Package 'simqi'

July 23, 2025

Type Package

Title Simulate Quantities of Interest from Regression Models

Version 0.2.0

Depends R (>= 3.5.0)

Maintainer Steve Miller <steve@svmiller.com>

Description This is an all-encompassing suite to facilitate the simulation of so-called quantities of interest by way of a multivariate normal distribution of the regression model's coefficients and variance-covariance matrix.

License GPL-2

Encoding UTF-8

LazyData true

Imports MASS, tibble, methods

RoxygenNote 7.3.2

NeedsCompilation no

Author Steve Miller [aut, cre] (ORCID: <https://orcid.org/0000-0003-4072-6263>)

Repository CRAN

Date/Publication 2025-03-26 07:30:02 UTC

Contents

kgss_sample *A Sample of Korean General Social Survey Data, 2023*

Description

This is a simple sample of the Korean General Social Survey (KGSS) data from 2023.

Usage

```
kgss_sample
```

Format

A data frame with the following variables.

`year` a numeric vector communicating the year of the survey

`respid` a numeric vector communicating a unique identifier for the respondent

`age` a numeric vector communicating the age of the respondent

`female` a numeric vector communicating whether the respondent self-identifies as a woman or a man

`employed` a dummy variable for whether the respondent is employed or 'not employed'

`unived` a dummy variable for whether the respondent has a four-year university degree

`netuse` a numeric vector for hours of internet usage for the respondent

`ideo` a numeric vector communicating the ideology of the respondent on a 1-5 scale. 1 = "very liberal". 5 = "very conservative"

`si_gbh` a numeric vector of the extent to which the respondent considers gender-based hatred (toward both men and women) to be a serious issue. 1 = 'very serious'. 5 = 'not serious at all'

`satisfin` a numeric vector communicating the respondent's satisfaction with their financial situation. 1 = 'very dissatisfied'. 5 = 'very satisfied'

`fp_mord` a numeric vector for whether the respondent thinks maintaining order is the most important priority for South Korea.

`fpcat` a character vector for what the respondent believes is the most important priority for South Korea. Possible values: "Maintain Order", "Fight Rising Prices", "Give People More Say", "Protect Freedom of Speech".

`cntryaffq` a character vector for the country to which the respondent feels closest. Possible values include "USA", "China", "North Korea", "Russia", and "Japan".

Details

Missingness is substantial for one reason or the other. The data are complete cases only. It's not problematic for this purpose, but I did want to make a note of it.

Data were created based on the English version of the data made available by the Survey Research Center at Sung Kyun Kwan University.

sim_qi *Get simulations from a model object*

Description

`sim_qi()` is a function to simulate quantities of interest from a regression model

Usage

```
sim_qi(
  mod,
  nsim = 1000,
  newdata,
  original_scale = TRUE,
  return_newdata = FALSE,
  vcov = NULL
)
```

Arguments

`mod`	a model object
`nsim`	number of simulations to be run, defaults to 1,000
`newdata`	A data frame with a hypothetical prediction grid. If absent, defaults to the model frame.
`original_scale`	logical, defaults to TRUE. If TRUE, the ensuing simulations are returned on their original scale. If FALSE, the ensuing simulations are transformed to a more practical/intuitive quantity that is potentially more intuitive for the user (e.g. a probability for a logistic regression). This argument is ignored in the context of simulations on the linear model.
`return_newdata`	logical, defaults to FALSE. If TRUE, the output returns additional columns corresponding with the inputs provided to `newdata`. This may facilitate easier transformation along with greater clarity as to what the simulations correspond.
`vcov`	a manual variance-covariance matrix to supply to the simulation process. Use with caution, and if you did some kind of on-the-fly standard error adjustment in your regression model to make your results "robust". If nothing is supplied, the model's default variance-covariance matrix is used.

Value

`sim_qi()` returns a data frame (as a `tibble`) with the quantities of interest and identifying information about the particular simulation number. For linear models, or simple generalized linear models where the dependent variable is either "there" or "not there", the quantity of interest returned is a single column (called `y`). For models where the underlying estimation of the dependent variable is, for lack of a better term, "multiple" (e.g. ordinal models with the basic proportional odds assumption), the columns returned correspond with the number of distinct values of the outcome. For example, an ordinal model where there are five unique values of the dependent variable will return columns `y1`, `y2`, `y3`, `y4`, and `y5`.

Supported Models

1. Linear models produced by `lm()` in base R.
2. Generalized linear models produced by `glm()`. Families (links) include:
   - Binomial (logit, probit)
   - Poisson (log)
3. Cumulative link models produced by **ordinal** package.
   - Links: logit, probit

What `original_scale` Does in This Function:

`original_scale` defaults to TRUE in this function. When TRUE, the simulated quantity that's returned is a quantity on its "original scale." Understanding what exactly that means requires some knowledge about the model in question and what exactly the model is estimating on your behalf. In the simple linear model produced by the `lm()` function in base R, this is straightforward (and, thus, this argument does nothing for that model). The quantity returned is the estimated value of the dependent variable. However, models with some kind of link function return fitted values on some particular scale that might not be so user-friendly (e.g. a probit index, or a natural logged odds). However, that is the "original scale" on which the fitted values are returned. This summary table may help you better understand what this argument does with respect to what you want.

