Package: nlsur (via r-universe)

September 12, 2024

Type Package

Version 0.8

Title Estimating Nonlinear Least Squares for Equation Systems

Author Jan Marvin Garbuszus

Maintainer Jan Marvin Garbuszus <jan.garbuszus@ruhr-uni-bochum.de>

Depends R (>= 4.1.0)

Suggests testthat

Imports Rcpp (>= 1.0.1), Matrix, methods, stats

Description ``Estimation of Nonlinear Least Squares (NLS), Feasible Generalized NLS (FGNLS) and Iterative FGNLS (IFGNLS) for Equation Systems."

License MIT + file LICENCE

LinkingTo Rcpp, RcppArmadillo

ByteCompile yes

Encoding UTF-8

RoxygenNote 7.3.1

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

RemoteUrl https://github.com/JanMarvin/nlsur

RemoteRef HEAD

RemoteSha 5f9cad2d99e02f48064801f6b35f636daaca4f28

Contents

`.nlsur` *Non-Linear Seemingly Unrelated Regression*

Description

`.nlsur()` is a function for estimation of a non-linear seemingly unrelated regression model in R.

Usage

```
.nlsur(
  eqns,
  data,
  startvalues,
  S = NULL,
  robust = robust,
  nls = FALSE,
  fgnls = FALSE,
  ifgnls = FALSE,
  qrsolve = FALSE,
  MASS = FALSE,
  trace = FALSE,
  eps = eps,
  tau = tau,
  maxiter = maxiter,
  tol = tol,
  initial = initial
)
```

Arguments

`eqns`	is can be a single equation or a equation system. If eqns is a single equation it will internally be converted to a list. Estimation of a single equation might as well be done using `nls()`.

`data`	is the data set on which the equation is applied. This can be of every type `eval()` can handle.
`startvalues`	is a vector of initial start values. For
`S`	is a weighing matrix used for estimation in Feasible Generalized Non-Linear Least Squares (FGNLS) and Iterative FGNLS. For `nlsur()` this is assumed to be the identity matrix. Hence, it is not included. If included S is expected to be a matrix.
`robust`	should a robust standard error be calculated
`nls`	is a logical and default if estimation is done for NLSUR or NLS.
`fgnls`	is a logical and must be set, if estimation is done for FGNLS. This is called in a function called `fgnls()` and should not be set by the user.
`ifgnls`	is a logical and must be set, if estimation is done for ifgnls. This is called in a function called `nlsur()` and should not be set by the user.
`qrsolve`	is a logical, if TRUE `qr.coef(qr(x), r)` is called which should be the most robust way for estimation of nls. For this all equations will be rbinded, which might lead to memory bottlenecks.
`MASS`	is a logical, if TRUE `lm_gls()` is called for estimation of a linear regression with a weighting matrix.
`trace`	is a logical. If TRUE the current iterations SSR is called.
`eps`	the epsilon used for convergence in nlsur(). Default is 1e-5.
`tau`	is another convergence variable. Default is 1e-3.
`maxiter`	maximum number of iterations
`tol`	qr.solves tolerance for detecting linear dependencies.
`initial`	logical if initial calculation set. used to avoid calculation of svd numerous times

Details

nlsur is a function for estimation of a non-linear least squares (NLS). In addition to `nls()` it is capable of estimation of system of equations. This estimation is done in a non-linear seemingly unrelated regression approach.

References

Bates, D. M. and Watts, D. G. (1988) Nonlinear Regression Analysis and Its Applications, Wiley

Gallant, A. Ronald (1987): Nonlinear Statistical Models. Wiley: New York

ai *Estimation of an Almost-Ideal Demand System*

Description

Estimation of an Almost-Ideal Demand System using nlsur().

Usage

```
ai(w, p, x, z, a0 = 0, data, scale = FALSE, logp = TRUE, logexp = TRUE, ...)
```

Arguments

w	character vector of m budgetshares used for estimation.
p	character vector of m prices.
x	single character vector of total expenditure.
z	character vector of k demographic variables.
a0	start value for translog price index.
data	data.frame containing the variables.
scale	logical if TRUE Rays (1983) scaling is used.
logp	logical if prices are log prices.
logexp	logical if expenditure is log expenditure.
...	additional options passed to nlsur

References

Deaton, Angus S., Muellbauer, John: An Almost Ideal Demand System, The American Economic Review 70(3), American Economic Association, 312-326, 1980

Ray, Ranjan: Measuring the costs of children: An alternative approach, Journal of Public Economics 22(1), 89-102, 1983

See Also

qai and ai.model

`ai.model` *Q/AI model function*

Description

Modelfunction to create Deatons and Muellbauers (1980) famous Almost-Ideal Demand System or the Quadratic Almost-Ideal Demand System by Banks et al. (1997).

