Package: coreSim (via r-universe)

October 11, 2024

Type Package

Title Core Functionality for Simulating Quantities of Interest from Generalised Linear Models

Description Core functions for simulating quantities of interest from generalised linear models (GLM). This package will form the backbone of a series of other packages that improve the interpretation of GLM estimates.

Version 0.2.4

Date 2017-05-15

BugReports https://github.com/christophergandrud/coreSim/issues

License GPL (>= 3)

LazyData TRUE

RoxygenNote 6.0.1

Depends R (>= 2.10)

Imports dplyr (>= 0.5.0), MASS

Suggests car, splines, survival, testthat

Repository https://christophergandrud.r-universe.dev

RemoteUrl https://github.com/christophergandrud/coresim

RemoteRef HEAD

RemoteSha f5b7a79111397c29b4c8e43c4fd7b1fc43d3307d

Contents

Admission — *Graduate school admissions data*

Description

A data set containing 400 graduate school admissions decisions.

Usage

```
Admission
```

Format

A data set with 400 rows and 4 variables.

Source

UCLA IDRE http://stats.idre.ucla.edu/r/dae/logit-regression/

b_sim — *Simulate coefficients from a GLM by making draws from the multivariate normal distribution*

Description

Simulate coefficients from a GLM by making draws from the multivariate normal distribution

Usage

```
b_sim(obj, mu, Sigma, nsim = 1000)
```

Arguments

obj	a fitted model object.
mu	an optional vector giving the means of the variables. If obj is supplied then mu is ignored.
Sigma	an optional positive-definite symmetric matrix specifying the covariance matrix of the variables. If obj is supplied then Sigma is ignored. If your model includes an intercept, this should be given the name intercept_.
nsim	number of simulations to draw.

Value

A data frame of simulated coefficients from obj.

Examples

```
library(car)

# Estimate model
m1 <- lm(prestige ~ education + type, data = Prestige)

# Create fitted values
prestige_sims <- b_sim(m1)

# Manually supply coefficient means and covariance matrix
coefs <- coef(m1)
vcov_matrix <- vcov(m1)

prestige_sims_manual <- b_sim(mu = coefs, Sigma = vcov_matrix)
```

---

linear_systematic *Find the systematic component in the linear form for fitted values in across each simulation (note: largely for internal use by qi_builder)*

Description

Find the systematic component in the linear form for fitted values in across each simulation (note: largely for internal use by qi_builder)

Usage

```
linear_systematic(b_sims, newdata, inc_intercept = TRUE)
```

Arguments

b_sims	a data frame created by b_sim of simulated coefficients.
newdata	a data frame of fitted values with column names corresponding to variable names in b_sims. Variables in b_sim not present in newdata will be treated as fitted at 0. Interactions will automatically be found if they were entered into to the model using the * operator.
inc_intercept	logical whether to include the intercept in the lineary systematic component.

Value

A data frame fitted values supplied in newdata and associated linear systematic component estimates for all simulationed coefficient estimates. The linear systematic components are included in a column named ls_.

Source

King, Gary, Michael Tomz, and Jason Wittenberg. 2000. "Making the Most of Statistical Analyses: Improving Interpretation and Presentation." American Journal of Political Science 44(2): 341-55.

Examples

```
library(car)

# Estimate model
m1 <- lm(prestige ~ education + type, data = Prestige)

# Create fitted values
fitted_df <- expand.grid(education = 6:16, typewc = 1)

# Simulate coefficients
m1_sims <- b_sim(m1, nsim = 1000)

# Find linear systematic component for fitted values
ls <- linear_systematic(b_sims = m1_sims, newdata = fitted_df)
```

`qi_builder` *Find quantities of interest from generalized linear models*

Description

Find quantities of interest from generalized linear models

Usage

```
qi_builder(obj, newdata, FUN, ci = 0.95, nsim = 1000, slim = FALSE,
  large_computation = FALSE, original_order = FALSE, b_sims, mu, Sigma,
  verbose = TRUE, ...)
```

Arguments

`obj`	a fitted model object from which to base coefficient simulations on.
`newdata`	an optional data frame of fitted values with column names corresponding to coefficient names in `obj` or `mu`/`Sigma`. Note that variables not included in `newdata` will be fitted at 0. If `missing` then observations used to fit the model in `obj` will be used.
`FUN`	a function for calculating how to find the quantity of interest from a vector of the fitted linear systematic component. It must return a numeric vector. If `missing` then a normal linear regression model is assumed and the predicted values are returned (i.e. the fitted linear systematic component from `linear_systematic`).
`ci`	the proportion of the central interval of the simulations to return. Must be in (0, 1] or equivalently (0, 100]. Note: if `ci = 1` then the full interval (i.e. 100 percent) is assumed.
`nsim`	number of simulations to draw.
`slim`	logical indicating whether to (if `FALSE`) return all simulations in the central interval specified by `ci` for each fitted scenario or (if `TRUE`) just the minimum, median, and maxium values. See `qi_slimmer` for more details.

