

# Package: easySimData (via r-universe)

October 17, 2024

**Type** Package

**Title** A Wrapper of 'simdata' Package

**Version** 0.1.0

**Maintainer** Han Zhang <zhangh.ustc@gmail.com>

**Description** Simulating data according to marginal distributions and pairwise correlation. This is a wrapper for the 'simdata' package to make it easier to use.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Imports** dplyr, ggplot2, gridExtra, simdata, stringr

**RoxygenNote** 7.3.0

**NeedsCompilation** no

**Depends** R (>= 2.10)

**LazyData** true

**Author** person) [aut] (``aut'', ``cre"), Han Zhang [cre]

**Date/Publication** 2024-10-16 16:30:05 UTC

**Repository** <https://zhangh12.r-universe.dev>

**RemoteUrl** <https://github.com/cran/easySimData>

**RemoteRef** HEAD

**RemoteSha** be52e0489411acbe86ea6d79de3c4fea6c71825f

## Contents

configureTrial . . . . .	2
df . . . . .	2
plot.easySimData . . . . .	3
simulateTrialData . . . . .	3

<b>Index</b>	<b>6</b>
--------------	----------

---

configureTrial	<i>Convert calls into quantile functions</i>
----------------	--

---

### Description

Calls come from ... in simulateTrialData. For individual level data, empirical quantile will be generated using simdata::quantile\_functions\_from\_data. For calls specifying common distribution families and their parameters in plain text, quantile functions will be generated by quantile functions in stats, e.g., qnorm, qunif, etc. For calls of user-customized quantile functions, it returns those directly.

### Usage

```
configureTrial(..., var_prefix)
```

### Arguments

... calls describing the marginal distributions of endpoints. See ... in simulateTrialData.  
var\_prefix prefix to name variables that users do not give to those.

### Value

a list of quantile functions.

---

df	<i>Example data from simdata package</i>
----	--

---

### Description

A data frame containing the variables as follows:

### Usage

```
data(df)
```

### Format

A data frame with 4709 rows and 7 variables:

**Gender** Gender

**Age** Age

**Race** Race

**Weight** Weight

**BMI** BMI

**BPsys** BPsys

**BPdia** BPdia

**Source**

simdata R package.

**Examples**

```
data(df)
df
```

---

plot.easySimData	<i>Plot data generated by the easySimData package.</i>
------------------	--

---

**Description**

Plot data generated by the easySimData package.

**Usage**

```
## S3 method for class 'easySimData'
plot(x, ...)
```

**Arguments**

x	an object of class easySimData returned by simulateTrialData.
...	ignored.

**Value**

an object returned by gridExtra::grid.arrange, a grid of plots comparing the simulated marginal distribution with the ones specified by users.

---

simulateTrialData	<i>Generate simulation endpoint data for an arm in a trial.</i>
-------------------	---

---

**Description**

Parameters of marginal distribution of endpoints and their target correlation are specified. This function returns endpoint data of one arm. To generate data of multiple arms, one need to call this function for each of the arms.

**Usage**

```
simulateTrialData(
  ...,
  cor_matrix,
  trial_size = 100,
  arm_label = NULL,
  var_prefix = "easySimData"
)
```

**Arguments**

...	calls describing the marginal distributions of endpoints. They can be data frames, e.g., df, or valid R expressions evaluated to be data frames, e.g., data.frame(x = runif(10), y = 1), df %>% filter(x < 1), myFunctionReturnDataFrame(), or simply any quantile functions (see the dist argument of simdata::simdesign_norta).
cor_matrix	a target correlation matrix. See the cor_target_final argument of simdata::simdesign_norta.
trial_size	size of an arm.
arm_label	a character to name the arm. When specified, a column arm will be added to the end of the simulated data frame. NULL by default.
var_prefix	a character as the prefix of endpoints specified by common distributions.

**Value**

A data frame of endpoints.

**Examples**

```
library(dplyr)
data(df)
head(df)

cmat <- matrix(.5, 10, 10)
diag(cmat) <- 1.

## the first argument should always be x if a user-defined quantile function
## is supplied. This is required by the `simdata` package.
user_qfun <- function(x, shape1 = .6, shape2 = 1.2){
  qbeta(x, shape1 = shape1, shape2 = shape2)
}

data <- simulateTrialData(
  df[, c('Gender', 'Age')] %>% dplyr::filter(Age < 50), # an expression with
  # its value a data frame
  df %>% dplyr::select(-c(Gender, Age, Race)), # same as above
  norm(mean = 0.2, sd = 1.2), # distribution name. Note that it is not a valid
  # R function but a symbol.
  binom(prob = .4, size = 1), # spaces are ignore, order does not matter
  norm(mean = -0.4, sd = 0.9), # same distribution with different parameters
```

```
    user_qfun,                # user-defined quantile

    cor_matrix = cmatrix,
    trial_size = 1000,
    arm_label = 'placebo'
  )

## for diagnosis
plot(data)
cor(data %>% dplyr::select(-matches('arm')))

## Note that user_qfun is simply the quantile function of a beta distribution,
## so a preferred way to use easySimData is as follow
data <- simulateTrialData(
  df[, c('Gender', 'Age')] %>% dplyr::filter(Age < 50), # an expression with
                                # its value a data frame
  df %>% dplyr::select(-c(Gender, Age, Race)), # same as above
  norm(mean = 0.2, sd = 1.2), # distribution name. Note that it is not a valid
                                # R function but a symbol.
  binom(prob = .4, size = 1), # spaces are ignored, order does not matter
  norm(mean = -0.4, sd = 0.9), # same distribution with different parameters
  beta(shape1 = .5, shape2 = 1.2), # distribution name

  cor_matrix = cmatrix,
  trial_size = 1000,
  arm_label = 'placebo'
)
```

# Index

\* **datasets**

df, [2](#)

configureTrial, [2](#)

df, [2](#)

plot.easySimData, [3](#)

simulateTrialData, [3](#)