing a grouping variable column.
<code>n_vars</code>	Number of variables to simulate.
<code>var</code>	Variance used to construct variance-covariance matrix.
<code>cov</code>	Covariance used to construct variance-covariance matrix.
<code>group_means</code>	A vector of the same length as the number of grouping variables.
<code>name</code>	An optional name to be appended to the column names in the output.
<code>seed</code>	An optional seed for random number generation. If <code>'NA'</code> (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	<i>Simulate missing values</i>
-------------	--------------------------------

---

**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