Model Function	*Family (Link)*	*original_scale = TRUE*	*original_scale = FALSE*
`lm()`	NA	NA (Estimated value of y)	NA (Estimated value of y)
`glm()`	binomial(link='logit')	natural logged odds of $y = 1$	probability of $y = 1$
`glm()`	binomial(link='probit')	probit index of $y = 1$	probability of $y = 1$
`glm()`	poisson(link='log')	logged lambda	lambda
`clm()`	link = 'logit'	natural logged odds of y = value j	probability of y = value j
`clm()`	link = 'probit'	probit index of y = value j	probability of y = value j
`logistf()`	NA	natural logged odds of $y = 1$	probability of $y = 1$

For ordinal models, I recommend setting `original_scale` to be FALSE. The function, underneath the hood, is actually calculating things on the level of the probability. It's just transforming back to a natural logged odds or a probit index, if that is what you say you want.

Other Details:

Specifying a variable in `newdata` with the exact same name as the dependent variable (e.g. `mpg` in the simple example provided in this documentation file) is necessary for matrix multiplication purposes. The function will do that for you if you have not done it yourself. I recommend letting this function do that for you. For matrix multiplication purposes, this column this function creates will have a default of 0. It does not (should not?) matter for the simulations.

If nothing is supplied in `newdata`, `model.frame()` is called and the simulations are run on the data that inform the model itself. I don't recommend this, but it works for debugging purposes.

This function builds in an implicit assumption that your dependent variable in the regression model is not called `y`. Nothing about this function will misbehave (as far as I know) if your dependent variable is called `y` in the model, but it may lead to some confusion in how you interpret the results of the simulations. The simulated values are always returned as a column called `y`.

Factors (so-called "fixed effects") behave curiously in this function. For now, this function will politely assume your factors are *all* present in the `newdata` you create (even if you don't want

them). Future updates will try to understand this behavior better. The only loss here is the efficiency of the simulation procedure, especially if you are not interested in simulated values of the dependent variable for particular combinations of the factor variable.

Examples

```
set.seed(8675309)

M1 <- lm(mpg ~ hp + am, mtcars)

newdat <- data.frame(am = c(0,1), hp = 123)

sim_qi(M1, nsim = 100, newdat, return_newdata = TRUE)
```

---

som_sample *A Sample of SOM Institute Data, 2019-2020*

Description

This is a simple sample of the SOM (Society-Opinion-Media) data that is run by the University of Gothenburg. The sample comes from the cumulative data set (v. 2021-1) for observations in 2019-2020. The SOM Institute Cumulative Dataset contains data from the National SOM study, which is an annually repeated cross-sectional self-administered mail survey conducted in Sweden since 1986. I think of it as a Swedish corollary to the General Social Survey in the United States. I'll use it for it various testing purposes.

Usage

```
som_sample
```

Format

A data frame with 2841 observations on the following 14 variables.

`year` a numeric vector communicating the year of the survey

`idnr` a numeric vector communicating a unique identifier for the respondent

`lan` a character vector for the county in which the respondent lives

`lptrust` a numeric vector communicating what I term "political trust" of the respondent. I include more information about this variable in the details section.

`satisdem` a numeric vector, ranging from 1-4, communicating satisfaction with democracy in Sweden. 1 = not at all satisfied. 4 = very satisfied.

`trust_rf` a numeric vector, ranging from 1-5, communicating trust in the royal family of Sweden. 1 = very low trust. 5 = very high trust.

`attitude_eu` a numeric vector, ranging from 1-5, communicating the attitude of the respondent toward the European Union. 1 = very negative. 5 = very positive.

`age` a numeric vector communicating the age of the respondent

`female` a numeric vector communicating whether the respondent self-identifies as a woman or a man

`edu3` a numeric vector ranging from 1-3 communicating an education-level attained. 1 = "low" (below grade 9). 2 = "medium" (above grade 9, but below university). 3 = "high" (i.e. at least some university)

`ideo` a numeric vector communicating the ideology of the respondent on a 1-5 scale. 1 = "clearly to the left". 5 = "clearly to the right"

`hinc` a numeric vector communicating the gross household income of the respondent on a 1-5 scale. 1 = "very low". 5 = "very high".

`resarea` a numeric vector communicating the area where the respondent lives. 1 = "rural area". 2 = "village". 3 = "city/town". 4 = "Stockholm/Gothenburg/Malmö".

`interestp` a numeric vector communicating the respondent's interest in politics. 1 = "not at all interested". 4 = "very interested".

Details

Missingness is substantial for one reason or the other. The data are complete cases only. It's not problematic for this purpose, but I did want to make a note of it.

The political trust variable is a simple latent estimate derived from a graded response model of the items from the original data on trust in government (`aa10a`), trust in parliament (`aa10n`), trust in the political parties (`aa10q`), and trust in Swedish politicians (`ab12`). The first three items were on 1-5 scales while the last one (about Swedish politicians) is on a 1-4 scale. All items were reverse coded from their original scales and the user should interpret the ensuing latent estimate to be communicating higher political trust with higher values on the scale. The user is also free to question just how valid of a measure of political trust this is, though I will only add that the factor loadings for all four items were as low as .81 and as high as .91. The proportional variance is .764.

The variables for satisfaction with democracy, trust in the royal family, attitude about the European Union, and interest in politics are reverse coded from their original scale.

SOM is unique from other long-standing survey data sets of which I'm aware by allowing respondents to self-identify as some other gender. In 2019 and 2020, only 71 of 21,195 respondents self-identified this way (before any other case-exclusions). I remove these observations from the data.

If I understand the codebook correctly, the household income variable is coded by SOM's researchers and is not a self-placement by the respondent.

You may want to explicitly factor the residential area variable, though this is basically how it was presented in the data.

Index