

# Package ‘GlmSimulatorR’

November 4, 2021

**Type** Package

**Title** Creates Ideal Data for Generalized Linear Models

**Version** 0.2.5

**Author** Greg McMahan

**Maintainer** Greg McMahan <gmcmacran@gmail.com>

**Description** Creates ideal data for all distributions in the generalized linear model framework.

**License** GPL-3

**Encoding** UTF-8

**Imports** assertthat, stats, stringr, dplyr, statmod, magrittr, MASS, tweedie, ggplot2, cplm

**RoxygenNote** 7.1.1

**Suggests** testthat (>= 3.0.0), knitr, rmarkdown, covr

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2021-11-04 18:10:02 UTC

## R topics documented:

simulate\_gaussian . . . . . 2

**Index** 6

---

simulate\_gaussian      *Create ideal data for a generalized linear model.*

---

### Description

Create ideal data for a generalized linear model.

### Usage

```
simulate_gaussian(  
  N = 10000,  
  link = "identity",  
  weights = 1:3,  
  xrange = 1,  
  unrelated = 0,  
  ancillary = 1  
)  
  
simulate_binomial(  
  N = 10000,  
  link = "logit",  
  weights = c(0.1, 0.2),  
  xrange = 1,  
  unrelated = 0  
)  
  
simulate_gamma(  
  N = 10000,  
  link = "inverse",  
  weights = 1:3,  
  xrange = 1,  
  unrelated = 0,  
  ancillary = 0.05  
)  
  
simulate_poisson(  
  N = 10000,  
  link = "log",  
  weights = c(0.5, 1),  
  xrange = 1,  
  unrelated = 0  
)  
  
simulate_inverse_gaussian(  
  N = 10000,  
  link = "1/mu^2",  
  weights = 1:3,
```

```

    xrange = 1,
    unrelated = 0,
    ancillary = 0.3333
)

simulate_negative_binomial(
  N = 10000,
  link = "log",
  weights = c(0.5, 1),
  xrange = 1,
  unrelated = 0,
  ancillary = 1
)

simulate_tweedie(
  N = 10000,
  link = "log",
  weights = 0.02,
  xrange = 1,
  unrelated = 0,
  ancillary = 1.15
)

```

### Arguments

N	Sample size. (Default: 10000)
link	Link function. See <a href="#">family</a> for details.
weights	Betas in glm model.
xrange	range of x variables.
unrelated	Number of unrelated features to return. (Default: 0)
ancillary	Ancillary parameter for continuous families and negative binomial. See details.

### Details

For many families, it is possible to pick weights that cause inverse link( $X * weights$ ) to be mathematically invalid. For example, the log link for binomial regression defines  $P(Y=1)$  as  $\exp(X * weights)$  which can be above one. If this happens, the function will error with a helpful message.

The intercept in the underlying link( $Y = X * weights + intercept$ ) is always  $\max(weights)$ . In `simulate_gaussian(link = "inverse", weights = 1:3)`, the model is  $(1/Y) = 1*X1 + 2*X2 + 3*X3 + 3$ .

links

- gaussian: identity, log, inverse
- binomial: logit, probit, cauchit, loglog, cloglog, log, logc, identity
- gamma: inverse, identity, log
- poisson: log, identity, sqrt
- inverse gaussian:  $1/\mu^2$ , inverse, identity, log

- negative binomial: log, identity, sqrt
- tweedie: log, identity, sqrt, inverse

The default link is the first link listed for each family.

ancillary parameter

- gaussian: standard deviation
- binomial: N/A
- gamma: scale parameter
- poisson: N/A
- inverse gaussian: dispersion parameter
- negative binomial: theta.
- tweedie: rho

### Value

A tibble with a response variable and predictors.

### Examples

```
library(GlmSimulator)
library(ggplot2)
library(MASS)

# Do glm and lm estimate the same weights? Yes
set.seed(1)
simdata <- simulate_gaussian()
linearModel <- lm(Y ~ X1 + X2 + X3, data = simdata)
glmModel <- glm(Y ~ X1 + X2 + X3, data = simdata, family = gaussian(link = "identity"))
summary(linearModel)
summary(glmModel)
rm(linearModel, glmModel, simdata)

# If the link is not identity, will the response
# variable still be normal? Yes
set.seed(1)
simdata <- simulate_gaussian(N = 1000, link = "log", weights = c(.1, .2))

ggplot(simdata, aes(x = Y)) +
  geom_histogram(bins = 30)
rm(simdata)

# Is AIC lower for the correct link? For ten thousand data points, depends on seed!
set.seed(1)
simdata <- simulate_gaussian(N = 10000, link = "inverse", weights = 1)
glmCorrectLink <- glm(Y ~ X1, data = simdata, family = gaussian(link = "inverse"))
glmWrongLink <- glm(Y ~ X1, data = simdata, family = gaussian(link = "identity"))
summary(glmCorrectLink)$aic
summary(glmWrongLink)$aic
rm(simdata, glmCorrectLink, glmWrongLink)
```

```
# Does a stepwise search find the correct model for logistic regression? Yes
# 3 related variables. 3 unrelated variables.
set.seed(1)
simdata <- simulate_binomial(N = 10000, link = "logit", weights = c(.3, .4, .5), unrelated = 3)

scopeArg <- list(
  lower = Y ~ 1,
  upper = Y ~ X1 + X2 + X3 + Unrelated1 + Unrelated2 + Unrelated3
)

startingModel <- glm(Y ~ 1, data = simdata, family = binomial(link = "logit"))
glmModel <- stepAIC(startingModel, scopeArg)
summary(glmModel)
rm(simdata, scopeArg, startingModel, glmModel)

# When the response is a gamma distribution, what does a scatter plot between X and Y look like?
set.seed(1)
simdata <- simulate_gamma(weights = 1)
ggplot(simdata, aes(x = X1, y = Y)) +
  geom_point()
rm(simdata)
```

# Index

family, [3](#)

`simulate_binomial (simulate_gaussian)`, [2](#)

`simulate_gamma (simulate_gaussian)`, [2](#)

`simulate_gaussian`, [2](#)

`simulate_inverse_gaussian`  
    `(simulate_gaussian)`, [2](#)

`simulate_negative_binomial`  
    `(simulate_gaussian)`, [2](#)

`simulate_poisson (simulate_gaussian)`, [2](#)

`simulate_tweedie (simulate_gaussian)`, [2](#)