Usage

```
ai.model(
  w,
  p,
  exp,
  alph0 = 10,
  logp = TRUE,
  logexp = TRUE,
  priceindex = "translog",
  modeltype = "AI",
  ray = FALSE,
  demogr
)
```

Arguments

`w`	character vector of m budgetshares used for estimation.
`p`	character vector of m prices.
`exp`	single character vector of total expenditure.
`alph0`	start value for translog price index.
`logp`	logical if prices are log prices.
`logexp`	logical if expenditure is log expenditure.
`priceindex`	character either "translog" or "S" for the stone price index.
`modeltype`	character either "AI" or "QAI" for AI or QAI model.
`ray`	logical if Ray (1983) scaling should be included. If TRUE requires demographic vector.
`demogr`	character vector of k demographic variables.

Details

While Ray and Stata use log(m0) for the demographic variables, there is no guarantee, that m0 is positive. The model relies much on the correct starting values. Therefore log(abs(m0)) is used.

References

Deaton, Angus S., Muellbauer, John: An Almost Ideal Demand System, The American Economic Review 70(3), American Economic Association, 312-326, 1980

Banks, James, Blundell, Richard, Lewbel, Arthur: Quadratic Engel Curves and Consumer Demand, The Review of Economics and Statistics 79(4), The MIT Press, 527-539, 1997

Ray, Ranjan: Measuring the costs of children: An alternative approach, Journal of Public Economics 22(1), 89-102, 1983

See Also

ai and qai

`arma_reshape` *Reshape matrix for blockwise WLS estimation*

Description

reshape mm for blockwise multiplication in wls_est

Usage

```
arma_reshape(mm, sizetheta)
```

Arguments

`mm`	a matrix
`sizetheta`	integer of length(theta) to shrink mm into

Examples

```
mm <- matrix(c(11,21,31,41,
 12,22,32,42,
 13,23,33,43,
 14,24,34,44),
 ncol = 4)

mm_a <- arma_reshape(mm, 2)

mm_m <- matrix(t(mm), nrow = 2, byrow = TRUE)
```

constant *Check if formula contains constant*

Description

Check if formula contains constant

Usage

```
constant(x)
```

Arguments

x	formula

Details

Primitive function to check a formula for a constant part. Function checks first and last term on rhs for constant variables at front and back position.

Examples

```
## Not run:
constant(y ~ x + a * z) # x
constant(y ~ x * b + 1) # 1
constant(y  ~ 0 + x) # NULL
constant(y  ~ x) # x
constant(y ~ x1 * b1 + b0 + x2 * b2) # wont find b0
constant( y  ~ (x*b +k) + a*y + b*z ) # wont find k
constant( y  ~ (k+ x*b) + a*y + b*z ) # k
constant( y  ~  a*y + b*z + (k + x*b) ) # wont find k
constant( y  ~  a*y + b*z + (x*b + k) ) # k

## End(Not run)
```

costs *PRICE AND QUANTITY INDEXES OF CAPITAL, LABOR, ENERGY, AND OTHER INTERMEDIATE INPUTS and TOTAL COST AND COST SHARES OF CAPITAL, LABOR, ENERGY, AND OTHER INTERMEDIATE MATERIALS - U.S. MANUFACTURING 1947-1971*

Description

A dataset combining tables 1 and 2 of Brendt and Wood 1975

Usage

```
data(costs)
```

Format

A data frame with 25 rows and 14 cols

Details

- Year years from 1947 - 1971
- Cost Total Input Cost in billion dollars (Table 2)
- Sk Cost Shares K (Table 2)
- Sl Cost Shares L (Table 2)
- Se Cost Shares E (Table 2)
- Sm Cost Shares M (Table 2)
- Pk Price Index K (Table 1)
- Pl Price Index L (Table 1)
- Pe Price Index E (Table 1)
- Pm Price Index M (Table 1)
- K Quantity Index K (Table 1)
- K Quantity Index L (Table 1)
- K Quantity Index E (Table 1)
- K Quantity Index M (Table 1)

References

Berndt, E. R. and Wood, D. O. (1975). Technology, prices, and the derived demand for energy. The review of Economics and Statistics, pages 259–268.

