Package ‘GlmSimulatoR’

March 17, 2020

Type Package

Title Creates Ideal Data for Generalized Linear Models

Version 0.2.1

Author Greg McMahan

Maintainer Greg McMahan <gmcmacran@gmail.com>

Description
Have you ever struggled to find ``good data'' for a generalized linear model? Would you like to test how quickly statistics converge to parameters, or learn how picking different link functions affects model performance? This package creates ideal data for both common and novel generalized linear models so your questions can be empirically answered.

License GPL-3

Encoding UTF-8

LazyData true

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

RoxygenNote 6.1.1

Suggests testthat, knitr, rmarkdown, covr

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2020-03-17 10:10:03 UTC

R topics documented:

simulate_gaussian ................................................. 2

Index 5

1


**simulate_gaussian**  
Create ideal data for a generalized linear model.

Description

Create ideal data for a generalized linear model.

Usage

```r
simulate_gaussian(N = 10000, link = defaultLink,  
weights = defaultWeights, unrelated = 0,  
ancillary = defaultAncillary)
```

```r
simulate_binomial(N = 10000, link = defaultLink,  
weights = defaultWeights, unrelated = 0,  
ancillary = defaultAncillary)
```

```r
simulate_gamma(N = 10000, link = defaultLink,  
weights = defaultWeights, unrelated = 0,  
ancillary = defaultAncillary)
```

```r
simulate_poisson(N = 10000, link = defaultLink,  
weights = defaultWeights, unrelated = 0,  
ancillary = defaultAncillary)
```

```r
simulate_inverse_gaussian(N = 10000, link = defaultLink,  
weights = defaultWeights, unrelated = 0,  
ancillary = defaultAncillary)
```

```r
simulate_negative_binomial(N = 10000, link = defaultLink,  
weights = defaultWeights, unrelated = 0,  
ancillary = defaultAncillary)
```

```r
simulate_tweedie(N = 10000, link = defaultLink,  
weights = defaultWeights, unrelated = 0,  
ancillary = defaultAncillary)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>N</code></td>
<td>Sample size. (Default: 10000)</td>
</tr>
<tr>
<td><code>link</code></td>
<td>Link function. See family for details.</td>
</tr>
<tr>
<td><code>weights</code></td>
<td>Betas in glm model. See details. simulate_binomial: c(.1, .2) All other: c(1, 2, 3)</td>
</tr>
<tr>
<td><code>unrelated</code></td>
<td>Number of unrelated features to return. (Default: 0)</td>
</tr>
<tr>
<td><code>ancillary</code></td>
<td>Ancillary parameter for continuous families and negative binomial. See details.</td>
</tr>
</tbody>
</table>
Details

The gaussian family accepts the links identity, log and inverse. The binomial family accepts the links logit, probit, cauchit, (corresponding to logistic, normal and Cauchy CDFs respectively) log and cloglog (complementary log-log). The gamma family accepts the links inverse, identity and log. The poisson family accepts the links log, identity, and sqrt. The inverse gaussian family accepts the links 1/mu^2, inverse, identity and log. The negative binomial family accepts the links log, identity, and sqrt using glm.nb in MASS. The tweedie family accepts the links log, identity, sqrt and inverse.

Default links are identity for gaussian, logit for binomial, inverse for gamma, log for poisson, 1/mu^2 for inverse gaussian, log for negative binomial and log for tweedie.

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. For P(Y=1) to be between zero and one, weights should be small. In general, the log link is the most troublesome to work with. It is recommended to use weights in the neighborhood of .1 to .3 for log link.

For the inverse gaussian family, the inverse link function needs weights*X to be positive.

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

The continuous families have an ancillary parameter. For the gaussian family, it is standard deviation. Default value is 1. For the gamma family, it is the scale parameter. Default value is .05. For inverse gaussian, it is the dispersion parameter. Default value 1/3. For negative binomial it is theta. Default value 1. For tweedie, it is rho. Default value 1.15. Not used in binomial and poisson.

Value

A tibble with a response variable and predictors.

Examples

```r
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 effects are multiplicative instead of additive,
# will my 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)) +
```

simulate_gaussian

```r
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, 2

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