`large_computation`
: logical. If `newdata` is not supplied, whether to allow > 100000 simulated quantities of interest to be found.

`original_order`
: logical whether or not to keep the original scenario order when `slim = TRUE`. Choosing `FALSE` can imporove computation time.

`b_sims`
: an optional data frame created by `b_sim` of simulated coefficients. Only used if `obj` is not supplied.

`mu`
: an optional vector giving the means of the variables. If `obj` or `b_sims` is supplied then `mu` is ignored.

`Sigma`
: an optional positive-definite symmetric matrix specifying the covariance matrix of the variables. If `obj` is supplied then `Sigma` is ignored. If your model includes an intercept, this should be given the name `intercept_`.

`verbose`
: logical. Whether to include full set of messages or not.

`...`
: arguments to passed to `linear_systematic`.

Value

If `slimmer = FALSE` a data frame of fitted values supplied in `newdata` and associated simulated quantities of interest for all simulations in the central interval specified by `ci`. The quantities of interest are in a column named `qi_`.

If `slimmer = TRUE` a data frame of fitted values supplied in `newdata` and the minimum, median, and maximum values of the central interval specified by `ci` for each scenario are returned in three columns named `qi_min`, `qi_median`, and `qi_max`, respectively.

Examples

```
library(car)

## Normal linear model
m1 <- lm(prestige ~ education + type, data = Prestige)

# Using observed data as scenarios
linear_qi_obs <- qi_builder(m1)

# Create fitted values
fitted_df_1 <- expand.grid(education = 6:16, typewc = 1)

linear_qi <- qi_builder(m1, newdata = fitted_df_1)

# Manually supply coefficient means and covariance matrix
coefs <- coef(m1)
vcov_matrix <- vcov(m1)

linear_qi_custom_mu_Sigma <- qi_builder(mu = coefs, Sigma = vcov_matrix,
                                newdata = fitted_df_1)

## Logistic regression
# Load data
data(Admission)
```

```
Admission$rank <- as.factor(Admission$rank)

# Estimate model
m2 <- glm(admit ~ gre + gpa + rank, data = Admission, family = 'binomial')

# Specify fitted values
m2_fitted <- expand.grid(gre = seq(220, 800, by = 10), gpa = c(2, 4),
                         rank = '4')

# Function to find predicted probabilities from logistic regression models
pr_function <- function(x) 1 / (1 + exp(-x))

# Find quantity of interest
logistic_qi_1 <- qi_builder(m2, m2_fitted, FUN = pr_function)

logistic_qi_2 <- qi_builder(m2, m2_fitted, FUN = pr_function,
                         slim = TRUE)
```

`qi_slimmer` *Find maximum, minimum, and median values for each scenario found using* `qi_builder`

Description

Find maximum, minimum, and median values for each scenario found using `qi_builder`

Usage

```
qi_slimmer(df, scenario_var = "scenario_", qi_var = "qi_")
```

Arguments

`df`	a data frame of simulated quantities of interest created by `qi_builder`.
`scenario_var`	character string of the variable name marking the scenarios.
`qi_var`	character string of the name of the variable with the simulated quantity of interest values.

Details

This funciton slims down a simulation data set to some of its key features (minimun, median, and maximum value for each fitted scenario) so that it takes up less memory and can be easily plotted.

The function is incorporated into `qi_builder` and can be run using `slim = TRUE`.

Value

A data frame with the fitted values and the minimum (`qi_min`), median (`qi_median`), and maximum (`qi_max`) values from the central interval specified with `ci` in `qi_builder`.

Examples

```
library(car)

# Normal linear model
m1 <- lm(prestige ~ education + type, data = Prestige)

# Simulate coefficients
m1_sims <- b_sim(m1)

# Create fitted values
fitted_df <- expand.grid(education = 6:16, typewc = 1)

# Find predicted outcomes (95% central interval, by default)
linear_qi <- qi_builder(b_sims = m1_sims, newdata = fitted_df, slim = FALSE)

# Slim data set
linear_slim <- qi_slimmer(linear_qi)
```

Index