---

cov_robust *Calculate a robust covariance matrix*

Description

As discussed in Wooldridge (2002, 160)

Usage

```
cov_robust(x, u, qS, w, sizetheta)
```

Arguments

x	matrix of derivatives
u	u
qS	weighting matrix
w	vector of weights
sizetheta	sizetheta

dm *dm simple delta method implementation*

Description

dm simple delta method implementation

Usage

```
dm(object, form, level = 0.05)
```

Arguments

object	of class nlsur
form	formula e.g. "be/bk".
level	value for conf. interval default is 0.05

elasticities *Estimation of elasticities of the (Quadratic) Almost-Ideal Demand System*

Description

Estimates the income/expenditure elasticity, the uncompensated price elasticity and the compensated price elasticity

Usage

```
elasticities(object, data, type = 1, usemean = FALSE)
```

Arguments

object	qai result
data	data vector used for estimation
type	1 = expenditure; 2 = uncompensated; 3 = compensated
usemean	evaluate at mean

Details

Formula for the expenditure (income) elasticity

$$\mu_i = 1 + \frac{1}{w_i}\left[\beta_i + \frac{2\lambda_i}{b(\mathbf{p})} * ln\left\{\frac{m}{a(\mathbf{p})}\right\}\right]$$

Formula for the uncompensated price elasticity

$$\epsilon_{ij} = \delta_{ij} + \frac{1}{w_i}\left(\gamma_{ij} - \beta_i + \frac{2\lambda_i}{b(\mathbf{p})}\right)\left[\ln\left\{\frac{m}{a(\mathbf{p})}\right\}\right] \times \left(\alpha_j + \sum_k \gamma_{jk}\ln p_k\right) - \frac{\beta_j\lambda_i}{b(\mathbf{p})}\left[\ln\left\{\frac{m}{a(\mathbf{p})}\right\}\right]$$

Compensated price elasticities (Slutsky equation)

$$\epsilon_{ij}^C = \epsilon_{ij} + \mu_i w_j$$

References

Banks, James, Blundell, Richard, Lewbel, Arthur: Quadratic Engel Curves and Consumer Demand, The Review of Economics and Statistics 79(4), The MIT Press, 527-539, 1997

Poi, Brian P.: Easy demand-system estimation with quaids, The Stata Journal 12(3), 433-446, 2012

See Also

ai and qai

Examples

```
## Not run:
library(nlsur)
library(readstata13)

dd <- read.dta13("http://www.stata-press.com/data/r15/food.dta")

w <- paste0("w", 1:4); p <- paste0("p", 1:4); x <- "expfd"

est <- ai(w = w, p = p, x = x, data = dd, a0 = 10, scale = FALSE,
          logp = F, logexp = F)

mu <- elasticities(est, data = dd, type = 1, usemean = FALSE)

ue <- elasticities(est, data = dd, type = 2, usemean = FALSE)

ce <- elasticities(est, data = dd, type = 3, usemean = FALSE)
## End(Not run)
```

getstartvals *Function to create startvalues for nlsur models*

Description

Function to create startvalues for nlsur models

Usage

```
getstartvals(model, data, val)
```

Arguments

model	nlsur model
data	the data frame used for evaluation
val	value

is.formula *Check if object is of class formula*

Description

Check if object is of class formula

Usage

```
is.formula(x)
```

Arguments

x	object

lm_gls *Calculate WLS using sparse matrix and qr*

Description

calculate WLS using eigen similar to the approach in MASS::lm.gls

Usage

```
lm_gls(X, Y, W, neqs, tol = 1e-07, covb = FALSE)
```

Arguments

X	n x m X matrix
Y	n x k matrix
W	n x n
neqs	k
tol	tolerance for qr
covb	if true covb is calculated else theta

nlcom *Estimate nonlinear combinations of nlsur estimates*

Description

Estimate nonlinear combinations of nlsur estimates

Usage

```
nlcom(object, form, alpha = 0.05, rname, envir)
```

Arguments

object	of class nlsur
form	formula e.g. "be/bk". May contain names(coef(object)) or prior nlcom estimations.
alpha	value for conf. interval default is 0.05
rname	optional rowname for result
envir	optional name of environment to search for additional parameters

See Also

deltaMethod

Examples

```
## Not run:
dkm <- nlcom(object = erg, form = "-dkk -dkl -dke", rname = "dkm")
dkm

dlm <- nlcom(object = erg, form = "-dkl -dll -dle", rname = "dlm")
dlm

dem <- nlcom(object = erg, form = "-dke -dle -dee", rname = "dem")
dem

dmm <- nlcom(object = erg, form = "-dkm -dlm -dem", rname = "dmm")
dmm

# last one is equivalent to the longer form of:
dmm <- nlcom(object = erg,
 form = "-(-dkk -dkl -dke) -(-dkl -dll -dle) -(-dke -dle -dee)")
dmm

## End(Not run)
```

`nlsur` *Fitting Iterative Feasible Non-Linear Seemingly Unrelated Regression Model*

Description

`nlsur()` is used to fit nonlinear regression models. It can handle the feasible and iterative feasible variants.

