

Package: GlmSimulatorR (via r-universe)

September 1, 2024

Type Package

Title Creates Ideal Data for Generalized Linear Models

Version 1.0.1.999

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 7.2.3

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

VignetteBuilder knitr

Config/testthat/edition 3

Repository <https://gmcmacran.r-universe.dev>

RemoteUrl <https://github.com/gmcmacran/glmsimulator>

RemoteRef HEAD

RemoteSha 4c2828d21de34c90f183fc90983ca44e4e9f638a

Contents

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

```

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

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

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

```

Arguments

N	Sample size. (Default: 10000)
link	Link function. See family for details.
weights	Betas in glm model.
x_range	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()
linear_model <- lm(Y ~ X1 + X2 + X3, data = simdata)
glm_model <- glm(Y ~ X1 + X2 + X3,
  data = simdata,
  family = gaussian(link = "identity")
)
summary(linear_model)
summary(glm_model)
rm(linear_model, glm_model, 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)
glm_correct_link <- glm(Y ~ X1,
```

```
    data = simdata,
    family = gaussian(link = "inverse")
  )
glm_wrong_link <- glm(Y ~ X1,
  data = simdata,
  family = gaussian(link = "identity")
)
summary(glm_correct_link)$aic
summary(glm_wrong_link)$aic
rm(simdata, glm_correct_link, glm_wrong_link)

# 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
)

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

starting_model <- glm(Y ~ 1,
  data = simdata,
  family = binomial(link = "logit")
)
glm_model <- stepAIC(starting_model, scope_arg)
summary(glm_model)
rm(simdata, scope_arg, starting_model, glm_model)

# 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](#)