Package 'GlmSimulatoR'

August 12, 2019

Type Package

Title Creates Ideal Data for Generalized Linear Models

Version 0.1.0

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

RoxygenNote 6.1.1

Suggests testthat, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2019-08-12 07:20:02 UTC

R topics documented:

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 = defaultLink,
  weights = defaultWeights, unrelated = 0,
  dispersion = defaultDispersion)

simulate_binomial(N = 10000, link = defaultLink,
  weights = defaultWeights, unrelated = 0,
  dispersion = defaultDispersion)

simulate_gamma(N = 10000, link = defaultLink,
  weights = defaultWeights, unrelated = 0,
  dispersion = defaultDispersion)

simulate_poisson(N = 10000, link = defaultLink,
  weights = defaultWeights, unrelated = 0,
  dispersion = defaultDispersion)

simulate_inverse_gaussion(N = 10000, link = defaultLink,
  weights = defaultWeights, unrelated = 0,
  dispersion = defaultDispersion)
```

Arguments

N	Sample size. (Default: 10000)
link	Link function. See family for details.
weights	Betas in glm model. See details. simulate_binomial: c(.1, .2) All other: c(1, 2, 3)
unrelated	Number of unrelated features to return. (Default: 0)
dispersion	Dispersion parameter for continuous families. See details.

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.

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

The default value for argument weights works well for all link family combinations. The functions also validate input and provide helpfull error messages. Mistakes like passing a link of "1/mu^2 to the gaussion function will error.

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). If this happens, the function will error with a helpfull 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 inverse gaussion, the inverse of the default link function needs weights*X to be postive.

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 all continuous families have a dispersion 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 gaussion, it is the dispersion parameter. Defalut value 1/3. For the discrete families, this argument is not used.

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 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)) +
  geom_histogram(bins = 30)
rm(simdata)

# Is AIC lower for the correct link? For ten thousand data points, depends on seed!
# For larger N, AIC is lower.
```

```
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 forward 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 resposne 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