Usage

```
nlsur(
  eqns,
  data,
  startvalues,
  type = NULL,
  S = NULL,
  trace = FALSE,
  robust = FALSE,
  stata = TRUE,
  qrsolve = FALSE,
  weights,
  MASS = FALSE,
  maxiter = 1000,
  val = 0,
  tol = 1e-07,
```

```
  eps = 1e-05,
  ifgnlseps = 1e-10,
  tau = 0.001,
  initial = FALSE
)
```

Arguments

eqns	is a list object containing the model as formula. This list can handle contain only a single equations (although in this case nls() might be a better choice) or a system of equations.
data	an (optional) data frame containing the variables that will be evaluated in the formula.
startvalues	initial values for the parameters to be estimated.
type	can be 1 Nonlinear Least Squares (NLS), 2 Feasible Generalized NLS (FGNLS) or 3 Iterative FGNLS (IFGNLS) or the respective abbreviations in character form.
S	is a weight matrix used for evaluation. If no weight matrix is provided the identity matrix I will be used.
trace	logical whether or not SSR information should be printed. Default is FALSE.
robust	logical if true robust standard errors are estimated.
stata	is a logical. If TRUE for nls a second evaluation will be run. Stata does this by default. For this second run Stata replaces the diagonal of the I matrix with the coefficients.
qrsolve	logical
weights	Additional weight vector.
MASS	is a logical whether an R function similar to the MASS::lm.gls() function should be used for weighted Regression. This can cause sever RAM usage as the weight matrix tend to be huge (n-equations * n-rows).
maxiter	Maximum number of iterations.
val	If no start values supplied, create them with this start value. Default is 0.
tol	qr.solves tolerance for detecting linear dependencies.
eps	the epsilon used for convergence in nlsur(). Default is 1e-5.
ifgnlseps	is epsilon for ifgnls(). Default is 1e-10.
tau	is another convergence variable. Default is 1e-3.
initial	logical value to define if rankMatrix is calculated every iteration of nlsur.

Details

nlsur() is a wrapper around .nlsur(). The function was initially inspired by the Stata Corp Function nlsur. Nlsur estimates a nonlinear least squares demand system. With nls, fgnls or ifgnls which is equivalent to Maximum Likelihood estimation. Nonlinear least squares requires start values and nlsur requires a weighting matrix for the demand system. If no weight matrix is provided, nlsur will

use the identity matrix I. If type = 1 or type = "nls" is added, nlsur will use the matrix for an initial estimation, once the estimation is done, it will swap the diagonal with the estimated results.

Most robust regression estimates shall be returned with both qrsolve and MASS TRUE, but memory consumption is largest this way. If MASS is FALSE a memory efficient RcppArmadillo solution is used for fgnls and ifgnls. If qrsolve is FALSE as well, only the Armadillo function is used.

If `robust` is selected Whites HC0 is used to calculate Heteroscedasticity Robust Standard Errors.

If `initial` is TRUE rankMatrix will be calculated every iteration of nlsur. Meaning for nls at least once, for fgnls at least twice and for ifgnls at least three times. This adds a lot of overhead, since rankMatrix is used to calculate k. To assure that k does not change this can be set to TRUE.

Nlsur has methods for the generic functions coef, confint, deviance, df.residual, fitted, predict, print, residuals, summary and vcov.

Value

The function returns a list object of class nlsur. The list includes:

coefficients: estimated coefficients

residuals: residuals

xi: residuals of each equation in a single list

eqnames: list of equation names

sigma: the weight matrix

ssr: Residual sum of squares

lhs: Left hand side of the evaluated model

rhs: Right hand side of the evaluated model

nlsur: model type. "NLS", "FGNLS" or "IFGNLS"

se: standard errors

t: t values

covb: asymptotic covariance matrix

zi: equation wise estimation results of SSR, MSE, RMSE, MAE, R2 and Adj-R2. As well as n, k and df.

model: equation or system of equations as list containing formulas

References

Gallant, A. Ronald (1987): Nonlinear Statistical Models. Wiley: New York

See Also

nls

Examples

```
## Not run:
# Greene Example 10.3
library(nlsur)

url <- "http://www.stern.nyu.edu/~wgreene/Text/Edition7/TableF10-2.txt"
dd <- read.table(url, header = T)

names(dd) <-
 c("Year", "Cost", "Sk", "Sl", "Se", "Sm", "Pk", "Pl", "Pe", "Pm")


eqns <-
 list( Sk ~ bk + dkk * log(Pk/Pm) + dkl * log(Pl/Pm) + dke * log(Pe/Pm),
       Sl ~ bl + dkl * log(Pk/Pm) + dll * log(Pl/Pm) + dle * log(Pe/Pm),
       Se ~ be + dke * log(Pk/Pm) + dle * log(Pl/Pm) + dee * log(Pe/Pm))

strtvls <- c(be = 0, bk = 0, bl = 0,
             dkk = 0, dkl = 0, dke = 0,
             dll = 0, dle = 0, dee = 0)


erg <- nlsur(eqns = eqns, data = dd, startvalues = strtvls, type = 2,
             trace = TRUE, eps = 1e-10)

erg

## End(Not run)
```

`predict.nlsur` *Predict for Non-Linear Seemingly Unrelated Regression Models*

Description

`predict()` is a function to predict nlsur results.

Usage

```
## S3 method for class 'nlsur'
predict(object, newdata, ...)
```

Arguments

`object`	is an nlsur estimation result.
`newdata`	an optional data frame for which the prediction is evaluated.
`...`	further arguments for predict. At present no optional arguments are used.

Details

predict.nlsur evaluates the nlsur equation(s) given nlsurs estimated parameters using either the original data.frame or newdata. Since nlsur() restricts the data object only to complete cases observations with missings will not be fitted.

Examples

```
# predict(nlsurObj, dataframe)
```

---

qai *Estimation of an Quadratic Almost-Ideal Demand System*

Description

Estimation of an Quadratic Almost-Ideal Demand System using nlsur().

Usage

```
qai(w, p, x, z, a0 = 0, data, scale = FALSE, logp = TRUE, logexp = TRUE, ...)
```

Arguments

w	character vector of m budgetshares used for estimation.
p	character vector of m prices.
x	single character vector of total expenditure.
z	character vector of k demographic variables.
a0	start value for translog price index.
data	data.frame containing the variables.
scale	logical if TRUE Rays (1983) scaling is used.
logp	logical if prices are log prices.
logexp	logical if expenditure is log expenditure.
...	additional options passed to nlsur

References

Banks, James, Blundell, Richard, Lewbel, Arthur: Quadratic Engel Curves and Consumer Demand, The Review of Economics and Statistics 79(4), The MIT Press, 527-539, 1997

Ray, Ranjan: Measuring the costs of children: An alternative approach, Journal of Public Economics 22(1), 89-102, 1983

See Also

ai and ai.model

ssr_est *Estimate residual sum of squares*

Description

calculate SSR where $SSR(\beta) = u'D'Du$.

Usage

```
ssr_est(r, s, w)
```

Arguments

r	residuals
s	weighting matrix
w	vector of weights

wls_est *Blockwise WLS estimation*

Description

Blockwise WLS estimation. Usually for $(X'X)^{-1}W^{-1}X'Y$ X and Y X, W and Y are of similar dimensions. In nlsur W is a cov-matrix of size $k \times k$ and usually way smaller than X. To avoid blowing all matrices up for the estimation, a blockwise approach is used. X is shrunken to match size k. W is D'D so XDX is calculated. XDy is only calculated if wanted for a full WLS. For the cov-matrix only XDX is required.

Usage

```
wls_est(x, r, qS, w, sizetheta, fullreg, tol)
```

Arguments

x	matrix of derivatives
r	residual matrix
qS	weighting matrix of sizetheta x sizetheta
w	vector of weights
sizetheta	integer defining the amount of coefficients
fullreg	bool defining if WLS or Cov is calculated
tol	tolerance used for qr()

Details

as reference see: http://www.navipedia.net/index.php/Block-Wise_Weighted_Least_Square

wt_mean *Calculate a weighted mean*

Description

Calculate a weighted mean

Usage

```
wt_mean(x, w)
```

Arguments

x	matrix of derivatives
w	vector of weights

Index